

T.C.  
BİLECİK ŐEYH EDEBALI ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
BİLGİSAYAR MÜHENDİSLİĐİ ANABİLİM DALI

**GÖMÜLÜ SİSTEM ÜZERİNDE DERİN ÖĐRENME BAZLI SÜRÜCÜ YORGUNLUK  
TESPİTİ**

YÜKSEK LİSANS TEZİ

ESRA ÇİVİK

TEZ DANIŐMANI  
PROF. DR. UĐUR YÜZGEÇ

BİLECİK, 2020

10373819

T.C.  
BİLECİK ŐEYH EDEBALI ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
BİLGİSAYAR MÜHENDİSLİĐİ ANABİLİM DALI

**GÖMÜLÜ SİSTEM ÜZERİNDE DERİN ÖĐRENME BAZLI SÜRÜCÜ YORGUNLUK  
TESPİTİ**

YÜKSEK LİSANS TEZİ

ESRA ÇİVİK

TEZ DANIŐMANI  
PROF. DR. UĐUR YÜZGEÇ

BİLECİK, 2020

10373819

## BEYAN

“Gömülü Sistem Üzerinde Derin Öğrenme Bazlı Sürücü Yorgunluk Tespiti” adlı yüksek lisans tezi hazırlık ve yazımı sırasında bilimsel ahlak kurallarına uyduğumu, başkalarının eserlerinden yararlandığım bölümlerde bilimsel kurallara uygun olarak atıfta bulunduğumu, kullandığım verilerde herhangi bir tahrifat yapmadığımı, tezin herhangi bir kısmının Bilecik Şeyh Edebali Üniversitesi veya başka bir üniversitede başka bir tez çalışması olarak sunulmadığını beyan ederim.

Bu çalışmamın, Bilimsel Araştırmalar Projeleri (BAP), TÜBİTAK veya benzeri kuruluşlarca desteklenmesi durumunda; projenin ve destekleyen kurumun adı proje numarası ile birlikte beyan edilmelidir.	
<b>DESTEK ALINMIŞTIR</b>	<b>DESTEK ALINMAMIŞTIR</b>
Destek alındı ise;	
<b>Destekleyen Kurum:</b>	
<b>Desteğin Türü</b>	<b>Proje Numarası</b>
1- BAP (Bilimsel Araştırma Projesi)	
2- TÜBİTAK	
<b>Diğer;</b> .....	

**Öğrenci Adı ve Soyadı**

.....

**Tarih**

.....

**İmza**

.....

## ÖN SÖZ

Bu tez çalışmasının yazılmasında, çalışmamı sahiplenerek takip eden danışmanım Sayın Prof. Dr. Uğur YÜZGEÇ 'e ve Cutesafe Teknoloji Firması çalışanlarına değerli katkı ve emekleri için teşekkürlerimi ve saygılarımı sunarım.

Savunma sınavı sırasında değerli jüri üyeleri/proje danışmanım Sayın Prof. Dr. Cihan KARAKUZU' ya ve Sayın Doç. Dr. Halit ÖZTEKİN'e çalışmamın son haline gelmesindeki değerli katkıları adına teşekkürlerimi ve saygılarımı sunarım.

Son olarak bu günlere ulaşmamdaki emekleri adına değerli aileme teşekkür ederim.

**Esra ÇİVİK**

**Tarih**



## ÖZET

### GÖMÜLÜ SİSTEM ÜZERİNDE DERİN ÖĞRENME BAZLI SÜRÜCÜ YORGUNLUK TESPİTİ

Trafik kazaları, yanlış davranış, dikkatsizlik, ihmal gibi sebeplerin bir araya gelmesi ile oluşmaktadır. Bu kazalar sonucunda can ve mal kayıpları yaşanmaktadır. Dünyadaki trafik kazalarının başlıca nedenlerinden biri sürücünün yorgun ve uykusuz araç kullanmasından kaynaklanmaktadır. Bu nedenle araç içerisinde sürücünün anlık durumu izlenip, yorgunluk tespiti yapılarak kazaların sayısında büyük oranda azalma sağlanabilir. Bunun için gerçek zamanlı çalışan, sürekli olarak sürücüyü izleyen ve yüksek doğrulukla çalışabilen bir sisteme ihtiyaç vardır. Ayrıca, bu sistem araç içerisine yerleştirilebilmesi için gömülü bir cihaz üzerinde çalıştırılabilir olmalıdır. Bu çalışmada, ilgili sorunun çözülebilmesi kapsamında düşük maliyetli gömülü bir cihaz üzerinde gerçek zamanlı çalışan ve derin öğrenme bazlı yüksek doğrulukta başarımlı gösteren yeni bir yaklaşım önerilmiştir. Önerilen sistem, Nvidia Jetson Nano gömülü cihazı üzerinde sürücünün göz ve ağız bölgelerini kullanarak dört farklı durum ile sınıflandırma yapmaktadır. Böylece olası bir kazanın önüne geçilerek can ve mal kaybının minimum seviyeye indirilmesi hedeflenmektedir.

**Anahtar Kelimeler:** Sürücü Yorgunluk; Gömülü Sistem; Gerçek Zamanlı; Derin Öğrenme; Görüntü İşleme; Trafik Kazaları.

## **ABSTRACT**

### **DEEP LEARNING BASED FATIGUE DETECTION IN EMBEDDED SYSTEM**

Traffic accidents are occurred by a combination of causes such as misbehavior, carelessness, and negligence. As a result of these accidents, lethal accidents and property loss are experienced. One of the main causes of traffic accidents in the world is due to tired and sleepless driving. For this reason, the instantaneous situation of the driver in the vehicle can be monitored and fatigue can be detected and the number of accidents can be greatly reduced. For this, there is a need for a system that works in real-time, continuously monitors the driver, and can work with high accuracy. In addition, this system must be operable on an embedded device in order to be placed in the vehicle. In this study, a new approach that works in real-time on a low cost embedded device and shows high accuracy based on deep learning is proposed in order to solve the related problem. The proposed system classifies four different situations using the driver's eye and mouth areas on the Nvidia Jetson Nano embedded device. Thus, it is aimed to minimize the loss of life and property by preventing a possible accident.

**Keywords:** Drowsiness; Embedded System; Real Time; Deep Learning; Image Processing; Traffic Accidents

# İÇİNDEKİLER

	Sayfa No
ÖN SÖZ.....	i
ÖZET .....	ii
ABSTRACT .....	iii
İÇİNDEKİLER.....	iv
TABLolar LİSTESİ .....	vi
ŞEKİLLER LİSTESİ.....	vii
GRAFİKLER LİSTESİ .....	viii
KISALTMALAR VE SİMGELER LİSTESİ .....	ix
1. GİRİŞ.....	1
2. YAPAY SİNİR AĞLARI.....	4
2.1. Yapay Sinir Ağları Yapısı.....	4
2.1.1. Biyolojik Sinir Hücresi .....	4
2.1.2. Ysa Genel Yapısı ve Matematiksel Modeli .....	5
2.2. Yapay Sinir Ağlarının Sınıflandırılması (Modelleri) .....	8
2.2.1. Yapılarına Göre Yapay Sinir Ağları.....	8
2.2.1.1. İleri Beslemeli Ağlar .....	8
2.2.1.2. Geri Beslemeli Ağlar .....	9
2.2.2. Öğrenme Algoritmalarına Göre Yapay Sinir Ağları.....	10
2.2.3. Öğrenme Zamanına Göre Yapay Sinir Ağları.....	11
3. DERİN ÖĞRENME .....	12
3.1. Derin Öğrenme Mimarileri.....	12
3.1.1. Evrişimsel Sinir Ağları (CNN/ConvNet).....	13
3.1.1.1. Evrişimsel Sinir Ağını Oluşturan Katmanlar .....	13
3.1.2. Tekrarlayan Sinir Ağları (RNN) .....	17
3.1.3. Uzun Kısa Dönemli Bellek (LSTM) .....	17

<b>4. SÜRÜCÜ YORGUNLUK TESPİTİ.....</b>	<b>18</b>
<b>4.1. Sistem Mimarisi.....</b>	<b>18</b>
<b>4.2. Yazılım Altyapısı.....</b>	<b>20</b>
<b>4.2.1. Keras Derin Öğrenme Kütüphanesi .....</b>	<b>20</b>
<b>4.2.2. VGG-16 Ağ Modeli .....</b>	<b>21</b>
<b>4.2.3. YawDD Veri Seti.....</b>	<b>21</b>
<b>4.3. Donanım Altyapısı .....</b>	<b>23</b>
<b>5. DENEYSEL SONUÇLAR VE TARTIŞMA.....</b>	<b>25</b>
<b>6. SONUÇ VE TARTIŞMA.....</b>	<b>27</b>
<b>KAYNAKÇA .....</b>	<b>28</b>
<b>ÖZ GEÇMİŞ.....</b>	<b>32</b>

## TABLULAR LİSTESİ

Sayfa No

<b>Tablo 2.1.</b> Biyolojik ve yapay sinir sistemlerinin karşılıkları .....	<b>5</b>
<b>Tablo 4.1.</b> Eğitim veri seti .....	<b>22</b>



## ŞEKİLLER LİSTESİ

	Sayfa No
Şekil 2.1. Biyolojik sinir hücresinin yapısı.....	5
Şekil 2.2. Yapay sinir ağının genel yapısı .....	6
Şekil 2.3. Yapay sinir ağının matematiksel modeli.....	7
Şekil 2.4. İleri beslemeli örnek ağ.....	8
Şekil 2.5. Geri beslemeli ağ ile ileri beslemeli ağ yapılarının karşılaştırılması .....	9
Şekil 2.6. Danışmanlı öğrenme modeli .....	10
Şekil 2.7. Danışmansız öğrenme modeli .....	10
Şekil 3.1. Evrimsel/Konvolüsyonel sinir ağı yapısı.....	13
Şekil 3.2. Konvolüsyon katmanında gerçekleşen işlem örneği .....	14
Şekil 3.3. Konvolüsyon ve ReLu katmanlarının görüntü üzerindeki etkisi.....	15
Şekil 3.4. Maksimum ve ortalama havuzlama işlemlerinin mantığı .....	16
Şekil 3.5. RNN mimarisi çalışma yapısı .....	17
Şekil 3.6. LSTM çalışma yapısı.....	17
Şekil 4.1. Sistem donanımı öbek şeması .....	18
Şekil 4.2. Sistem akış diyagramı .....	19
Şekil 4.3. Sistemin çalışma yapısı .....	19
Şekil 4.4. Keras kütüphanesinin çalışma akışı .....	20
Şekil 4.5. VGG-16 mimarisi.....	21
Şekil 4.6. Göz modeli örnek veri seti .....	22
Şekil 4.7. Ağız modeli örnek veri seti .....	23
Şekil 4.8. Nvidia Jetson Nano gömülü cihazı modeli örnek veri seti.....	23
Şekil 4.9. Sistemin araç içi görüntüsü .....	24
Şekil 5.1. Sürücü yorgunluk tespitinin test edilmesi: normal durum (a), göz açık esneme durumu (b), gözün kapalı olma durumu (c), yorgunluk tespiti (d).....	26

## GRAFİKLER LİSTESİ

Sayfa No

<b>Grafik 1.1.</b> 2009-2018 Trafik kazasına neden olan sürücü, yolcu, yaya, yol ve taşıtın kusur oranları.....	<b>1</b>
<b>Grafik 3.4.</b> ReLu aktivasyon fonksiyonu grafiği.....	<b>15</b>
<b>Grafik 5.1.</b> Göz ve ağız modelleri eğitim ve doğrulama hata kaybı grafikleri.....	<b>24</b>

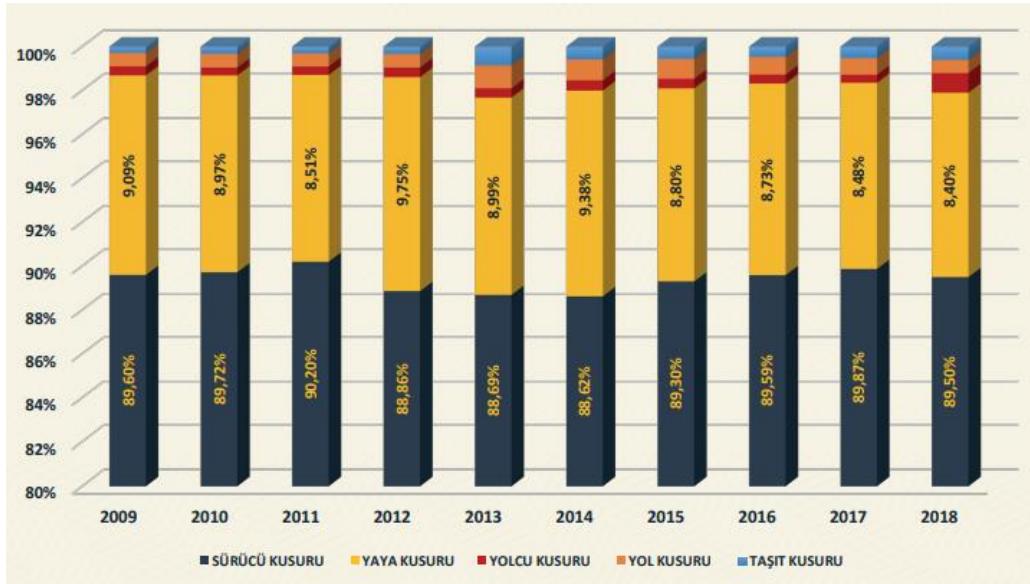


## KISALTMALAR VE SİMGELER LİSTESİ

<b>G<sub>b</sub></b>	: Göz Modelinde Başarılı Test Veri Sayısı
<b>G<sub>t</sub></b>	: Göz Modelinde Kullanılan Toplam Test Veri Sayısı
<b>A<sub>b</sub></b>	: Ağız Modelinde Başarılı Test Veri Sayısı
<b>A<sub>t</sub></b>	: Ağız Modelinde Kullanılan Toplam Test Veri Sayısı
<b>YSA</b>	: Yapay Sinir Ağları
<b>GPU</b>	: Graphics Processing Unit
<b>CNN</b>	: Convolution Neural Networks
<b>RNN</b>	: Recurrent Neural Network
<b>LSTM</b>	: Long Short Term Memory
<b>ReLu</b>	: Rectified Linear Units
<b>YawDD</b>	: Yawning Detection Dataset
<b>NHTSA</b>	: National Highway Traffic Safety Administration

## 1. GİRİŞ

Trafik kazaları büyük bir toplum sorunu olup, maddi ve manevi kayıplara yol açmaktadır. Türkiye’de meydana gelen trafik kazalarının 2018 yılı dağılımı ve özelliklerine bakıldığında; kazaların %84,81’ i maddi hasarlı, %14,66’ sı yaralamalı ve %0,53’ü de ölümlü trafik kazası olarak gerçekleştiği görülmektedir (Yaprak ve Akbulut, 2019). Ayrıca TÜİK (Türkiye İstatistik Kurumu) tarafından yayınlanan raporlara göre trafiğe çıkan araç sayısı da giderek artış göstermektedir (TÜİK, 2019). Bu artış trafik kazalarının artmasına neden olmaktadır. Artan araç sayılarının yanı sıra trafik kazalarına neden olan başlıca unsurlar arasında sürücü, yaya, yol ve yolcu yer almaktadır. Bu kapsamda, Karayolları Trafik Kaza İstatistiklerine göre, ülkemizde 2018 yılında toplam 1 milyon 229 bin 364 adet trafik kazası meydana geldiği belirtilmiştir. Ölümlü ve yaralanmalı toplam 217 bin 898 adet trafik kazalarının kusurlarına bakıldığında, kusurların %89,5’ inin sürücü kaynaklı olduğu belirtilmiştir (TÜİK, 2018). Sürücü kusurlarında ise sürüş esnasında uyku halinden kaynaklanan kazaların oranı Avrupa’da %1-16 iken ülkemizde bu oran %70’ e kadar ulaşmaktadır (URSAVAŞ, A., ve Ercüment, 2004:38). ABD Ulusal Karayolları Trafik Güvenliği Dairesi (NHTSA) verilerine göre yorgun ve uykusuz araç kullanmaya bağlı her yıl yaklaşık 100.000 kaza (tüm kazaların %1,5’ i) ve yılda yaklaşık 1500 ölüm (tüm ölümlü kazaların %4’ ü) meydana gelmektedir (Sönmez, 1999). Grafik 1.1’de yer alan grafiksel görselde, kazalardaki en yüksek kusur oranının ortalama %90 ile sürücüye ait olduğu görülmektedir.



**Grafik 1.1.** 2009-2018 Trafik kazasına neden olan sürücü, yolcu, yaya, yol ve taşıtın kusur oranları

**Kaynak:** (Yaprak ve Akbulut, 2019)

Trafik kazaları, gerekli analizler ve analizlerin sonucunda sorun gidermeye yönelik alınan önlemler ile ciddi oranda azaltılabilir. Bu kapsamda yapılan araştırma ve analizler sonucunda sürücü kaynaklı kazaların oranının yüksek olduğu tespit edilmiştir. Alkollü, yorgun ve uykusuzluk gibi durumlarda araç kullanımı bu oranları oldukça etkilemektedir. Alkollü araç kullanımının ne denli tehlikeli olduğu birçok kişi tarafından bilinmektedir. Ancak yorgun ve uykusuz araç kullanımının en az alkollü araç kullanmak kadar tehlikeli olduğu ve en önemli kaza nedenlerinden biri olduğu göz ardı edilmektedir. Oysa uykusuzluk, tıpkı alkol ve uyuşturucunun etkisi altındayken olduğu gibi sürücü performansını olumsuz etkilemekte ve ciddi trafik kazalarına neden olmaktadır. Literatürde konu ile ilgili yapılan birçok çalışma bulunmaktadır. Girit ve arkadaşları, kamera tarafından kaydedilen video bölümlerini analiz ederek, sürücünün uyanık veya uykulu olduğunu tespit etmek için bir sistem önermişlerdir (Girit vd., 2014). Suryaprasad ve meslektaşları, çalışmalarında gerçek zamanlı sürücü uyku durumunu tespit etmek için video verileri üzerinde çalışmışlardır (Suryaprasad vd., 2013). Nagargoje ve Shilvant, araç sürücüleri için uyku analizi yapma üzere görüntü işleme tekniklerini ve bir titreşim ölçer cihazı kullanmışlardır (Nagargoje ve Shilvant, 2015:78). Golgiyaz ve arkadaşları, sürücünün bakış yönü ve göz kırpma durumuna bakarak sürücünün ikaz edilip edilmeyeceğini bulmaya çalışmışlardır (Golgiyaz vd., 2014). Başka bir çalışmada Dwivedi ve arkadaşları, %78 doğruluk ile çalışan, özelleştirilmiş veri setleri kullanılan basit bir CNN tabanlı uykulu sürücü algılama sistemi önermişlerdir (Dwivedi vd., 2014: 995). Park ve meslektaşları çalışmalarında, RGB video üzerinde sürücü uyuşukluğunu tespit etmek için AlexNet, VGG-FaceNet ve FlowImageNet ağlarından oluşan derin bir CNN sunmaktadırlar (Park vd., 2016). Eğitim setinin boyutunu azaltmak için transfer öğrenimi kullanmışlardır. Algılama sistemini değerlendirirken %73'lük doğruluk ile önerilen sistemlerinin çalıştığını belirtmişlerdir. Bir başka çalışmada Reddy ve arkadaşları, gömülü bir sisteme yerleştirilebilecek daha hafif bir model üretmek için modellerini sıkıştırarak daha hızlı ve daha kesin bir gerçek zamanlı %89,5 doğruluk ile uyuşukluk algılama sistemi sunmuşlardır (Reddy vd., 2017).

Yan ve arkadaşları tarafından, yorgunluk tespiti için PERCLOS'a ve gri tonlamalı görüntü işlemeye dayalı gerçek zamanlı bir sistem geliştirilmiştir (Yan vd., 2016). Çalışmada gri tonlamalı görüntüler üzerinde sürücünün yüzünün yaklaşık konumu ve göz konumları kullanılmıştır. Deng ve Wu yaptıkları çalışmada, %92 doğruluk ile çalışan Dricare adlı bir sürücü yorgunluk sistemini önermişlerdir (Deng ve Wu, 2019). Bu çalışmada da diğer çalışmalara benzer olarak yüz ve göz bölgeleri ele alınmıştır. Mehta ve meslektaşları, görüntü

işleme teknikleri, makine öğrenmesi ve yüzdeki önemli noktalar olan Göz En Boy Oranını (EAR) ve Göz Kapatma Oranını (ECR) ) kullanarak, %84 doğruluk oranında çalışan Android tabanlı gerçek zamanlı sürücü yorgunluk sistemi geliştirmişlerdir (Mehta vd., 2019). Galarza ve arkadaşları doğruluk oranı %93,37 olan, uyuşukluğun tespiti için sürücünün gözlerinin davranışları, başın hareketleri ve esneme durumlarının dikkate alındığı bir yöntem geliştirmişlerdir (Galarza vd., 2018). Ravi ve arkadaşları çalışmalarında, Raspberry Pi üzerinde çalışan ve bulut tabanlı yönetim platformunun yanı sıra görüntü işleme tekniğine dayalı otomatik bir gerçek zamanlı sürücü yorgunluğu algılama sistemi önermektedir (Ravi vd., 2020). Ayrıca GPS alıcı modülü kullanarak konum detaylarının öğrenildiği belirtilmiştir. Qian ve meslekdaşları, yaptıkları çalışmada çoklu yüz özelliklerine sahip iki akışlı ağ modellerini kullanan bir sürücü yorgunluğu algılama algoritması önermişlerdir (Qian vd.,2019). Ağız ve göz bölgelerinin konumlandırılmasında çok görevli basamaklı konvolüsyonel sinir ağları (MTCNNs) kullanılmıştır. Ulusal Tsing Hua Üniversitesi sürücü uyuşukluk tespiti (NTHU-DD) veri setinde %97,06'lık bir doğruluk elde edildiği belirtilmiştir. Becerikli ve Savaş, geliştirdikleri çalışmalarında sürücü uyuşukluğunun tespiti için çok görevli bir Konvolüsyon Sinir Ağı (ConNN) modeli önermişlerdir (Becerikli ve Savaş, 2020). Sürücü yorgunluğu göz kapanma süresi/göz kapama yüzdesi (PERCLOS) ve esneme sıklığı/ağız sıklığı (FOM, frequency of mouth) hesaplanarak belirlenmektedir. Ayrıca önerilen modelin YawDD ve NthuDD veri setlerinde %98,81 doğrulukla çalıştığını belirtmişlerdir.

Bu çalışma diğer çalışmalardan farklı olarak, düşük maliyetli bir gömülü sistemi üzerinde yüz ve gözün yanı sıra esnemeyi de ele alarak doğruluk oranı yüksek sürücü yorgunluk tespiti yapan bir çözüm önerisi sunmaktadır. Önerilen çözüm ile trafik kazalarının minimum seviyeye düşürülerek, güvenli bir ulaşım sağlanması ve ulaşım boyunca insan hayatının korunması hedeflenmektedir. Bu amaçla, hareket halindeki araç içerisinde gerçek zamanlı olarak sürücünün yorgunluk durumunu tespit edebilen derin öğrenme bazlı araç içi taşınabilir bir sistem geliştirilmiştir. Geliştirilen sistem ile anlık olarak sürücü izlenmekte olup, yorgunluk veya uyku durumu tespitinde sürücü uyarılmaktadır. Böylece olası trafik kazalarının önlenmesinde önerilen sistemin yüksek oranda etkili olacağı öngörülmektedir.

## **2. YAPAY SINİR AĞLARI (YSA)**

Yapay sinir ağı, insan beyninin çalışma mantığı esas alınarak beynin öğrenme sürecinin matematiksel modellenmesi sonucu ortaya çıkmıştır (Karakuzu, 2015). Genel olarak YSA, insanlar tarafından oluşturulan ve beynin öğrenme, yeni bilgiler türetebilme, hatırlama ve benzeri temel işlevlerini otomatik olarak gerçekleştirebilmek amacı ile geliştirilen bilgisayar sistemleridir. YSA' nın çıkış tarihi çok eskilere dayanmaktadır. Geçmişte ysa yaklaşımları birçok farklı alanda var olan problemlere başarıyla uygulanmıştır. Ancak artan veri kümeleri ve hesaplama maliyetleri bu alandaki çalışmaları yavaşlatmıştır. Son zamanlarda donanımsal bakımdan yaşanan gelişmeler ile derin öğrenme yaklaşımları ortaya çıkmıştır. Buna bağlı olarak yapay sinir ağlarının gizli ve çok katmanlı yapıları kullanılmaya başlanmıştır.

### **2.1. Yapay Sinir Ağlarının Yapısı**

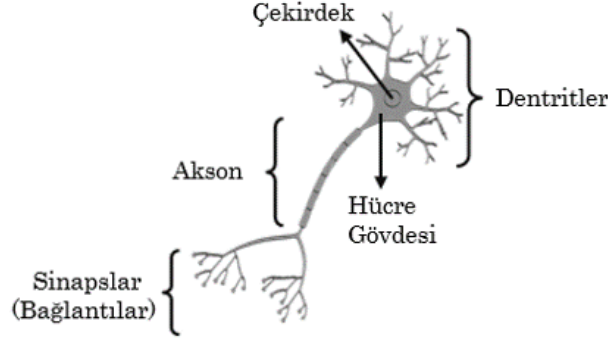
İnsan beyninden esinlenerek geliştirilen yapay sinir ağı, basit biyolojik sinir sisteminin çalışma şeklini taklit eder. Biyolojik sinir sisteminde yer alan nöronlar, sinyalleri bir nörondan diğer nörona ileten ağırlıklandırılmış bağlantılar (sinapslar) ile birbirine bağlı yapılardır. Sinir hücreleri arasındaki iletişim bu ağırlık bağlantıları yardımıyla gerçekleşir. Yapay sinir ağı, sinir hücrelerine gelen bilgileri toplayarak, belirlenen aktivasyon fonksiyonundan geçirip çıkışı üretir.

#### **2.1.1. Biyolojik Sinir Hücresinin Yapısı**

Bir insanın beyninde milyarlarca nöron ve bu nöronların birbirleri ile olan bağlantı sayısının trilyonlarca olduğu tahmin edilmektedir. Bu sinirler girdi bilgilerini duyu organlarından alırlar. Daha sonra taşıyıcı sinirler yardımı ile bu sinyaller işlenip bir sonraki sinire aktararak sinyallerin merkezi sinir sistemine kadar ulaşılması sağlanır. Merkezi sinir sistemi, kendisine ulaşan bu sinyalleri alır ve yorumlar. Yorumladıktan sonra tepki sinyallerini üretir. Üretilen bu sinyaller de tepkilerin gerçekleşeceği ilgili organlara iletilir. Böylece duyu organlarından gelen bilgiler, ilgili organlara işaretler sinir sistemi vasıtasıyla yollar (Çayıroğlu, 2015).

Biyolojik sinir sisteminin temelini oluşturan nöronlar, dört ana bölümden oluştuğu bilinmektedir. Bunlardan ilki olan dendritler, sinir hücresinin ucunda bulunan ve ağaç kökü görünümüne sahip bir yapıya sahiptirler. Dendritlerin amacı, tabii olduğu nöronlardan gelen sinyalleri çekirdeğe iletmektir. Çekirdek dendrit tarafından gelen sinyalleri bir araya

toplayarak aksona iletir. Toplanarak aksona iletilen bu sinyaller, akson tarafından işlenerek nöronun diğer ucunda bulunan bağlantılara gönderilir. Bağlantılar da, üretilen yeni sinyalleri diğer nöronlara iletir. Sinir hücresinin yapısı Şekil 2.1’ de yer almaktadır. Biyolojik sinir sisteminde yer alan bölümlerin yapay sinir ağı sistemindeki karşılığı Tablo 2.1’de yer almaktadır.



**Şekil 2.1.** Biyolojik sinir hücresinin yapısı

**Tablo 2.1** Biyolojik ve yapay sinir ağı sistemlerinin karşılıkları

Biyolojik Sinir Sistemi	Yapay Sinir Sistemi
Nöron	İşlemci eleman
Dentrit	Toplama fonksiyonu
Hücre gövdesi	Transfer fonksiyonu
Aksonlar	Yapay nöron çıkışı
Sinapslar	Ağırlıklar

**Kaynak:** (Koç vd., 2004)

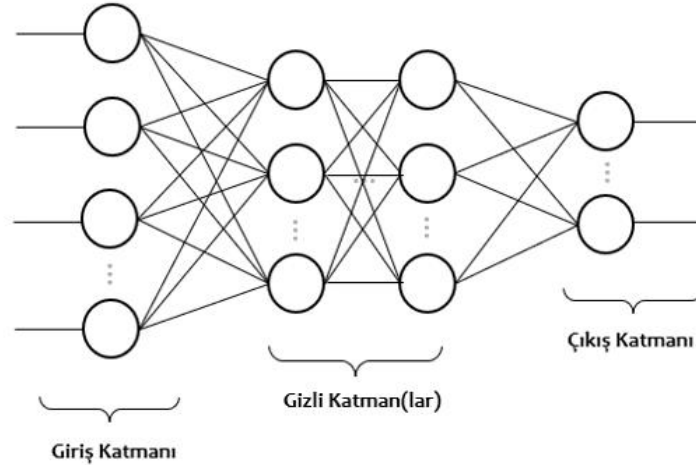
### 2.1.2. YSA Genel Yapısı ve Matematiksel Modeli

Yapay sinir ağları üç ana katmanda incelenir. İlki giriş katmanı, ikincisi gizli (ara) katmanı ve son olarak çıkış katmanıdır. İlk katman olan giriş katmanında, ağa girdi bilgisini veren girdi kümesi bulunur. Giriş katmanı ile çıkış katmanı arasında birden çok gizli katman bulunabilir. Bir katmandaki tüm nöronlar, bir önceki katmanda yer alan nörona çeşitli ağırlıklar ile bağlıdır. Giriş katmanında bulunan her bir veri ağırlık değeri ile çarpılıp, gizli

katmana iletilir. Giriş katmanından elde edilen toplam ağırlık değeri gizli katmandaki aktivasyon fonksiyonundan geçirilir ve çıkış katmanına gönderilir (Kutlu ve Batur, 2009).

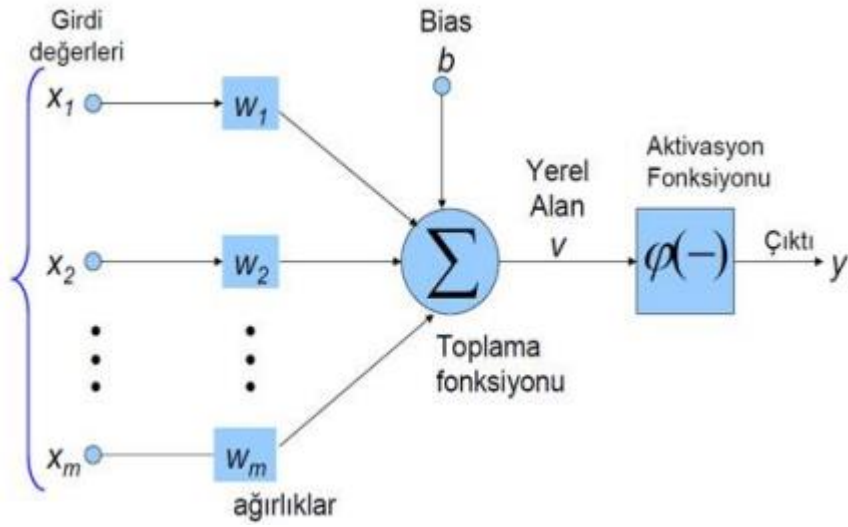
- A. Giriş Katmanı:** Yapay sinir ağına dış dünyadan girdilerin (verilerin) geldiği katmandır. Bu katmanda genelde girdiler herhangi bir işleme uğramadan bir sonraki katmana iletilmektedir (Yalçın, 2012).
- B. Gizli (Ara) Katman:** Giriş katmanından çıkan bilgiler bu katmana ya da katmanlara gelir ve gelen veriler bu katmanda işlenerek çıkış katmanına iletilir. Gizli katman sayısı, girdi ve çıktı katmanlarından bağımsız olarak ağdan ağa değişiklik gösterebilir (Yalçın, 2012).
- C. Çıkış Katmanı:** Gizli (ara) katmanlardan gelen bilgileri işleyerek, giriş katmanına gelen girdiye uygun ağın çıktılarını üreten katmandır. Bu katmanda elde edilen çıktılar dış dünyaya gönderilir.

Yapay sinir ağının genel yapısı Şekil 2.2’de yer almaktadır.



**Şekil 2.2.** Yapay sinir ağının genel yapısı

Yapay sinir ağı hücreleri de biyolojik sinir hücrelerine benzer yapıda oldukları için yapay nöronlar da aralarında bağ kurarak YSA’ yı oluştururlar. Bir yapay sinir ağı hücresi, biyolojik sinir ağı hücresi gibi bölümlerden oluşmaktadır (Şekil 2.3). Yani, yapay nöronların giriş sinyallerini aldıkları, bu sinyalleri toplayıp işledikleri ve çıktıları ilettikleri bölümleri bulunmaktadır. Bu bölümler girdiler, ağırlıklar, toplama fonksiyonu (birleştirme fonksiyonu), aktivasyon fonksiyonu, çıktılar olmak üzere beş bölümden oluşmaktadır.



Şekil 2.3. Yapay sinir ağının matematiksel modeli

**Kaynak:** (Keskenler, 2017)

- **Girdiler**

Katmanlardaki nöronlara gelen verilerdir. Girdiler, yapay sinir ağı hücresine direk olarak dış dünyadan veya diğer bir hücreden gelebilir. Bu girdilerden gelen veriler toplanmak üzere nöron çekirdeğine gönderilir.

- **Ağırlıklar**

Genel anlamda, hücreler arasındaki bağlantıların sayısal değerini belirtmektedir (Yalçın, 2015). Hücreye gelen bilgiler, geldikleri bağlantıların ağırlığıyla çarpılarak çekirdeğe iletilir. Bu sayede girdilerin üretilen çıktı üzerindeki etkisi ayarlanabilmektedir. Bu ağırlıkların değerleri pozitif, negatif veya sıfır olabilir (Çayıroğlu, 2015).

- **Toplama (Birleştirme) fonksiyonu**

Bir hücreye gelen net girişi hesaplamaktadır. En yaygın toplama fonksiyonu ağırlıklar toplamıdır.

- **Aktivasyon fonksiyonu**

Hücreye gelen net girdiyi işleyerek hücrenin bu girdiye karşılık üreteceği çıktıyı belirler. Bir yapay sinir ağındaki tüm hücrelerde aynı aktivasyon fonksiyonunun kullanılması gerekmez (Öztemel, 2006). Bu fonksiyonun seçimi ağın performansını önemli derecede etkiler. Kullanılacak fonksiyonun seçiminde dikkat edilmesi gereken bir başka durum, aktivasyon fonksiyonunun türevinin kolay hesaplanabilir olmasıdır. Aktivasyon fonksiyonu ağı doğrusal olmayan yapıyı öğrenmesini sağlayan bileşeni olup, genelde tek kutuplu (0 1),

çift kutuplu (-1 +1) ve doğrusal olarak seçilebilir (Uslu, 2016). Yapay sinir ağlarında genellikle, hiperbolik tanjant sigmoidal transfer (tansig), doğrusal transfer (purelin), logaritmik sigmoidal transfer (logsig), sert-limitli transfer (hardlim) ve doyunlaşan doğrusal transfer (satlin) aktivasyon fonksiyonları kullanılır (Karakuzu, 2015).

- **Çıktılar**

Aktivasyon fonksiyonu bölümünden çıkan değer, yapay sinir ağının çıktısı olarak dış dünyaya verilir ve gerektiğinde ağın içinde tekrar kullanılabilir.

## 2.2. Yapay Sinir Ağlarının Sınıflandırılması (Modelleri)

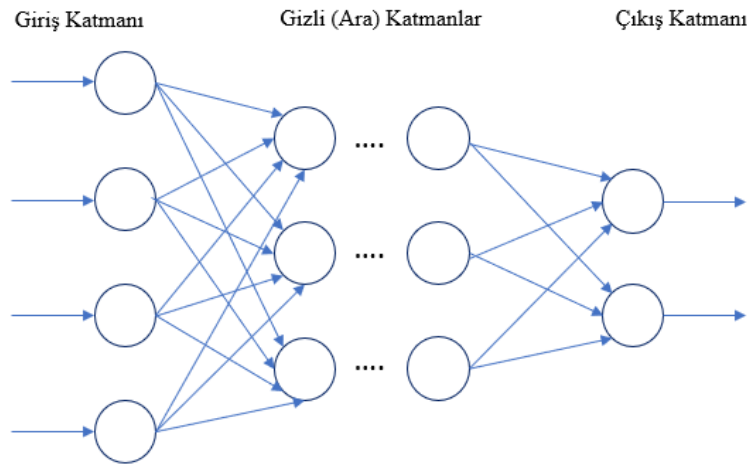
Yapay sinir ağları hücrelerin dizilimlerine, hücrelerin ağırlıklarının düzenlenmesi için yapılan hesaplamaların türüne ve zamanına göre üç farklı alanda incelenebilir (Kakıcı, 2017).

### 2.2.1. Yapılarına Göre Yapay Sinir Ağları

YSA, içerdiği nöronların birbiri ile olan bağlantı şekline göre ileri ve geri beslemeli olmak üzere ikiye ayrılmaktadır (Çayıroğlu, 2015).

#### 2.2.1.1. İleri Beslemeli Ağlar

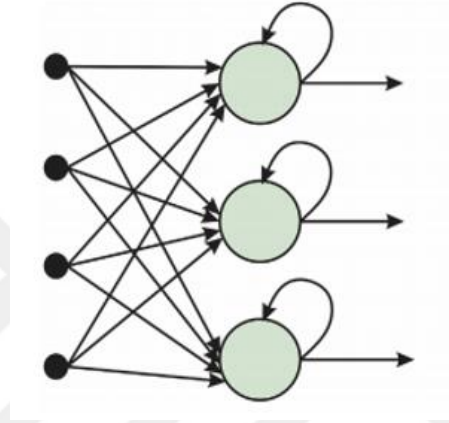
İleri beslemeli ağlar, yapay sinir ağlarının en basit ve en temel yapısıdır. Bu ağda bilgi akışı sadece ileri yönde olup, gizli (ara) katman/katmanlara ve çıkış katmanına doğru hareket eder (Şekil 2.4). Ağa gelen bilgiler giriş katmanına gelir ve bu katmanda hiçbir değişikliğe uğramadan gizli katmandaki hücrelere iletilir. Bilgi bu katman ve çıkış katmanında işlenerek ağın çıkışı belirlenir ve dış dünyaya aktarılır. Sistem belleksizdir. Ayrıca, bu ağ yapısına statik ağlarda denilmektedir (Aşkın vd.,2011).



Şekil 2.4. İleri beslemeli örnek ağ

### 2.2.1.2. Geri Beslemeli Ağlar

Geri beslemeli sinir ağları, en az bir hücrenin çıkışı kendisine giriş olarak verilerek geri beslemenin yapıldığı bir ağ yapısıdır (Şekil 2.5). Geri besleme, aynı katmandaki hücreler arasında olduğu gibi katmanlar arasındaki hücreler içerisinde de olabilir. Bu tarz ağlar dinamik hafızaya sahiptirler. Bu yapıdaki nöronların çıkışı yalnızca o anki girdi değerlerine değil, önceki girdi değerlerine de bağlıdır. Bundan dolayı, bu tarz ağ yapıları özellikle önceden tahmin uygulamaları için uygundur. Ayrıca geri beslemeli ağlar çeşitli tipteki zaman serilerinin tahmininde oldukça başarı sağlamışlardır (Aşkın vd.,2011).



Şekil 2.5. Geri beslemeli ağ yapısı örneği

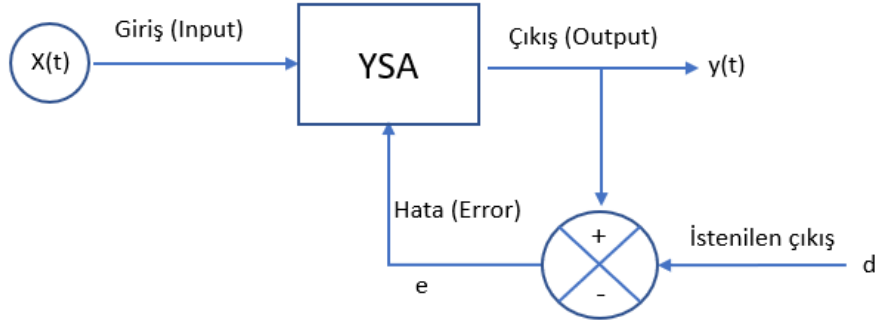
Kaynak: (Eliasy,2020)

### 2.2.2. Öğrenme Algoritmalarına Göre Yapay Sinir Ağları

Yapay sinir ağlarının girdi verilerine göre istenilen çıkışı üretebilmesi için ağın öğrenebilmesi gerekir. Bu öğrenme işleminde birden fazla yöntemi vardır. YSA, öğrenme algoritmalarına göre danışmanlı, danışmansız ve takviyeli öğrenme olmak üzere üç kısma ayrılır.

#### a) Danışmanlı (Denetimli) Öğrenme

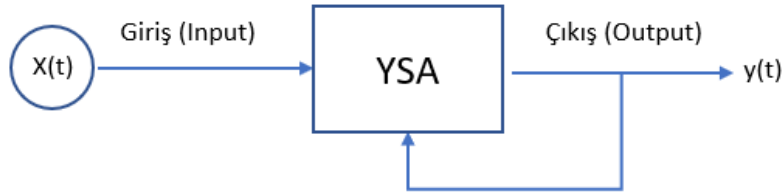
Danışmanlı öğrenmede, ağa verilen giriş değerleri ile birlikte çıktı değerleri de verilir. Ağ, verilen girdilere göre istenen çıktıları oluşturabilmek için kendi ağırlıklarını günceller. Ağın çıktıları ile beklenen çıktılar arasındaki hata hesaplanır ve ağın yeni ağırlıkları bu hata değerine göre düzenlenir. Hata değeri hesaplanırken ağın bütün çıktıları ile beklenen çıktıları arasındaki fark hesaplanır. Bu farka göre her hücreye düşen hata oranı tespit edilir. Ardından hücrelerin kendisine gelen ağırlıkları güncellenir (Şekil 2.6). Bu öğrenme algoritmasına, delta kuralı veya geri yayılım yöntemi örnek olarak verilebilir (Ataseven, 2013).



Şekil 2.6. Danışmanlı öğrenme modeli

### b) Danışmansız (Denetimsiz) Öğrenme

Danışmansız öğrenme metodunda, ağa sadece örnek girdiler verilmektedir. İstenen çıktı ile ilgili herhangi bir bilgi verilmez. Girişte verilen girdi verilerine göre ağ her birini kendi içerisinde sınıflandıracak şekilde kendi kurallarını oluşturur. Ağ bağlantı ağırlıklarını aynı özellikte olan grupları ayırabilecek şekilde düzenleyerek öğrenme işlemini tamamlar. Danışmansız öğrenme metodunun görselleştirilmiş hali Şekil 2.7’de yer almaktadır.



Şekil 2.7. Danışmansız öğrenme modeli

### c) Destekleyici (Takviyeli) Öğrenme

Pekiştirmeli öğrenme olarak bilinen bu metot, amaca yönelik ne yapılması gerektiğini öğrenen bir yaklaşımdır. Pekiştirmeli öğrenmede ajan adı verilen öğrenen makine, karşılaştığı durumlara bir tepki verir ve bunun karşılığında sayısal bir ödül sinyali alır. Ajan veya öğrenen olarak adlandırılan makine aldığı bu ödül puanını maksimuma çıkartmak için çalışır. Pekiştirmeli öğrenmenin ayırt edici özelliği deneme yanılma yöntemi ile çalışıyor olmasıdır (Sutton ve Barto, 2015).

### 2.2.3. Öğrenme Zamanına Göre Yapay Sinir Ağları

Yapay sinir ağları öğrenme zamanına göre statik ve dinamik öğrenme olmak üzere iki kısma ayrılır (Çayıroğlu, 2015).

#### • Statik öğrenme

Bu öğrenme mantığı ile çalışan bir yapay sinir ağı kullanılmadan önce eğitilir ve eğitim sona erdikten sonra ağ istenildiği gibi kullanılabilir. Ancak bu kullanım sırasında ağın üzerindeki ağırlıklarda herhangi bir değişiklik gerçekleşmez.

#### • Dinamik öğrenme

Dinamik öğrenme mantığı ise yapay sinir ağlarının çalıştığı süre boyunca öğrenme işleminin gerçekleştiği öngörülerek tasarlanmıştır. Ağ hem eğitim aşaması bittikten sonra hem de daha sonraki kullanımlarında çıkışların ağırlıklarını, istenilen çıkış olup olmamasına göre değiştirerek çalışmaya devam eder.

### 3. DERİN ÖĞRENME

Derin öğrenmenin, şu ana kadar literatürde yer alan kaynaklar arasında birçok farklı tanımı yapılmıştır. Şeker ve arkadaşları çalışmalarında, derin öğrenmeyi makine öğrenmesinin bir sınıfı olduğunu belirtmişlerdir (Şeker vd., 2017). Doğan ve Türkoğlu derin öğrenmeyi yapay sinir ağlarının ileri düzeyli bir yaklaşımı olarak tanımlamışlardır (Doğan ve Türkoğlu, 2018). Bir diğer kaynakta, derin öğrenme; yapay zekanın alt kümesi olan makine öğrenmesinin bir alt kümesidir. Yapay zeka, bilgisayarların insan davranışını taklit etmesini sağlayan tekniklere atıfta bulunan genel bir terimdir (Opperman, 2019). Kısaca, yapay sinir ağları temelli olan derin öğrenme, makine öğrenmesinin bir alt sınıfıdır. Derin öğrenme algoritmaları ise, mantıksal olarak, yapay sinir ağlarının yapısal bakımdan daha karmaşık hali olarak düşünülebilir. Derin öğrenme algoritmaları veri madenciliğindeki makine öğrenme algoritmalarından birkaç yönden ayrılır. Derin öğrenme algoritmalarında çok yüksek miktarda veri kullanılmaktadır. Aynı zamanda daha yüksek donanım ihtiyacı duymaktadırlar (Rende ve ark., 2016).

Derin öğrenme, özellik çıkarma ve dönüştürme için birçok doğrusal olmayan işlem birimi katmanını kullanır. Her ardışık katman, önceki katmandaki çıktıyı girdi olarak alır. Öğrenme algoritmaları denetimli (sınıflandırma gibi) veya denetimsiz (desen analizi gibi) olabilir. İlk başarılı derin sinir ağı uygulamasını Yann LeCun ve arkadaşları (LeCun vd., 1989) geliştirmişlerdir. Geliştirilen ağ başarılı çalışmış, ancak ağın eğitimi 3 güne yakın sürmüştür. Bu nedenle pratik olmadığı kanaatine varılmıştır. Bu ve benzeri çalışmalar başarılı olsa da hız ve güç bakımından yeterli donanımın olmaması ve giderek artan veri kümesi nedeniyle çalışmalara bir süre ara verilmiştir. Ancak son yıllarda gelişen hızlı ve güçlü donanımlar (Paralel İşlemcili GPU' lar) ve gelişen derin öğrenme mimarileri sayesinde yaygın bir şekilde kullanılmaya başlanmıştır (Şeker vd., 2017).

#### 3.1. Derin Öğrenme Mimarileri

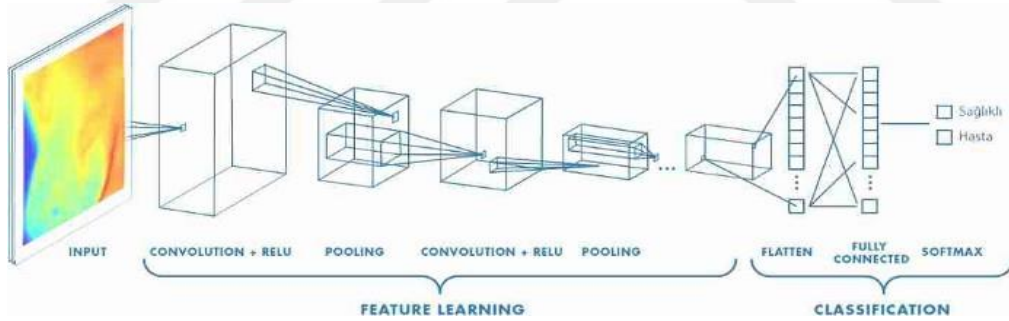
YSA' da artan katman sayıları nedeniyle, farklı birçok türde derin öğrenme mimarileri ortaya çıkmıştır. Bu bölümde en sık kullanılan ve bilinen 3 mimariden bahsedilecektir.

##### 3.1.1 Evrişimsel Sinir Ağları (CNN / ConvNet)

Evrişimsel sinir ağı mimarisi, derin öğrenme kavramına ait temel mimari olarak kabul edilir. Evrişimsel sinir ağı, bir girdi görüntüsünü alıp görüntüde yer alan çeşitli nesnelere birbirinden ayırt edebilen bir derin öğrenme algoritmasıdır (Okumuş, 2019). Konvolüsyonel

sinir ağı olarak da bilinir. Evrişimsel sinir ağı, yüksek boyutlu veriden düşük boyutlu özniteliklerin çıkarılmasını sağlamaktadır (Fırıldak ve Talu, 2019). Görüntü ve ses işleme, biyomedikal doğal dil işleme ve benzeri alanlarda, temel olarak görüntü sınıflandırma, benzerlikleri kümeleme veya nesne tanıma/ayırıştırma işlemleri yapmak için kullanılmaktadır. CNN, ileri yönlü bir yapay sinir ağı olup hayvanların görme merkezinden esinlenerek ortaya çıkmıştır. CNN'lerin en büyük avantajı, daha az oranda eğitime ve daha az parametreye ihtiyaç duymalarıdır (Gürbüz, 2018).

Evrişimsel sinir ağı, bir veya birden fazla evrişim (convolution), aktivasyon fonksiyonu (Relu) ve havuzlama (pooling) katmanlarını içermektedir. Son kısımda da tam bağlı katman ve sonrasında sınıflandırma katmanı yer almaktadır. Bu mimaride, giriş verileri alındıktan sonra sırasıyla her katmanda işlemler yapılarak eğitim süreci gerçekleştirilir. Sonucunda, doğru sonuç ile karşılaştırma yapılabilmesi için bir final çıktısı üretir. Üretilen sonuç ile istenen sonucun arasındaki fark kadar bir hata oluşur. Bu hatanın bütün ağırlıklara aktarılması için geriye yayılım algoritması kullanılır. Her bir iterasyon ile ağırlıklar güncellenerek, hatanın azaltılması sağlanır (İnik ve Ülger, 2017). Şekil 3.1'de evrişimsel sinir ağının genel yapısı gösterilmiştir.



**Şekil 3.1.** Evrişimsel/Konvolüsyonel sinir ağı yapısı

**Kaynak:** (Gürbüz, 2018)

### 3.1.1.1. Evrişimsel Sinir Ağlarını Oluşturan Katmanlar

Evrişimsel/Konvolüsyonel sinir ağı çok katmanlı yapıya sahiptir. Her katmanda ayrı bir işlem gerçekleşir ve veri bir sonraki katmana iletilir. Bu bölümde katmanların işlevleri ve tanımları yapılmıştır.

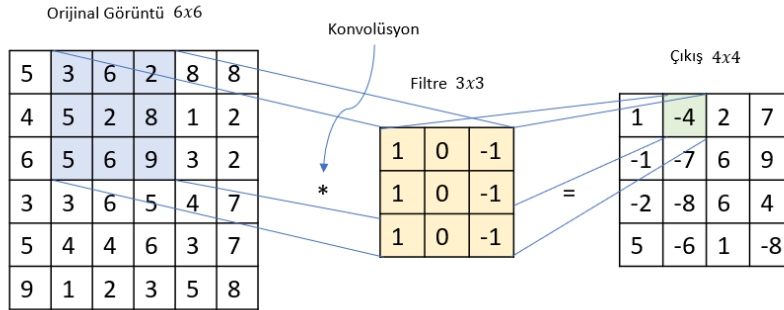
#### a) Giriş katmanı (Input Layer)

Giriş katmanı evrişimsel sinir ağı mimarisinin ilk katmanını oluşturmaktadır. Bu katmanda, veri hiç işlem görmemiş hali ile ağı girdi olarak verilmektedir. Bu kısımda verinin

boyutu modelin başarısı için önemlidir. Giriş görüntüsünün boyutu yüksek seçilirse hem yüksek bellek ihtiyacı hem eğitim süresi hem de görüntü başına düşen test süresi uzayabilir. Bunların yanı sıra başarısı artabilir. Tam tersi durumda yani giriş görüntüsünün boyutu düşük seçilirse bellek ihtiyacı azalır ve eğitim süresi kısalmır. Ancak tasarlanan ağın derinliği azalır ve performansı düşük olabilir. Bu nedenle görüntü analizinde hem ağ derinliği hem donanımsal hesaplama maliyeti ile birlikte ağ başarısı için uygun bir giriş görüntü boyutu seçilmelidir (İnik ve Ülger, 2017).

### b) Konvolüsyon katmanı (Convolution Layer)

Evrişimsel sinir ağının temelini oluşturmaktadır. Bu katman ayrıca dönüşüm katmanı olarak da bilinir. Buradaki işlem, tüm görüntü üzerinde belirli bir filtrenin dolaştırılmasına dayanmaktadır. Filtreler 3x3, 5x5, 11x11 gibi farklı boyutlarda olabilir. Örneğin, AlexNet'te 11x11, ZfNet'te 7x7 boyutunda filtreler kullanılırken, VggNet, GoogleNet, ResNet' te 5x5, 3x3, 2x2 boyutlarında filtreler kullanılmıştır (Doğan ve Türkoğlu, 2018). Kullanılan filtreler görüntüdeki özniteliklerin belirginleştirilmesini sağlar. Şekil 3.2'de, 6x6 boyutunda olan bir görüntü üzerine 3x3'lük filtre uygulanan bir örnek yer almaktadır.



Şekil 3.2. Konvolüsyon katmanında gerçekleşen bir işlem örneği

### c) Aktivasyon katmanı (Activation Layer)

Genellikle konvolüsyon katmanından sonra bu katman gelir. Doğrultucu birim katmanı olarakta bilinir. Burada gerçekleştirilen işlem, giriş verisinde yer alan negatif değerleri sıfıra çekmesidir. Aktivasyon fonksiyonu olarak farklı fonksiyonlarda kullanılabilir. Örneğin sigmoid, tanh, step vb. fonksiyonlar. Birden fazla aktivasyon fonksiyonu olmasına rağmen derin öğrenme mimarilerinde genellikle Relu fonksiyonu kullanılmaktadır. Şekil 3.3' de örnek bir görüntü üzerinde Relu katmanının etkisi görülmektedir.



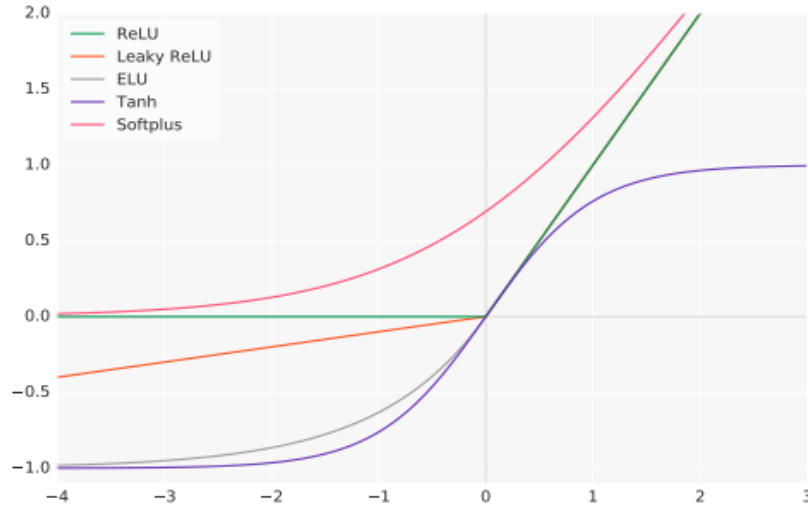
**Şekil 3.3.** Konvolüsyon ve ReLu katmanlarının görüntü üzerindeki etkisi

**Kaynak:** (İnik ve Ülger, 2017)

ReLU fonksiyonu doğrusal olmadığından, hatalara kolayca geri yayılım algoritması uygulanabilir ve birden çok nöron katmanına sahip sinir ağlarında kullanılabilir (Bozarık, 2020). ReLu  $[0, +\infty)$  aralığında değer alır ve negatif değerleri sıfıra dönüştürür. ReLu fonksiyonu Denklem (3.1)'de, birinci dereceden türevi ise Denklem (3.2.)'de yer almaktadır (Pedamonti, 2018). Grafik 3.4' de ReLu ve diğer aktivasyon fonksiyonları birlikte gösterilmektedir.

$$relu(x) = \max(0, x) \quad (3.1)$$

$$\frac{d}{dx} relu(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases} \quad (3.2)$$



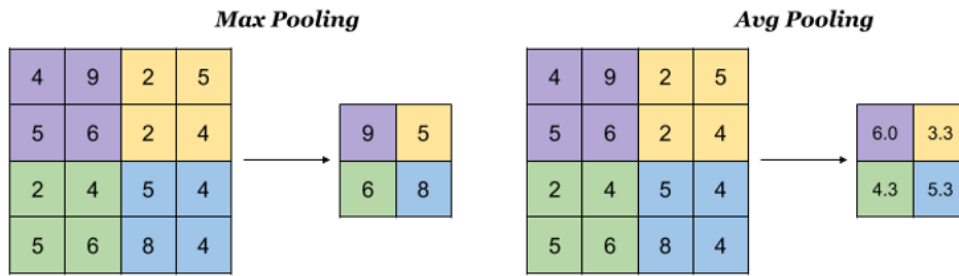
**Grafik 3.4.** ReLu ve diğer aktivasyon fonksiyonlarının grafikleri

**Kaynak:** (Ramachandran vd, 2017)

#### d) Havuzlama katmanı (Pooling Layer)

Aktivasyon katmanından sonra genelde havuzlama katmanı gelir. Bu katmandaki amaç, bir sonraki konvolüsyon katmanı için giriş boyutunu azaltmaktır. Boyutun azalması,

bilgi kaybına yol açar. Bu kaybın, ağ için sağladığı iki önemli durum vardır. İlki, bir sonraki ağ katmanlarında gerçekleştirilecek olan hesaplama yükünü azaltır. İkincisi ise sistemin ezberlemesini önler. Bu katmanda da konvolüsyon katmanında olduğu gibi belli başlı bazı filtreler uygulanır. Bu filtreler görüntü üzerinde  $n \times n$  boyutunda matris dolaştırılarak, görüntüdeki piksel değerlerinin maksimum değerleri (maksimum havuzlama) veya değerlerin ortalaması (ortalama havuzlama) alınarak işlem yapılır. Bu katmanda genellikle başarı oranı yüksek olduğu için maksimum havuzlama yöntemi kullanılır. Maksimum ve ortalama havuzlama işlemlerinin örneği Şekil 3.4’ de yer almaktadır.



**Şekil 3.4.** Maksimum ve ortalama havuzlama işlemlerinin mantığı

**Kaynak:** (Pokhrel, 2019)

#### e) Tam bağlantılı katman (Full-Connected Layer)

Bu katmanda, kendinden önce yer alan katmanın tüm alanlarına bağlıdır ve o katmandan gelen verileri tek boyutlu matrise indirir. Tam bağlantı katman sayısı, farklı mimarilerde değişiklik gösterebilir.

#### f) Dropout katmanı

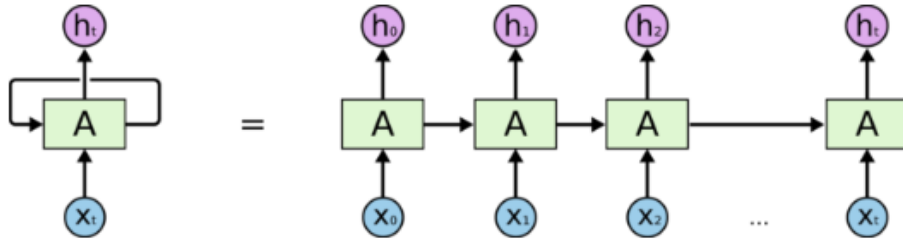
Derin öğrenme ağlarında, aşırı öğrenmeyi bir başka deyişle ezberlemeyi önlemek için bu katman kullanılır. Bu katmanda gerçekleşen temel mantık ağın bazı düğümlerinin ortadan kaldırılmasıdır (Tan ve Aydın, 2019).

#### g) Sınıflandırma katmanı (Classification Layer)

Mimarilerin bu katmanında sınıflandırma işlemi gerçekleştirilir ve tam bağlantı katmanından sonra gelir. Burada, sınıflandırılması yapılacak nesne sayısı kadar sonuç üretilir. Örneğin 7 farklı objenin sınıflandırılması yapılacak olursa, sınıflandırma katmanının çıkış değeri 7 olmalıdır. Bu katmanda farklı sınıflandırıcılar kullanılabilir. Genellikle, başarı oranı yüksek olduğu için softmax fonksiyonu kullanılır. Sonuç değeri 7 farklı nesne için 0-1 aralığında değer alacaktır ve değeri en yüksek olan sınıf tahmin değeri olacaktır (Tan ve Aydın, 2019)

### 3.1.2 Tekrarlayan Sinir Ağları (RNN, Recurrent Neural Network)

RNN' ler, ardışık bilgileri kullanan, gizli (ara) katman çıkışını tekrar aynı katmana girdi olarak gönderebilen bir derin öğrenme mimarisidir. RNN mimarisinin, çalışma yapısı Şekil 3.5' de yer almaktadır. Görselde yer alan  $X$ ,  $h$  ve  $A$  değerleri sırasıyla ağın girdi değerini, ağın çıktı değerini ve gizli katmanı temsil etmektedir. Gizli katmandan çıkan değer tekrar gizli katmana gelerek, kendi kendini çağıran fonksiyon işlevi görmektedir (Kayaalp ve Süzen, 2018)

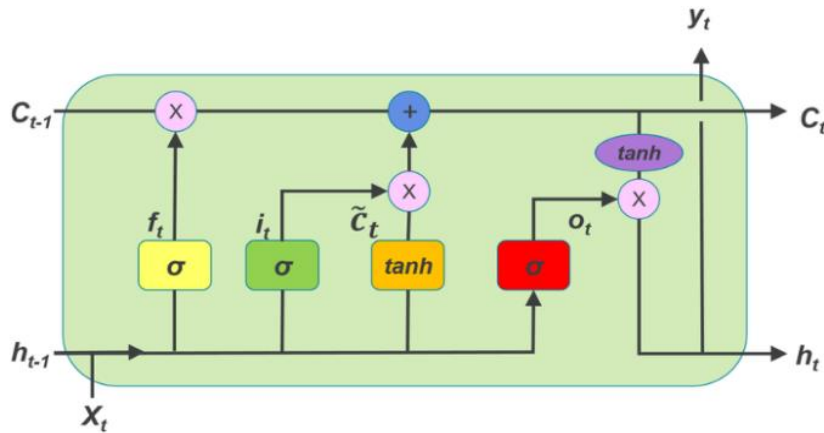


Şekil 3.5. RNN mimarisi çalışma yapısı

Kaynak: (Mittal, 2019)

### 3.1.3 Uzun Kısa Dönemli Bellek (LSTM, Long-Short Term Memory)

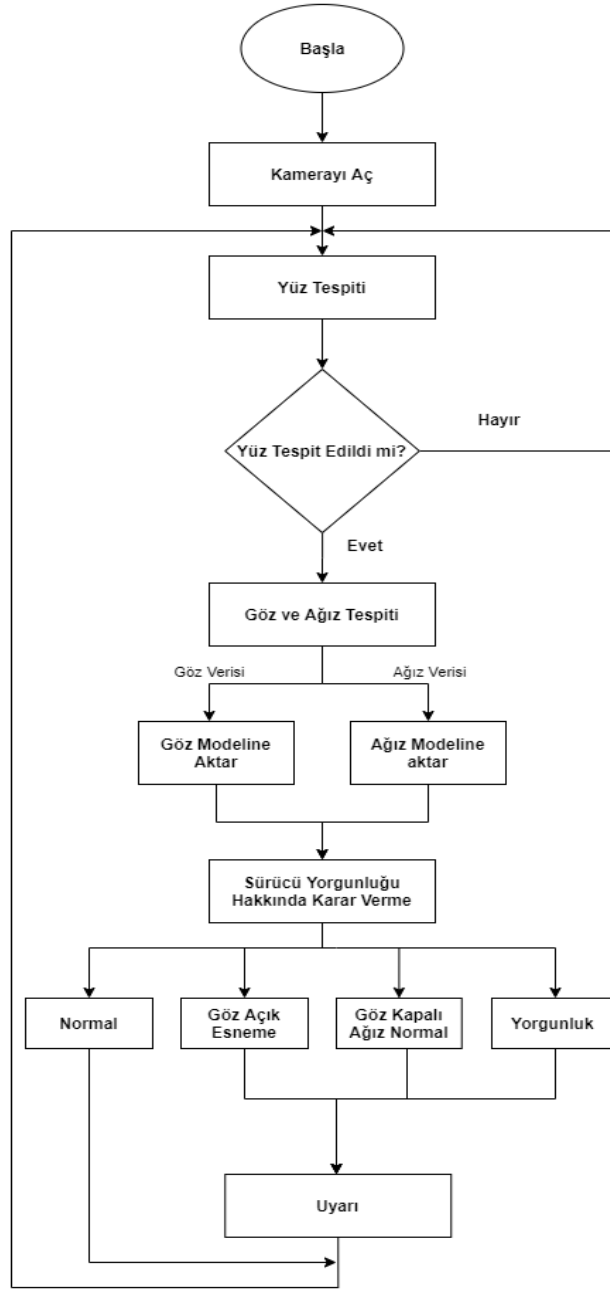
(Hochreiter ve Schmidhuber, 1997), tarafından uzun vadeli bağımlılıkları öğrenebilen ve RNN' nin gelişmiş bir türü olarak tanıtılmıştır. LSTM yapısı, bilginin hücre içine ya da dışına akışını denetleyen kapılardan oluşur. Girdi olarak verilen kapı, yeni gelen bilginin hafızaya ne zaman girmesi gerektiğini denetler. Unutma/hafıza kapısı mevcut olan bilginin unutulmasını, yeni verilerin ise hatırlanmasını sağlayarak denetler. Çıkış bölümü, hücredeki bilginin hücrenin çıkışında ne zaman kullanılacağını denetler (Gürbüz, 2018). Bahsedilen LSTM yapısı Şekil 3.6' da görülmektedir.



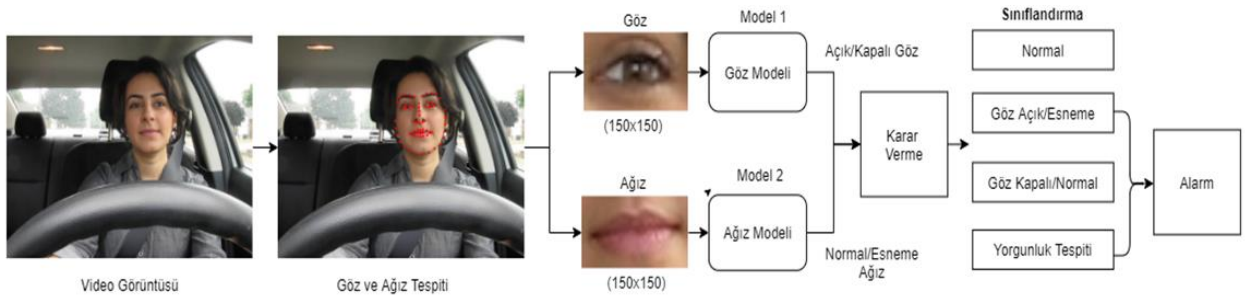
Şekil 3.6. LSTM çalışma yapısı

Kaynak: (Gürbüz, 2018)





Şekil 4.2. Sistemin akış diyagramı



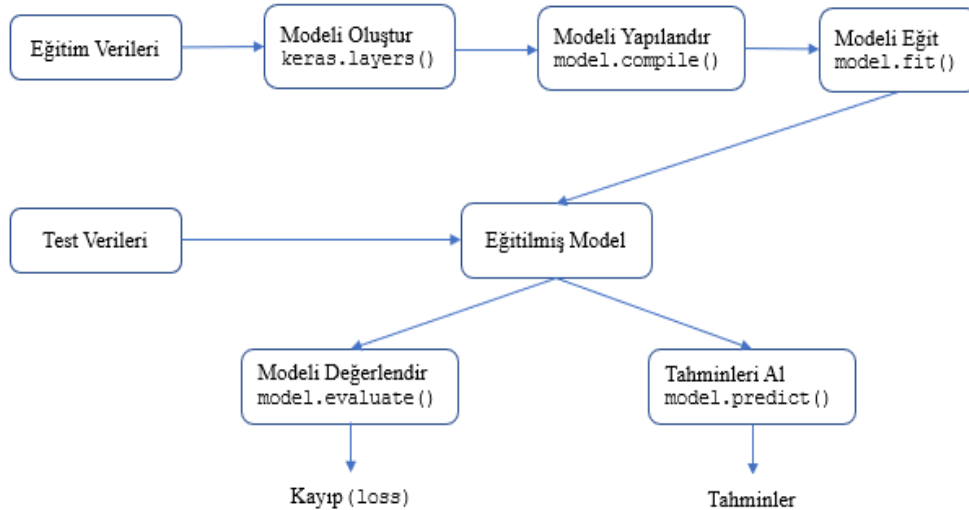
Şekil 4.3. Sistemin çalışma yapısı

## 4.2. Yazılım Altyapısı

Çalışmada, hem YawDD veri setinden model oluşturmak hem de sistem çalıştığı anda anlık olarak sürücünün yüz, göz ve ağız bölgelerini tespit etmek için dlib<sup>1</sup> kütüphanesi kullanılmıştır. Model eğitiminde ise Keras derin öğrenme kütüphanesi ve VGG-16 derin öğrenme ağı kullanılarak iki model oluşturulmuştur. Modellerin biri göz bölgesini kullanarak gözün açık olup olmadığını tespit ederken, diğer model ağız bölgesini kullanarak esnemenin olup olmadığını tespit etmektedir. Modellerin eğitiminde YawDD veri seti kullanılmıştır. Bu bölümde derin öğrenme kütüphanesi Keras, derin öğrenme ağı VGG-16 ve model eğitiminde kullanılan YawDD veri setinden bahsedilecektir.

### 4.2.1. Keras Derin Öğrenme Kütüphanesi

Keras, python programlama dili ile yazılmış, arka planda Tensorflow ve Theano kütüphaneleri üzerinde çalışabilen üst düzey bir API (Application Programming Interface), yani uygulama programlama arayüzüdür. Keras, basit ve işlevsel olmasıyla birlikte güçlü bir yapıya sahiptir. Bu sayede kolay ve hızlı model oluşumunu sağlar ve oluşturulan modeller GPU ve CPU üzerinde sorunsuz çalıştırılabilir. Ayrıca CNN ve RNN derin öğrenme mimarilerini destekler (Gupta ve Murzova, 2017). Keras kütüphanesinin çalışma akışı Şekil 4.4' de gösterilmiştir.



Şekil 4.4. Keras kütüphanesinin çalışma akışı

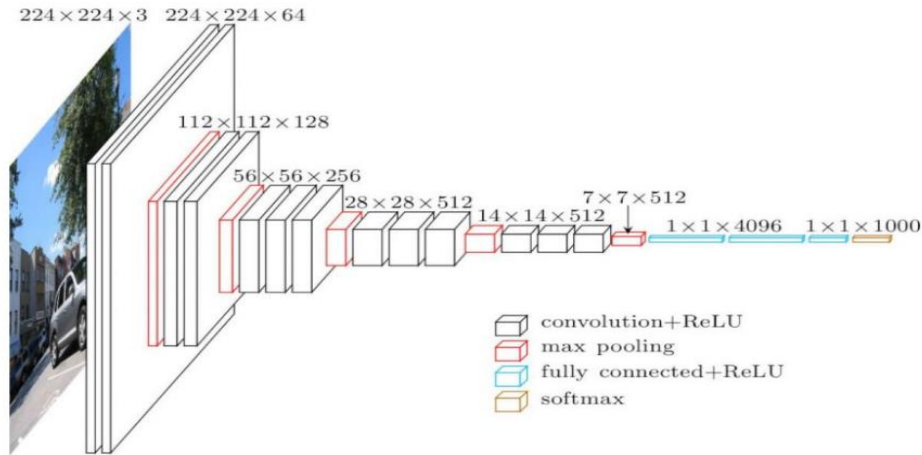
**Kaynak:** (Gupta ve Murzova, 2017)

<sup>1</sup> <http://dlib.net/>

Keras kütüphanesi ile model oluşturmanın iki farklı yöntemi vardır. İlki ve en sık kullanılanı, sıralı katmanlar oluşturmamızı sağlayan “Sequential” yöntemidir. Bu fonksiyonu kullanarak oluşturulan YSA, sadece birbirine sırası ile bağlı olan katmanlardan oluşabilir. Bir diğer yöntem “Functional API” yöntemidir. Sequential yönteminden farklı olarak, katmanların ardışık bağlanma zorunluluğu yoktur. Bu yöntem, daha karmaşık yapılar için kullanılır (Chollet, 2020).

#### 4.2.2. VGG-16 Ağ Modeli

VGG-16, Oxford Üniversitesi'nden K. Simonyan ve A. Zisserman tarafından "Büyük Ölçekli Görüntü Tanıma için Çok Derin Evrişimli Ağlar" makalesinde önerilen evrişimli bir sinir ağı modelidir (Simonyan ve Zisserman, 2014). VGG-16, basit ağ modeli olup, diğer ağlardan farkı evrişim katmanlarını ikili veya üçlü kullanmasıdır. VGG-16 mimarisinin yer aldığı Şekil 4.5’ de görüldüğü üzere, 16 katmanlı olan bu mimarinin son katmanı olan tam bağlantı (Full Connected) katmanında  $7 \times 7 \times 512 = 4096$  nöronlu bir öznitelik vektörü elde edilir. Bu katmanda aktivasyon fonksiyonu kullanılarak 1000 sınıflı başarımlı hesaplanabilir. Ayrıca, milyonlarca parametre hesabı yapılmakta ve matrislerin boyutları azaltılırken derinlik değeri artmaktadır.



Şekil 4.5. VGG-16 mimarisi

Kaynak: (Hassan, 2018)

#### 4.2.3. YawDD Veri Seti

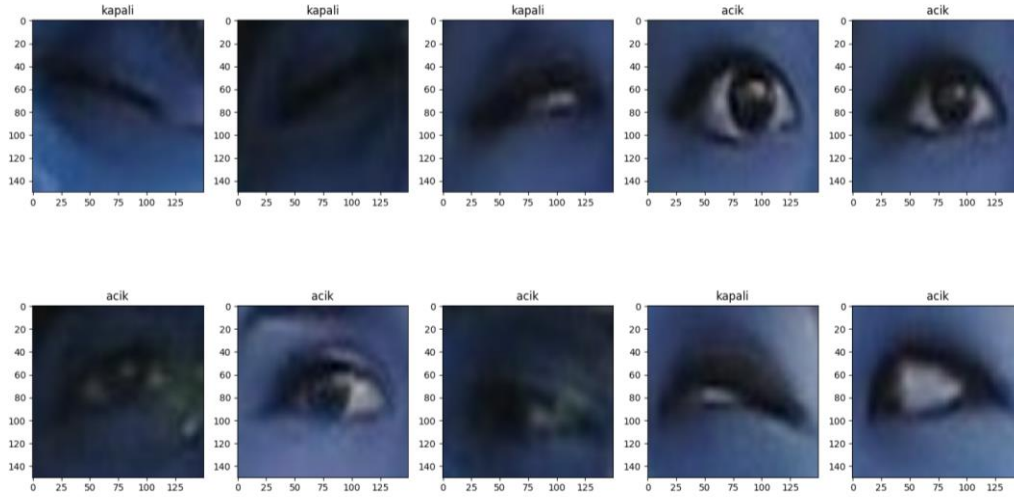
Modellerin eğitiminde YawDD veri seti kullanılmıştır. YawDD veri seti, gerçek bir arabadaki sürücülerin (erkekler ve kadınlar, gözlüklü / güneş gözlüklü ve gözlüksüz, farklı etnik kökenler) konuşma, şarkı söyleme, sessiz ve esneme videolarının araç içi kamera ile

kaydedilmiş veri setidir. Öncelikle esneme tespiti için algoritmalar, modeller geliştirmek ve test etmek, ayrıca yüzü ve ağız tanımak ve izlemek için kullanılabilir. Videolar doğal ve değişken ışık koşullarında çekilmiştir (Abtahi vd., 2020). Veri setinde yer alan esneme videolarından görüntüler elde edilip, göz ve ağız bölgeleri ayrıştırılarak özel bir veri seti oluşturulmuştur. Eğitim veri setlerine ait bilgiler Tablo 4.1’de yer almaktadır.

**Tablo 4.1.** Eğitim veri seti

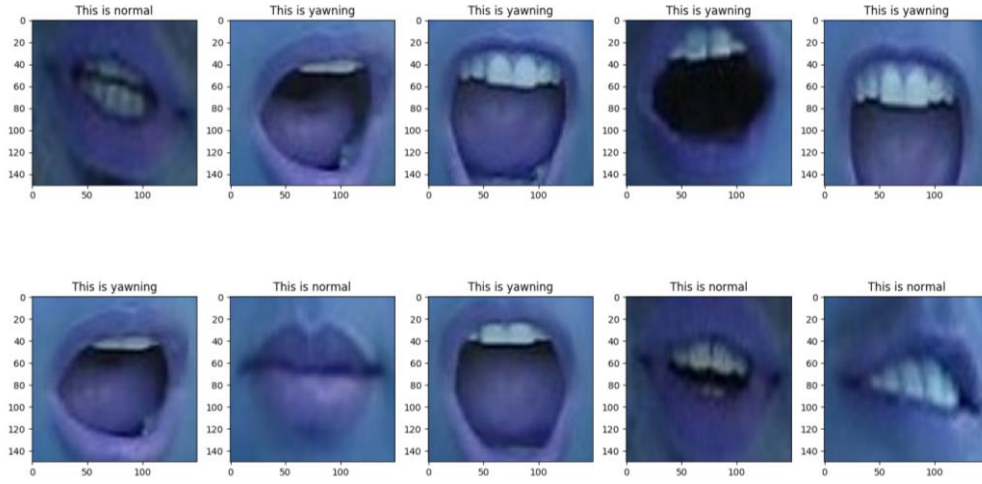
Modeller	Kadın		Erkek		Toplam Kişi Sayısı
	Gözlüklü	Gözlüksüz	Gözlüklü	Gözlüksüz	
Göz Modeli	4	6	8	5	23
Ağız Modeli	10		13		23

Elde edilen görüntüler modellere girdi olarak verilmeden önce OpenCV<sup>2</sup> kütüphanesi kullanılarak ön işlemden geçirilmektedir. Görüntüler 150x150 oranında yeniden boyutlandırılıp ardından modellere girdi olarak verilmektedir. Göz ve ağız modellerinin test verisinde kullanılan örnek görselleri Şekil 4.6 ve Şekil 4.7’de yer almaktadır.



**Şekil 4.6.** Göz modeli örnek veri seti

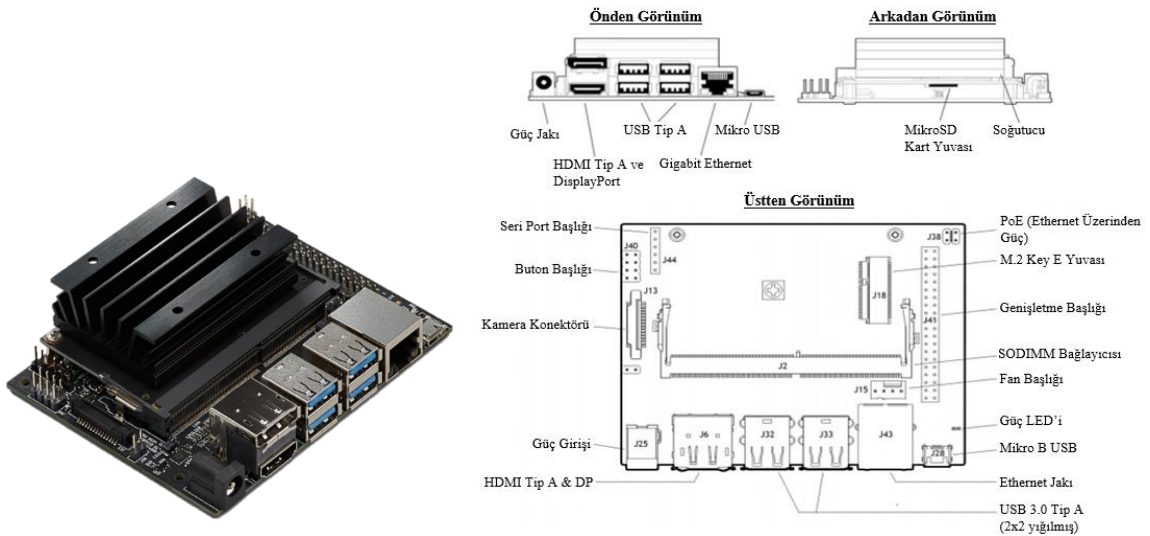
<sup>2</sup> <https://opencv.org/>



Şekil 4.7. Ağız modeli örnek veri seti

### 4.3. Donanım Altyapısı

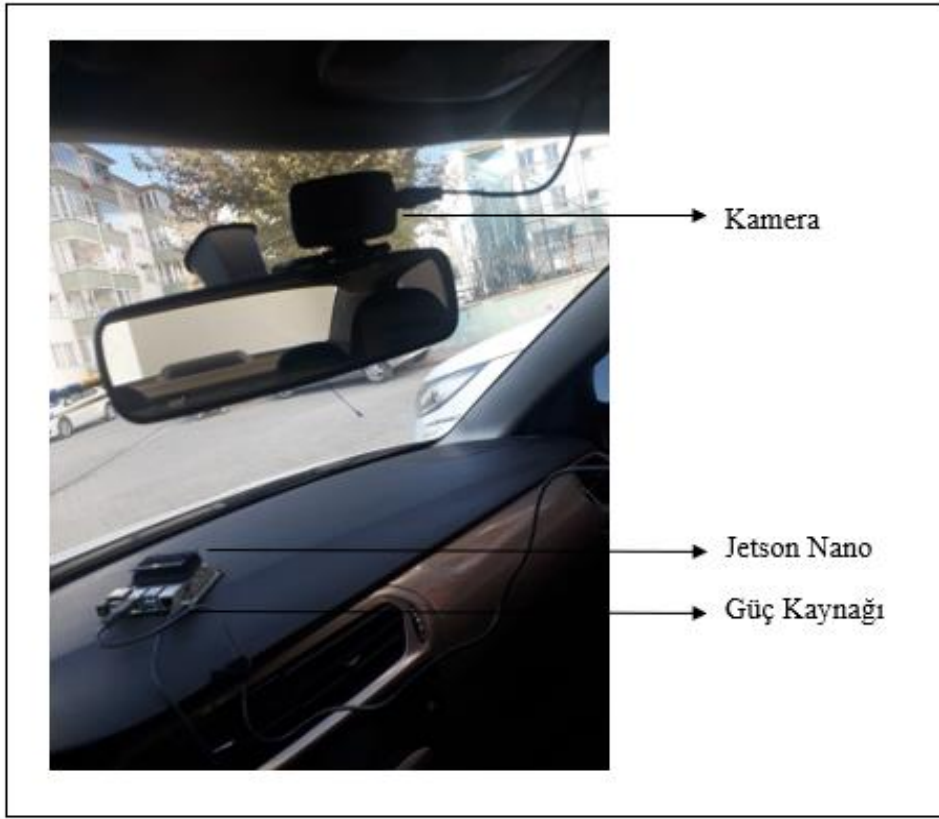
Derin öğrenme bazlı çalışan sistemde modellerin eğitimi Intel® Core™ i7-7500 CPU, Nvidia GeForce 950M GPU, 16GB RAM donanımlarına sahip bir bilgisayar üzerinde gerçekleştirilmiştir. Modeller hem bilgisayar hem de gömülü cihaz üzerinde test edilmiştir. Tasarlanan sistemde kullanılan Nvidia Jetson Nano gömülü cihazı (Şekil 4.8), 4 GB 64 bit LPDDR4 25.6 GB/s hafızasına, 4-çekirdekli ARM A57 CPU' ya ve 128 CUDA çekirdekli NVIDIA Maxwell™ GPU' ya sahip, yapay zeka algoritmalarının çalıştırılabilmesi için geliştirilen bir geliştirici kitidir (Cardinal, 2019). Nvidia Jetson Nano ile kameradan gerçek zamanlı olarak gelen görüntüler, görüntü işleme yöntemleri ile ön işlemden geçirilerek derin öğrenme bazlı modellere girdi olarak verilmektedir.



Şekil 4.8. Nvidia Jetson Nano gömülü cihazı

**Kaynak:** (Cardinal, 2019)

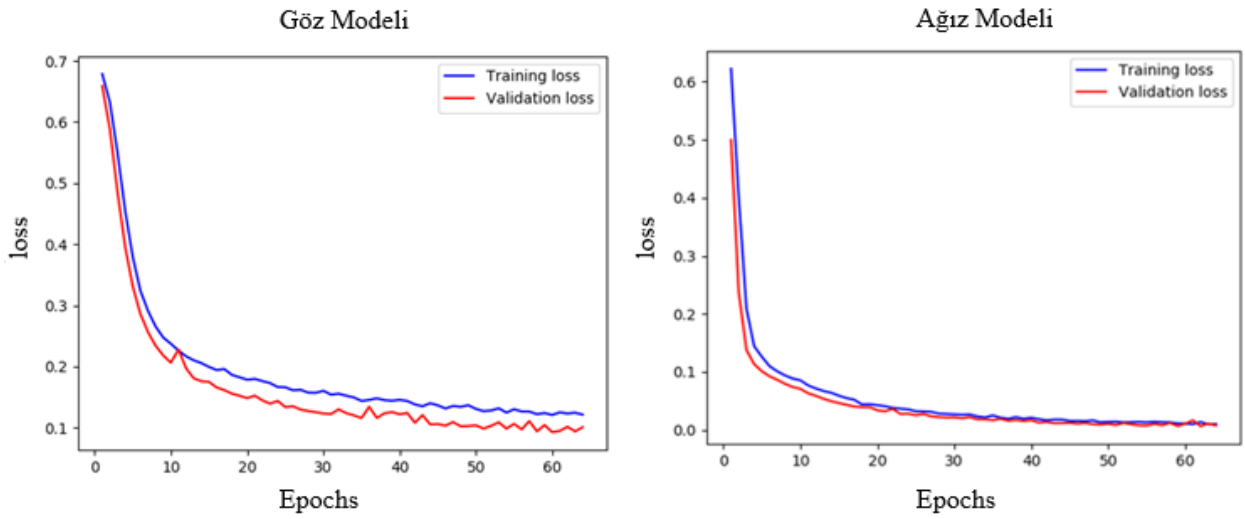
Modellerin çıkışından alınan bilgiler doğrultusunda hem sesli hem de görsel olarak alarm sistemi devreye girmektedir. Alarm sisteminde görsel uyarı için RGB led, sesli uyarı için de zil (buzzer) kullanılmıştır. Çalışmada tasarlanan sistemi, taşınabilir hale getirmek için araçların çakmaklıklarından kullanılan donanımlar ile güç sağlanmıştır. Araç çakmaklıkları genellikle 12-24V (DC) gerilim oluşturur. Çalışmada kullanılan Nvidia Jetson Nano cihazı 5V gerilim ile çalışmaktadır. Bu nedenle girişi 12-24V (DC), çıkışı 5V olan bir dönüştürücü kullanılarak sistemin güç kaynağı sağlanmıştır. Alarm için kullanılan buzzer ve ledler yine aynı gerilim kaynağı üzerinden beslenmektedir. Tasarlanan sistemin araç içerisindeki görüntüsü Şekil 4.9'da gösterilmektedir.



**Şekil 4.9.** Sistemin araç içi görüntüsü

## 5. DENEYSEL SONUÇLAR VE TARTIŞMA

Araç içinde sürücülerin yorgunluk durumlarının tespiti için tasarlanan sistem sürücünün yüz, göz ve ağız bölgelerini sürekli olarak takip etmektedir. Göz ve ağız bölgeleri tespit edildikten sonra iki ayrı modele gönderilerek yorgunluk tespiti yapılmaktadır. Böylece göz tespiti yapılamayan durumlarda ağız bölgesinden esnemenin olup olmadığı bağımsız olarak tespit edilebilmektedir. Modellerin eğitim (training) ve doğrulama (validation) hata kaybı grafikleri Grafik 5.1’de yer almaktadır.



**Grafik 5.1.** Göz ve ağız modelleri eğitim ve doğrulama hata kaybı grafikleri

Sistemde ağız ve göz olarak kullanılan iki model ayrı ayrı test edilmiştir. Göz modelinin performansı için Denklem (5.1), ağız modelinin performansı için Denklem (5.2) ve sistem genel performansı için ise Denklem (5.3) kullanılmıştır. Denklemlerde kullanılan  $G_b$  ve  $A_b$  değişkenleri sırasıyla göz ve ağız modellerinde başarılı test veri sayısını göstermektedir.  $G_t$  ve  $A_t$  değişkenleri ise sırasıyla göz ve ağız modellerinde kullanılan toplam test veri sayısını temsil etmektedir.

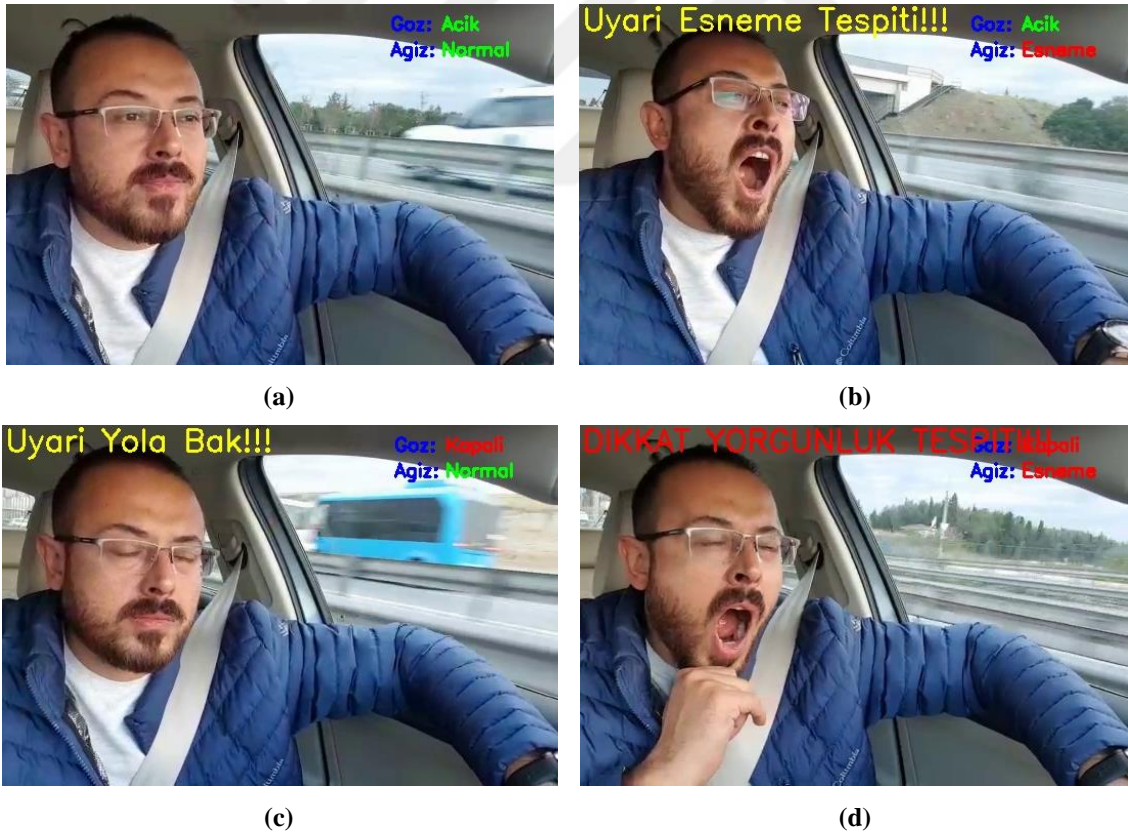
$$\text{Göz Modeli Doğruluk Oranı} = \frac{G_b}{G_t} \quad (5.1)$$

$$\text{Ağız Modeli Doğruluk Oranı} = \frac{A_b}{A_t} \quad (5.2)$$

$$\text{Sistem Doğruluk Oranı} = \left( \frac{G_b}{G_t} + \frac{A_b}{A_t} \right) / 2 \quad (5.3)$$

Test veri setinde göz modeli için 674 video görüntüsü (frame), ağız modeli için ise 613 video görüntüsü kullanılmıştır. Göz modelinin doğruluk oranı %93,6 iken ağız modelinin doğruluk oranı %94,5 olarak bulunmuştur. Bu doğrultuda ortalama toplam doğruluk oranı %94,05 olarak hesaplanmıştır. Sistem hız açısından değerlendirildiğinde bilgisayar üzerinde ortalama 22 fps, Nvidia Jetson Nano gömülü sisteminde ise ortalama 6 fps olarak çalışmaktadır.

Yorgunluk belirtileri gözlemlendiği anda sistem üzerinden uyarı devreye girmektedir. Sistem, dört farklı sınıflandırma durumu için çalışmaktadır. İlk durum olarak gözün açık ve ağız normal olması durumu ele alınmış olup Şekil 5.1a'da gösterilmiştir. İkinci durum, gözün açık esnemenin gerçekleşmesi durumu ele alınmış ve Şekil 5.1b'de yer verilmiştir. Üçüncü durum olarak da esnemenin olmadığı ve gözün belirli bir süre kapalı olması ele alınmış olup Şekil 5.1c'de gösterilmiştir. Dördüncü ve son durum, aynı anda hem esnemenin olması hem de gözün kapalı olması durumu ele alınmış olup Şekil 5.1d'de yer verilmiştir.



**Şekil 5.1.** Sürücü yorgunluk tespitinin test edilmesi: normal durum (a), göz açık esneme durumu (b), gözün kapalı olma durumu (c), yorgunluk tespiti (d).

## 6. SONUÇ VE TARTIŞMA

Bu tez çalışmasında önerilen sistem sonucunda, Nvidia Jetson Nano gömülü cihazı üzerinde sürücü yorgunluk tespiti başarılı bir şekilde çalıştırılmıştır. Geliştirilen sistem kapsamında göz ve ağız durumlarının tespitinde VGG-16 ağ mimarisi ve Keras derin öğrenme kütüphanesi kullanılmıştır. Bu tez çalışmasında geliştirilen sistemde sürücünün esneme ve gözünün açık/kapalı durumu için iki ayrı model ve dört sınıf/durum bulunmaktadır. Modellerin eğitiminde YawDD veri seti kullanılmıştır. Sürücünün göz ve ağız bölgeleri sürekli olarak izlenerek, "göz açık / normal ağız", "göz açık / esneme", "göz kapalı / ağız normal", "göz kapalı / esneme" olmak üzere dört farklı durum ile video görüntüleri sınıflandırılmaktadır.

Derin öğrenme göz modelinin doğruluk oranı %93,6 iken ağız modelinin doğruluk oranı ise %94,5 olarak bulunmuştur. Bu iki modelin doğruluk oranlarına göre ortalama doğruluk oranı %94,05 olarak hesaplanmıştır. Sistem hız açısından değerlendirildiğinde, bilgisayarda 22 fps ve Nvidia Jetson Nano gömülü cihazında da ortalama 6 fps'dir. Bu sistemde dikkat edilmesi gereken en önemli nokta kameranın araç içerisindeki konumudur. Kameranın konumu, sistemin doğru çalışma oranı üzerinde büyük bir etkiye sahiptir.

İleride yapılacak çalışmalarda modelin boyutu daha da küçültülerek gömülü sistem üzerinde çalışma hızı daha da artırılabilir. Farklı göz ve ağız modelleri (göz kırpma, esneme sıklığı vb.) ile sistemin performansının geliştirilmesi de gelecek çalışmalar arasında yapılabilir.

## KAYNAKÇA

- Abtahi, S., Omidyeganeh, M., Shirmohammadi, S. & Hariri, B.** (2020). *YawDD: Yawning Detection Dataset*. [Erişim: 28.12.2020, <https://iee-dataport.org/open-access/yawdd-yawning-detection-dataset> ]
- Ataseven, B.** (2013). *Yapay sinir Ağları ile Öngörü Modellemesi*. İstanbul Kültür Üniversitesi, Öneri.C.10.S.39.
- Bozarık, E.** (2020). *Sinir Ağları ve derin Öğrenme – Aktivasyon Fonksiyonları* [Erişim: 28.12.2020, <https://medium.com/smlr-center/sinir-a%C4%9Flar%C4%B1-ve-derin-%C3%B6%C4%9Frenme-xiv-aktivasyon-fonksiyonlar%C4%B1-c125885ee5f7> ]
- Cardinal, D.** (2019). *Hands On With Nvidia's New Jetson Nano*. [Erişim: 28.12.2020, <https://www.extremetech.com/computing/288153-hands-on-with-nvidias-new-jetson-nano> ]
- Chollet, F.** (2020). *Developer guides*. [Erişim: 28.12.2020, <https://keras.io/guides/> ]
- Çayıroğlu, İ.** (2015). *İleri Algoritma Analizi-5 Yapay Sinir Ağı Ders Notları*, Karabük Üniversitesi.
- Deng, W. & Wu, R.** (2019). *Real-Time Driver-Drowsiness Detection system Using Facial Features*. IEEE ACCESS, Volume 7.
- Doğan F., & Türkoğlu İ.,** (2018) *Derin Öğrenme Algoritmalarının Yaprak Sınıflandırma Başarımlarının Karşılaştırılması*, Sakarya University Journal Of Computer And Information Sciences Vol. 1, ID. SAUCIS-1-2018
- Dwivedi, K., Biswaranjan, K. & Sethi, A.** (2014). *Drowsy driver detection using representation learning*. In 2014 IEEE International Advance Computing Conference (IACC), s:995–999.
- Eliasy, A. & Przychodzen, J.** (2020). *The role of AI in capital structure to enhance corporate funding strategies*, Elsevier Inc., Volume 6, 100017.
- Fırıldak, K. & Talu, M. F.** (2019). *Evrişimsel Sinir Ağlarında Kullanılan Transfer Öğrenme Yaklaşımlarının İncelenmesi*. Anatolian Journal of Computer Science, Volume:4, No:2, s:88-95
- Galarza, E. E., Egas, F. D., Silva, F. M. & Velasco, P. M.** (2018). *Real Time Driver Drowsiness Detection Based on Driver's Face Image Behavior Using a System of Human Computer Interaction Implemented in a Smartphone*. Proceedings of the International Conference on Information Technology & Systems (ICITS) pp 563-572.
- Girit A.** (2014). *Drowsy Driver Detection Using Image Processing*, Yayınlanmamış Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi Fen Bilimleri Enstitüsü.
- Golgiyaz S., Kocamaz A. F., & Okumuş F.** (2014). *Video Tabanlı Uykulu Sürücü Algılama Sistemi*. Eleco-2014 Elektrik Elektronik Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu

**Gupta, V. & Murzova, A.** (2017), *Deep Learning Using Keras – The Basics*. [Erişim: 28.12.2020, <https://www.learnopencv.com/deep-learning-using-keras-the-basics/> ]

**Gürbüz, F. B.** (2018). *Convolutional Neural Networks (CNN)*. [Erişim: 28.12.2020, <https://medium.com/@batincangurbuz/konvol%C3%BCsyonel-sinir-a%C4%9Flar%C4%B1-evri%C5%9Fimli-sinir-a%C4%9Flar%C4%B1-olarak-da-bililir-convolutional-neural-4ecd2c5ad842> ]

**Gürbüz, F. B.** (2018). *Uzun Kısa Dönemli Bellek- Long Short Term Memory (LSTM)*. [Erişim: 28.12.2020, <https://medium.com/@batincangurbuz/uzun-k%C4%B1sa-d%C3%B6nemli-bellek-long-short-term-memory-lstm-c526980c28b1> ]

**Hassan, M.** (2018). *VGG16 – Convolutional Network for Classification and Detection*. [Erişim: 28.12.2020, <https://neurohive.io/en/popular-networks/vgg16/> ]

**İnik, Ö. & Ülger, E.** (2017). *Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modeller*. Gaziosmanpaşa Bilimsel Araştırma Dergisi (GBAD), 6(1), 85-104

**Kakıcı, A.** (2017). *Yapay Sinir Ağlarının Katmanları ve Sınıflandırılması*, [Erişim: 28.12.2020, <https://yapayzeka.ai/yapay-sinir-aglarinin-katmanlari-ve-siniflandirilmesi-3/> ]

**Karakuzu, C.** (2015), *Yapay Sinir Ağı Ders Notları*, Bilecik Şeyh Edebali Üniversitesi, Fen Bilimleri Enstitüsü, Bilecik

**Kayaalp, K. & Süzen, A.** (2018). *Derin Öğrenme ve Türkiye’deki Uygulamaları*, IKSAD International Publishing House, Institution Of Economic Development And Social Researches Publications.

**Keskenler, M. F. & Keskenler, E. F.** (2017). *Geçmişten Günümüze Yapay Sinir Ağları ve Tarihçesi*. Takvim-i Vekayi, Cilt:5, No:2, 8-18.

**Koç, M. L., Balas C. E. & Arslan, A.** (2004). *Taş Dolgu Dalgakıranların Yapay Sinir Ağları ile Ön Tasarımı*. İMO Teknik Dergi, 15(4), 3351-3375

**Kutlu, B., & Batur, B.** (2009). *Yapay Sinir Ağları ile Borsa Tahmini*. Yönetim Dergisi; İstanbul Üniversitesi İşletme Fakültesi İşletme İktisadı Enstitüsü, 25-40.

**Mehta S., Dadhich, S., Gumber, S. & Bhatt, A. J.** (2019). *Real-Time Driver Drowsiness Detection System Using Eye Aspect Ratio and Eye Closure Ratio*. Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur - India, February 26-28.

**Mikolov, T.** (2010). *Recurrent neural network based language model*, in Interspeech, 11th Annual Conference of the International Speech.

**Mittal, A.** (2019). *Understanding RNN and LSTM*. [Erişim: 28.12.2020, <https://towardsdatascience.com/understanding-rnn-and-lstm-f7cdf6dfc14e> ]

**Nagargoje, S., & Shilvant, D. S.** (2015) *Drowsy Driver Warning System Using Image Processing*. International Journal Of Engineering Development And Research (IJEDR), vol. 3(3), s. 78-83.

**Okumuş, R.** (2019). *Convolutional Neural Networks*. [Erişim: 28.12.2020, <https://medium.com/@rabiakumus96/convolutional-neural-networks-evri%C5%9Fimsel-sinir-a%C4%9Flar%C4%B1-cceb887a2979> ]

**Opperman, A.** (2019). *What is Deep Learning and How does it work*, [Erişim: 28.12.2020, <https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac> ]

**Öztemel, E.** (2006). *Yapay Sinir Ağları*. İstanbul: Papatya Yayıncılık Eğitim, 3. Basım

**Park, S., Pan, F., Kang, S., & Yoo, C. D.** (2016). *Driver drowsiness detection system based on feature representation learning using various deep networks*. In ACCV Workshops, 2016.

**Pedamonti, D.** (2018). *Comparison of non-linear activation functions for deep neural networks on MNIST classification task*. Bilgisayar Bilimleri Bölümü Edinburgh Üniversitesi, İskoçya

**Pokhrel, S.** (2019). *Beginners Guide to Convolution Neural Networks*. [Erişim: 28.12.2020, <https://towardsdatascience.com/beginners-guide-to-understanding-convolutional-neural-networks-ae9ed58bb17d> ]

**Qian, J., Liu, W., Yao, Z., Jiao, X., & Pan, J.** (2019). *Convolutional Two-Stream Network Using Multi-Facial Feature Fusion for Driver Fatigue Detection*. Special Issue on the Future of Intelligent Human-Computer Interface.

**Ramachandran, P., Zoph, B., & Le, Q. V.** (2017). *Swish: A Self-Gated Activation Function*. Google Brain Team, Kaliforniya.

**Ravi, A., Phanigna, T. R., Lenina, Y., Ramcharan, P., & Teja, P. S.** (2020). *Real Time Driver Driver Fatigue Detection and Smart Rescue System*. 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC).

**Reddy, B., Kim, Y. H., Yun, S., Seo, C. & Jang, J.** (2017). *Real-time driver drowsiness detection for embedded system using model compression of deep neural networks*. In 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 438–445.

**Rende F. Ş, Bütün G, & Karahan Ş.** (2016). *Derin Öğrenme Algoritmalarında Model Testleri: Derin Testler*. 10. Ulusal yazılım Mühendisliği Sempozyumu, Ekim, 2016

**S. Hochreiter & J. Schmidhuber,** (1997) “Long Short-Term Memory,” Neural Comput.

**Savas, B. K., & Becerikli, Y.** (2020). *Real Time Driver Fatigue Detection System Based on Multi-Task ConNN*. IEEE ACCESS, Volume 8, January 2020.

**Simonyan, K. & Zisserman A.** (2015). *Very Deep Convolutional Networks For Large – Scale Image Recognition*. International Conference on Learning Representations (ICLR).

**Sönmez, A.** (1999), *Ağır Vasıta Sürücülerinin Çalışma Koşulları ve Trafik Kazaları, Uzun Mesafe Yük ve Yolcu Taşımacılığı Yapan Sürücüler Üzerine Bir Çalışma*, T.C. Emniyet Genel Müdürlüğü Trafik Hizmetleri Başkanlığı Trafik Araştırma Merkezi Müdürlüğü, 198 sayfa.

**Suryaprasad, J., Sandesh, D., Saraswathi, J., Swathi, D. & Manjunath, S.** (2013). *Real Time Drowsy Driver Detection Using Haarcascade Samples*, İkinci Uluslararası İleri Bilgi Teknolojileri ve Uygulamaları Konferansı.

**Sutton, R. S. & Barto A. G.** (2015). *Reinforcement Learning: An Introduction*, The MIT Press Cambridge, Massachusetts London, England

**Şeker, A., Diri, B., & Balık, H. H.** (2017). *Derin Öğrenme yöntemleri ve Uygulamaları Hakkında Bir İnceleme*, Yayınlanmamış Doktora Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul.

**Tan, Z.** (2019). *Derin Öğrenme Yardımıyla Araç Sınıflandırma*. Yayınlanmamış Yüksek Lisans Tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ.

**Türkiye İstatistik Kurumu.** (2018). *Karayolu Trafik Kaza İstatistikleri*, Sayı: 30640

**Türkiye İstatistik Kurumu Haber Bülteni.** (2019). *Motorlu Kara Taşıtları*, Sayı: 30639

**Ursavaş, A., & Ercüment, E. G. E.** (2004). *Uyku Apne Sendromu ve Trafik Kazaları*. Uludağ Üniversitesi Tıp Fakültesi Dergisi, 30(1), 37-41

**Uslu, M.** (2016). *Yapay Sinir Ağları (YSA) - Bölüm 2*, Hacettepe Üniversitesi, Fen Fakültesi İstatistik Bölümü, [Erişim: 28.12.2020, <https://www.slideshare.net/uslumetin/gs-kanseri-verilerinin-yapay-sinir-alar-ile-snflandrlmas> ]

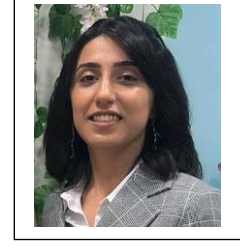
**Yaprak, Ş., & Akbulut A. M.** (2019). *Trafik Kaza ve Denetim İstatistikleri*, Polis Akademisi Yayınları: 75

**Yan, J. J., Kuo H. H, Lin, Y. F. & Liao, T. L.** (2016). *Real-Time Driver Drowsiness Detection System Based on PERCLOS and Grayscale Image Processing*. 2016 International Symposium on Computer, Consumer and Control (IS3C).

**LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R.E., Hubbard W. & Jackel, L. D.** (1989). *Backpropagation Applied to Handwritten Zip Code Recognition*, Neural Comput., vol. 1, no. 4, pp. 541–551.

**Yalçın, N.** (2012). *Sezgisel Algoritma Öğrenmeli Yapay Sinir Ağları ile Epilepsi Hastalığının Teşhisi*, Yayınlanmamış Yüksek Lisans Tezi Selçuk Üniversitesi Fen Bilimleri Enstitüsü, Konya

## ÖZ GEÇMİŞ



### Kişisel Bilgiler

Adı Soyadı : Esra ÇİVİK  
Doğum Yeri ve Tarihi : Siirt 07.01.1996  
ORCID : 0000-0002-4959-815X

### Eğitim Durumu

Lisans Öğrenimi : BŞEÜ Bilgisayar Mühendisliği  
Bildiği Yabancı Diller : İngilizce (orta)  
Bilimsel Faaliyetleri :

### İş Deneyimi

Stajlar : Havelsan A.Ş, Siirt Devlet Hastanesi Bilgi İşlem  
Projeler :  
Çalıştığı Kurumlar : CuteSafe Teknoloji A.Ş

### İletişim

0\*  
E-Posta Adresi : civik.esra@gmail.com

### Akademik Çalışmaları

- Çivik, E., Yüzgeç U., "Gömülü Sistem Üzerinde Derin Öğrenme ile Gerçek Zamanlı Sürücü Yorgunluk Tespiti", SİU 2020, 28. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı, (05.10.2020 - 07.10.2020), Gaziantep.

Tarih:...../...../.....