

**KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI**

DOKTORA TEZİ

**DERİN ÖĞRENME TABANLI SÜRÜCÜ YORGUNLUĞU
SEZME VE TAHMİN SİSTEMİ GELİŞTİRİLMESİ**

BURCU KIR SAVAŞ

KOCAELİ 2021

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

DOKTORA TEZİ

DERİN ÖĞRENME TABANLI SÜRÜCÜ YORGUNLUĞU
SEZME VE TAHMİN SİSTEMİ GELİŞTİRİLMESİ

BURCU KIR SAVAŞ

Prof. Dr. Yaşar BECERİKLİ
Danışman, Kocaeli Üniversitesi

Prof. Dr. Nevcihan DURU
Jüri Üyesi, Kocaeli Üniversitesi

Prof. Dr. Cabir VURAL
Jüri Üyesi, Marmara Üniversitesi

Prof. Dr. Cemil ÖZ
Jüri Üyesi, Sakarya Üniversitesi

Dr. Öğr. Üyesi Orhan AKBULUT
Jüri Üyesi, Kocaeli Üniversitesi

Tezin Savunulduğu Tarih: 16.04.2021

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında, geçici/sürekli fiziksel özelliklerden kaynaklanan yorgunluk ve/veya uykululuk halinin sürücü davranışları üzerindeki etkisinin araştırılması ve araştırma sonucunda yorgun/uykulu sürücülerin tespit edilmesi amaçlanmıştır.

Bu kapsamda, gün ışığı koşullarında (gündüz koşullarında) sürücüyü sabit kamera(lar) ile izleyip, elde edilen görüntüler üzerinden sürücünün yüz bölgesini tespit ederek, devamında yorgunluk tespiti için gereken davranışsal özellikleri çıkaran ve bu davranışsal özellikleri Adaboost, Destek Vektör Makinesi (Support Vector Machine-SVM), Tek Görevli ve Çok Görevli çalışan iki farklı (İki Çıkışlı, Üç Çıkışlı) Konvolüsyonel Sinir Ağı (Convolutional Neural Network-¹ConNN) ve Derin İnanç Ağları (Deep Belief Network-DBN) modelleri ile sınıflandıran bir sürücü yorgunluk tespit ve tahmin sistemi geliştirilmiştir. Bu modelleri test etmek amaçlı literatürde yer alan veri kümelerinin yanı sıra KouBM-DFD (Kocaeli Üniversitesi Bilgisayar Mühendisliği Sürücü Yorgunluk Tespiti Veri Kümesi) olarak adlandırılan yeni ve özgün bir görüntü veri kümesi oluşturulmuştur.

Doktora eğitim öğretim süresince desteğini esirgemeyen, tezimin her aşamasında çalışmalarına yön veren değerli Hocam ve danışmanım Sayın Prof. Dr. Yaşar BECERİKLİ'ye teşekkürlerimi sunarım.

Tez çalışmama bilgi ve tavsiyeleri ile katkıda bulunan tez izleme jürim Prof. Dr. Nevcihan DURU ve Prof. Dr. Cabir VURAL'a ve tez çalışmam boyunca beni destekleyen Doç. Dr. Suhap ŞAHİN'e teşekkürü borç bilirim.

Akademik çalışmalarım boyunca, bana her türlü desteği veren ve yanımda olan arkadaşlarım Dr. Öğr. Üyesi Meltem KURT PEHLİVANOĞLU'na, Seda KUL'a, Arş. Gör. Fulya AKDENİZ'e ve Pelinay TOHUMCU'ya teşekkürü borç bilirim.

Hayatım boyunca sıkıntılarımı, mutluluklarımı paylaşan, maddi manevi desteklerini benden hiçbir zaman esirgemeyen en büyük destekçilerim, annelerim Ayşe KIR ve Gönül SAVAŞ, babalarım Ahmet KIR ve Orhan SAVAŞ, sevgili kardeşlerim Pınar KIR ve Vural SAVAŞ, çok değerli eşim Oral SAVAŞ'a ve biricik kızım Maya SAVAŞ'a sonsuz teşekkürlerimi sunarım.

Bu doktora tez çalışmasını destekleyen Kocaeli Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimi'ne (BAP) (Proje No: 2017/087) teşekkürlerimi sunarım.

Nisan – 2021

Burcu KIR SAVAŞ

¹ * Bu tez çalışmasında **Konvolüsyonel Sinir Ağları (Convolutional Neural Network)** kısaltılması ConNN olarak verilmiştir. CNN ya da CoNN olarak kısaltılmamıştır. CNN uzun süredir Hücresel Sinir Ağları (Celluar Neural Network) tanımının kısaltılması olarak ve CoNN uzun süredir İşbirlikçi Sinir Ağları (Cooperative Neural Networks) sinir ağlarının kısaltılması olarak literatürde yer almaktadır.

İÇİNDEKİLER

ÖNSÖZ VE TEŞEKKÜR	i
İÇİNDEKİLER	ii
ŞEKİLLER DİZİNİ.....	iv
TABLolar DİZİNİ	viii
SİMGELER VE KISALTMALAR DİZİNİ	x
ÖZET.....	xiii
ABSTRACT.....	xiv
GİRİŞ	1
1. SÜRÜCÜ YORGUNLUĞU.....	8
1.1. Kullanılan ve Oluşturulan Veri Kümeleri.....	15
1.1.1. ZJU göz kırpma video veri kümesi	15
1.1.2. YawDD esneme tespiti video veri kümesi	16
1.1.3. Nthu-DDD yorgun sürücü tespiti video veri kümesi	17
1.1.4. KouBM-DFD: Kocaeli üniversitesi bilgisayar mühendisliği sürücü yorgunluk tespiti veri kümesi.....	18
1.2. Kullanılan Kütüphaneler	26
1.2.1. Dlib	26
1.2.2. OpenCV	27
1.2.3. TensorFlow	27
1.2.4. Pytorch	27
2. KULLANILAN SINIFLANDIRMA YÖNTEMLERİ	28
2.1. Adaboost	28
2.2. Destek Vektör Makineleri.....	28
2.3. Derin Öğrenme.....	29
2.4. Konvolüsyonel Sinir Ağları (ConNN).....	31
2.4.1. Konvolüsyon katmanı	32
2.4.2. Aktivasyon katmanı	34
2.4.3. Havuzlama katmanı	35
2.4.4. Düzleştirme katmanı	35
2.4.5. Tam bağlantı katmanı	36
2.4.6. Seyreltme (DropOut) katmanı	36
2.4.7. Çıkış katmanı	37
2.5. Derin İnanç Ağları	37
2.5.1. Kısıtlı boltzman makineleri	38
2.5.2. Kısıtlı boltzman makinelerinin öğrenmesi.....	40
2.5.3. Derin inanç ağları.....	41
3. KULLANILAN GÖMÜLÜ SİSTEM	44
3.1. NVIDIA Jetson	44
3.2. Jetson TX2	45
4. SÜRÜCÜ YORGUNLUĞU TESPİTİ ÇÖZÜM YAKLAŞIMLARI VE TEZ KAPSAMINDA GERÇEKLEŞTİRİLEN SENARYOLAR	47
4.1. Yorgunluk Parametreleri.....	47
4.2. Tez Kapsamında Gerçekleştirilen Senaryolar.....	49

4.2.1. Senaryo 1: Adaboost algoritması ile yorgunluk tahmin sistemi.....	50
4.2.2. Senaryo 2: SVM algoritması ile yorgunluk tahmin sistemi	53
4.2.3. Senaryo 3: Tek görevli ConNN modeli ile yorgunluk tahmin sistemi.....	57
4.2.3.1. Göz durum bilgisine dayanan yorgunluk tahmin sistemi.....	61
4.2.3.2. Ağız durum bilgisine dayanan yorgunluk tahmin sistemi.....	63
4.3.4. Senaryo 4: Çok Görevli ConNN ile yorgunluk tahmin sistemi	65
4.3.4.1. İki çıkışlı çok görevli ConNN ile yorgunluk tahmin sistemi.....	66
4.3.4.2. Üç çıkışlı çok görevli ConNN ile yorgunluk tahmin sistemi.....	88
4.3.5. Senaryo 5: DBN modeli ile yorgunluk tahmin sistemi	103
4.3.6. Senaryo 6: Jetson T2 üzerinde ConNN ve DBN modellerinin test edilmesi.....	114
5. SONUÇLAR VE ÖNERİLER	122
KAYNAKLAR	126
EKLER.....	139
KİŞİSEL YAYIN VE ESERLER	174
ÖZGEÇMİŞ	176

ŞEKİLLER DİZİNİ

Şekil 1.1.	Saat artışının sürücü yorgunluğuna etkisi	3
Şekil 1.2.	ZJU video veri kümesinden örnek görüntüler.....	16
Şekil 1.3.	YawDD video veri kümesinden örnek görüntüler	17
Şekil 1.4.	Nthu-DDD video veri kümesinden örnek görüntüler.....	18
Şekil 1.5.	Simülasyon ortamı.....	19
Şekil 1.6.	Aracın ön kamerası ile toplanan KouBM-DFD veri kümesinden örnek görüntüler	20
Şekil 1.7.	Aracın yan kamerası ile toplanan KouBM-DFD veri kümesinden örnek görüntüler.....	24
Şekil 1.8.	Dlib kütüphanesi ile bulunan yüz noktaları.....	26
Şekil 2.1.	Derin öğrenme kavramlarının birbiri ile ilişkisi.....	29
Şekil 2.2.	Derin öğrenme metotları	31
Şekil 2.3.	Genel ConNN modeli.....	31
Şekil 2.4.	Konvolüsyon işlemi örneği	34
Şekil 2.5.	ReLU fonksiyonu	35
Şekil 2.6.	Tam bağlantı katmanı.....	36
Şekil 2.7.	Seyreltme katmanı.....	37
Şekil 2.8.	RBM modeli.....	38
Şekil 2.9.	RBM eğitiminde kontrastısal diverjans	41
Şekil 2.10.	Genel DBN modeli.....	42
Şekil 3.1.	Çalışan Jetson TX2 kartının görüntüsü	46
Şekil 4.1.	Göz açık/kapalı olma durumu	48
Şekil 4.2.	Çerçeve kuyruğuna görüntünün alınması.....	50
Şekil 4.3.	Senaryo 1 blok diyagramı	51
Şekil 4.4.	Negatif görüntü örnekleri	51
Şekil 4.5.	Pozitif görüntü örnekleri	52
Şekil 4.6.	Senaryo 1 yüz bölgesi tespiti.....	52
Şekil 4.7.	Açık ve kapalı göz örnek görüntüleri.....	52
Şekil 4.8.	Gözün konum bilgileri, a) göz, b) sağ ve sol göz.....	53
Şekil 4.9.	GGAO'nun zamana göre değişimi.....	54
Şekil 4.10.	Göz kırpma hareketinin akış diyagramı	54
Şekil 4.11.	Ağız konum bilgileri	55
Şekil 4.12.	AAO'nun zamana göre değişimi.....	55
Şekil 4.13.	Esneme tespiti akış diyagramı.....	56
Şekil 4.14.	Gerçek zamanlı sürüş simülasyon test örnek görüntüleri.....	57
Şekil 4.15.	Yorgunluk tespiti test aşaması örnek gösterimi	57
Şekil 4.16.	Tek Görevli ConNN modeli.....	60
Şekil 4.17.	Göz durum bilgisine dayanan sistemin genel yapısı	61
Şekil 4.18.	Ağız durum bilgisine dayanan sistemin genel yapısı	63
Şekil 4.19.	Göz ve ağız durum bilgilerine dayanan sistemin genel yapısı	66
Şekil 4.20.	İki Çıkışlı Çok Görevli ConNN modeli	69
Şekil 4.21.	Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği.....	70

Şekil 4.22. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği.....	71
Şekil 4.23. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği.....	72
Şekil 4.24. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği.....	73
Şekil 4.25. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği.....	74
Şekil 4.26. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği.....	75
Şekil 4.27. Model-1(b) konvolüsyon 1. katman (6,236,316) çıktı örneği.....	76
Şekil 4.28. Model-1(b) havuzlama 1. katman (6,118,158) çıktı örneği.....	76
Şekil 4.29. Model-1(b) konvolüsyon 2. katman (8,114,154) çıktı örneği.....	76
Şekil 4.30. Model-1(b) havuzlama 2. katman (8,57,77) çıktı örneği.....	76
Şekil 4.31. Model-1(b) konvolüsyon 3. katman (10,54,74) çıktı örneği.....	76
Şekil 4.32. Model-1(b) havuzlama 3. katman çıktısı (10,27,37) çıktı örneği.....	76
Şekil 4.33. Model-2, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği.....	77
Şekil 4.34. Model-2, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği.....	77
Şekil 4.35. Model-2, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği.....	78
Şekil 4.36. Model-2 konvolüsyon 1. katman (6,158,118) çıktı örneği.....	79
Şekil 4.37. Model-2 havuzlama 1. katman (6, 79,59) çıktı örneği.....	79
Şekil 4.38. Model-2 konvolüsyon 2. katman (8, 38, 28) çıktı örneği.....	79
Şekil 4.39. Model-2 havuzlama 2. katman (8,19, 14) çıktı örneği.....	79
Şekil 4.40. Model-2 konvolüsyon 3. katman (10,8,6) çıktı örneği.....	79
Şekil 4.41. Model-2 havuzlama 3. katman (10,4,3) çıktı örneği.....	79
Şekil 4.42. Model-3, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği.....	81
Şekil 4.43. Model-3, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği.....	81
Şekil 4.44. Model-3, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği.....	82
Şekil 4.45. Model-3 konvolüsyon 1. katman (6,316,236) çıktı örneği.....	83
Şekil 4.46. Model-3 havuzlama 1. katman (6,105,78) çıktı örneği.....	83
Şekil 4.47. Model-3 konvolüsyon 2. katman (8,102,75) çıktı örneği.....	83
Şekil 4.48. Model-3 havuzlama 2. katman (8,33, 24) çıktı örneği.....	83
Şekil 4.49. Model-3 konvolüsyon 3. katman (10,30,21) çıktı örneği.....	83
Şekil 4.50. Model-3 havuzlama 3. katman (10,9,6) çıktı örneği.....	83
Şekil 4.51. Model-4, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği.....	84
Şekil 4.52. Model-4 konvolüsyon 1. katman (6,160,120) çıktı örneği.....	85
Şekil 4.53. Model-4 havuzlama 1. katman (6,53,39) çıktı örneği.....	85
Şekil 4.54. Model-4 konvolüsyon 2. katman (8,27,20) çıktı örneği.....	85
Şekil 4.55. Model-4 havuzlama 2. katman (8,8,6) çıktı örneği.....	86
Şekil 4.56. Model-4 konvolüsyon 3. katman (10,5,4) çıktı örneği.....	86
Şekil 4.57. Model-4 havuzlama 3. katman (10,1,1) çıktı örneği.....	86
Şekil 4.58. Göz, ağız ve kafa durum bilgisine dayanan sistemin genel	

yapısı	89
Şekil 4.59. Üç çıkışlı Çok Görevli ConNN modeli	91
Şekil 4.60. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği	93
Şekil 4.61. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği	93
Şekil 4.62. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği	94
Şekil 4.63. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği	95
Şekil 4.64. Model-1(b) konvolüsyon 1. katman (6, 316, 236) çıktı örneği	96
Şekil 4.65. Model-1(b) havuzlama 1. katman (6, 158, 118) çıktı örneği	96
Şekil 4.66. Model-1(b) konvolüsyon 2. katman (8, 154, 114) çıktı örneği	96
Şekil 4.67. Model-1(b) havuzlama 2. katman (8, 77, 57) çıktı örneği	96
Şekil 4.68. Model-1(b) konvolüsyon 3. katman (10, 74, 54) çıktı örneği	96
Model-1(b) havuzlama 3. katman (10, 37, 27) çıktı örneği	96
Şekil 4.70. Model-2, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği	97
Şekil 4.71. Model-2 konvolüsyon 1. katman (6,158,118) çıktı örneği	97
Şekil 4.72. Model-2 havuzlama 1. katman (6,79,59) çıktı örneği	98
Şekil 4.73. Model-2 konvolüsyon 2. katman (8,38,28) çıktı örneği	98
Şekil 4.74. Model-2 havuzlama 2. katman (8,19, 14) çıktı örneği	98
Şekil 4.75. Model-2 konvolüsyon 3. katman (10,8,6) çıktı örneği	98
Şekil 4.76. Model-2 havuzlama 3. katman (10,4,3) çıktı örneği	98
Şekil 4.77. Model-3, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği	99
Şekil 4.78. Model-3 konvolüsyon 1. katman (6,316,236) çıktı örneği	99
Şekil 4.79. Model-3 havuzlama 1. katman (6,105,78) çıktı örneği	100
Şekil 4.80. Model-3 konvolüsyon 2. katman (8,102,75) çıktı örneği	100
Şekil 4.81. Model-3 havuzlama 2. katman (8,33,24) çıktı örneği	100
Şekil 4.82. Model-3 konvolüsyon 3. katman (10,30,21) çıktı örneği	100
Şekil 4.83. Model-3 havuzlama 3. katman (10,9,6) çıktı örneği	100
Şekil 4.84. Model-4, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği	101
Şekil 4.85. Model-4 konvolüsyon 1. katman (6,160,120) çıktı örneği	101
Şekil 4.86. Model-4 havuzlama 1. katman (6,53,39) çıktı örneği	102
Şekil 4.87. Model-4 konvolüsyon 2. katman (8,27,20) çıktı örneği	102
Şekil 4.88. Model-4 havuzlama 2. katman (8,8, 6) çıktı örneği	102
Şekil 4.89. Model-4 konvolüsyon 3. katman (10,5,4) çıktı örneği	102
Şekil 4.90. Model-4 havuzlama 3. katman (10,1,1) çıktı örneği	102
Şekil 4.91. RBM eğitim algoritması	104
Şekil 4.92. DBN modeli eğitimi	105
Şekil 4.93. Test doğruluk oranları (KouBM-DFD)	110
Şekil 4.94. Test doğruluk oranları (Nthu-DDD)	111
Şekil 4.95. Test doğruluk oranları (YawDD)	111
Şekil 4.96. Test doğruluk oranları (KouBM-DFD)	112
Şekil 4.97. Test doğruluk oranları (Nthu-DDD)	112
Şekil 4.98. Test doğruluk oranları (YawDD)	113
Şekil 4.99. İki Çıkışlı Çok Görevli ConNN (KouBM-DFD) modelinin	

testi için, yorgunluğa geçiş evresindeki bir sürücüye ait rastgele seçilen 5 dk'lık videoda peşpeşe gelen 20 çerçevenin gösterimi	115
Şekil 4.100. İki Çıkışlı Çok Görevli ConNN (Nthu-DDD) modelinin testi için yorgun olmayan bir sürücüye ait rastgele seçilen 5 dk'lık videoda peşpeşe gelen 20 çerçevenin gösterimi	116
Şekil 4.101. Üç Çıkışlı Çok Görevli ConNN (Nthu-DDD) modelinin testi için yorgun bir sürücüye ait rastgele seçilen 2 dk'lık videoda peşpeşe gelen 60 çerçevenin gösterimi.....	118
Şekil 4.102. Üç Çıkışlı Çok Görevli ConNN (KouBM-DFD) modelinin testi için yorgun olmayan bir sürücüye ait rastgele seçilen 2 dk'lık videoda peşpeşe gelen 20 çerçevenin gösterimi	119
Şekil 4.103. Göz bilgileri için oluşturulan DBN (ZJU) modelinin testi için yorgun olmayan bir sürücüye ait rastgele seçilen 4 sn'lik videoda peşpeşe gelen 10 çerçevenin gösterimi.....	120
Şekil 4.104. Gerçek sürüş sahnesinde performans gösterimi.....	121

TABLolar DİZİNİ

Tablo 1.1. Sürücü yorgunluğu sebebi ile meydana gelen kaza sayıları.....	2
Tablo 3.1. Jetson Nano / Jetson TX2 / Tam yapay zeka kiti karşılaştırması.....	44
Tablo 3.2. Jetson TX2 teknik özellikleri	45
Tablo 4.1. Kafa düşme durum bilgisi	49
Tablo 4.2. Eğitim veri kümesi örneği.....	56
Tablo 4.3. Test veri kümesi örneği	56
Tablo 4.4. Veri kümeleri dağılımı	58
Tablo 4.5. Tek Görevli ConNN modellerinin ağ yapısı	58
Tablo 4.6. Göz durum bilgisine dayanan Tek Görevli ConNN modeli kullanılarak elde edilen doğruluk oranları	61
Tablo 4.7. Doğruluk oranlarının literatürde bulunan diğer yaklaşımlarla karşılaştırılması.....	62
Tablo 4.8. Ağız durum bilgisine dayanan Tek Görevli ConNN modeli ile elde edilen doğruluk oranları.....	64
Tablo 4.9. Doğruluk oranlarının literatürde bulunan diğer yaklaşımlarla karşılaştırılması.....	65
Tablo 4.10. Veri kümeleri dağılımı	66
Tablo 4.11. Modellerin ağ yapıları	67
Tablo 4.12. Model-1 eğitiminden elde edilen doğruluk oranları.....	68
Tablo 4.13. Eğitim ve test hata değerleri (döngü sayısı=100).....	70
Tablo 4.14. Eğitim ve test hata değerleri (döngü sayısı=100).....	72
Tablo 4.15. Eğitim ve test hata değerleri (döngü sayısı=100)	74
Tablo 4.16. Model-2 için elde edilen doğruluk oranları.....	80
Tablo 4.17. Model-3 için elde edilen doğruluk oranları.....	84
Tablo 4.18. Model-4 için elde edilen doğruluk oranları.....	86
Tablo 4.19. İki Çıkışlı Çok Görevli ConNN ağ modelleri için doğruluk oranları	87
Tablo 4.20. İki Çıkışlı Çok Görevli ConNN ağ modelleri eğitimlerinin zaman ve doğruluk oranı karşılaştırması.....	87
Tablo 4.21. Doğruluk oranlarının literatürde bulunan diğer yaklaşımlarla karşılaştırılması.....	88
Tablo 4.22. Veri kümeleri dağılımı	88
Tablo 4.23. Modellerin ağ yapıları	89
Tablo 4.24. Model-1 için elde edilen doğruluk oranları.....	92
Tablo 4.25. Üç Çıkışlı Çok Görevli ConNN ağ modelleri için doğruluk oranları	103
Tablo 4.26. Üç Çıkışlı Çok Görevli ConNN ağ modelleri eğitimlerinin zaman ve doğruluk oranı karşılaştırması.....	103
Tablo 4.27. DBN modelinde kullanılan veri kümelerinin dağılımı.....	104
Tablo 4.28. İlk gizli katman üzerinde farklı düğümlerin ters/negatif faz hatası (KouBm_DFD)	105
Tablo 4.29. Göz durum bilgisine dayanan farklı RBM katmanlarından elde edilen RBM eğitim doğruluk oranı.....	106

Tablo 4.30. Göz durum bilgisine dayanan farklı RBM katmanlarından elde edilen DBN test doğruluk oranı.....	107
Tablo 4.31. Ağız durum bilgisine dayanan farklı RBM katmanlarından elde edilen RBM eğitim doğruluk oranı.....	108
Tablo 4.32. Ağız durum bilgisine dayanan farklı RBM katmanlarından elde edilen DBN Test Doğruluk Oranı.....	109
Tablo 4.33. Önerilen yöntemin doğruluk oranlarının literatürde bulunan diğer yaklaşımlarla karşılaştırılması.....	113
Tablo 5.1. Önerilen özgün farklı 5 senaryodan elde edilen doğruluk oranları	123



SİMGELER VE KISALTMALAR DİZİNİ

x	: Girdi
n	: n girdi sayısı
$s(k)$: Konvolüsyon Katmanında Üretilen Özellik Haritası
$S(i, j)$: İki Boyut için Konvolüsyon Fonksiyonu
k	: Zaman İndeksi
W	: Ağırlık
K	: İki Boyutlu Filtre
A	: Adım Sayısı
F	: Filtre sayısı
D_1	: Girdi Görüntüsü Derinliği
D_2	: Çıktı Görüntüsü Derinliği
H_1	: Girdi Görüntüsü Yüksekliği
H_2	: Çıktı Görüntüsü Yüksekliği
$l-1$: Tam Bağlantı Katmanı
W_1	: Girdi Görüntüsü Genişliği
W_2	: Çıktı Görüntüsü Genişliği
$Y_i^{(l)}$: Tam Bağlantı Katmanındaki i . Özellik Haritası
a	: RBM Görünür Katmandaki Yanlılık Değeri
b	: RBM Gizli Katmandaki Yanlılık Değeri
v	: RBM Görünür Katman
h	: RBM Gizli Katman
$E()$: Enerji Fonksiyonu
$P()$: Olasılık Kütle Fonksiyonu
Z	: Normalizasyon Sabiti
α	: Öğrenme Adımı
z	: Gibbs Örnekleme Zinciri Adımı
$h(z)$: Gibbs Örnekleme Zinciri Adım Örnekleme
P	: PERCLOS Metriği
E_c	: Kapalı Göz Çerçevesinin Sayısı
E_t	: Toplam Açık Ve Kapalı Göz Çerçevesinin Sayısı
y	: Hedef Değerlerin Vektörü
\hat{y}	: Tahmin Değerlerinin Vektörü

Kısaltmalar

ARM	:Acorn RISC Machine (RISC Tabanlı İşlemci Mimarisi)
AVI	:Audio Video Interleave (Ses Görüntü Birleşimi)

CUDA	:Compute Unified Device Architecture (Hesaplanmış Birleşik Cihaz Mimarisi)
CEW	:The Closed Eyes in the Wild (CEW Veri Kümesi)
CNN	:Cellular Neural Network (Hücresel Sinir Ağı)
CoNN	:Cooperative Neural Networks (İşbirlikçi Sinir Ağları)
ConNN	:Convolutional Neural Network (Konvolüsyonel Sinir Ağları)
CPU	:Central Processing Unit (Merkezi İşlem Birimi)
CSS	:Cascading Style Sheets (Basamaklanmış Stil Katmanları)
DBN	:Deep Belief Network (Derin İnanç Ağları)
DDD:	:Driver Drowsiness Detection (Sürücü Uykululuk Algılama)
EEG	:Elektroensefalogram
FDCN	:Fatigue Detection Convolutional Neural Network (Konvolüsyonel Sinir Ağları Yorgunluk Algılama)
F-DDD	:Forward Driver Drowsiness Detection (İleri Sürücü Uykululuk Algılama)
FOM:	:Frequency of Open Mouth (Esneme Frekansı)
FPS	:Frame Per Second (Saniyedeki Kare Sayısı)
GAN	:Generative Adversarial Network (Çekişmeli Üretici Ağlar)
GPU	:Graphics Processing Unit (Grafik İşlem Birimi)
HTDBN	:Hierarchical Temporal Deep Belief Network (Hiyerarşik Zamansal Derin İnanç Ağı)
HTML	:Hyper Text Markup Language (Hiper Metin İşaret Dili)
IDE	:Integrated Development Environment (Bütünleşik Geliştirme Ortamı)
IOS	:Iphone Operating System (Iphone İşletim Sistemi)
IoT	:Internet of Things (Nesnelerin İnterneti)
k-NN	:k-nearest Neighbor (K-En Yakın Komşuluk)
KouBM-DFD	:Kocaeli Üniversitesi Bilgisayar Mühendisliği Sürücü Yorgunluk Tespiti
LBP	:Local Binary Pattern (Yerel İkili Örüntü)
LDA	:Linear Discriminant Analysis (Doğrusal Diskriminant Analizi)
LFPW	:Labeled Face Parts in the Wild (LFPW Veri Kümesi)
LSTM	:Long Short-Term Memory (Uzun-Kısa Vadeli Bellek)
macOS	:Macintosh (Mac İşletim Sistemi)
NHTSA	:National Highway Traffic Safety Administration (ABD Ulusal Karayolu Trafik Güvenliği İdaresi)
Nthu-DDD :	NTHU Drowsy Driver Detection (Nthu-DDD Veri Kümesi)
OpenCV	:Open Source Computer Vision
PERCLOS	:The Percentage Of Eyelid Closure (Göz Kapanma Oranı)
RBF	:Radial Basis Function (Radyal Tabanlı Fonksiyon)
RBM	:Restricted Boltzman Machine (Kısıtlı Boltzman Makineleri)
ReLU	:Rectified Linear Units Layer (Düzleştirilmiş Doğrusal Birim Katmanı)
RGB	:Red-Green-Blue (Kırmızı-Yeşil-Mavi)
RNN	:Recurrent Neural Network (Özyinelemeli Sinir Ağları)
SGD	:Stochastic Gradient Decent (Stokastik Gradyan İnişi)
SVM	:Support Vector Machine (Destek Vektör Makinesi)
TFLOPs	:Floating Point Operations per Second (Saniyede Kayan Nokta İşlemleri)
USB	:Universal Serial Bus (Flaş Sürücü)
XML	:Extensible Markup Language (Genişletilebilir İşaretleme Dili)

YawDD :A Yawning Detection Dataset (YawDD Veri Kümesi)
YSA :Yapay Sinir Ağları



DERİN ÖĞRENME TABANLI SÜRÜCÜ YORGUNLUĞU SEZME VE TAHMİN SİSTEMİ GELİŞTİRİLMESİ

ÖZET

Trafik kazalarının birçok nedeni vardır. Kazalar çevresel koşulların yanı sıra sürücü kusurlarından da kaynaklanmaktadır. Bu kusurların en önemlilerinden biri, sürücünün yorgun/uykulu durumda iken araç kullanmasıdır. Yorgunluk ve/veya uykululuk hali fiziksel olarak doğal bir durum olduğu için önlemesi zor bir açık problemdir. Bu nedenle araştırmacılar ve araç geliştiren firmalar, sürücülerin sürüş esnasındaki yorgunluk ve/veya uykululuk halini tespit ve tahmin eden sistemler geliştirmektedir.

Bilişim teknolojilerindeki gelişmelere paralel olarak, akıllı araç geliştirme çalışmaları da son yıllarda hız kazanmıştır. Akıllı araç tasarımlarının bir parçası olarak sürücü yorgunluğu tespit sistemleri de ön plana çıkmıştır, çünkü bu sistemler güvenli araç sürüşüne destek olmaktadır. Güvenli sürüş desteği sağlayan bu sistemler, sürücünün sürüş esnasındaki davranışlarını ve fiziksel durumunu takip ederek elde edilen verilere göre sürücünün yorgunluk durumunu/derecesini tespit ederler.

Bu çalışmanın amacı; gün ışığı koşullarında (gündüz koşullarında) sürücüyü sabit kamera(lar) ile izleyip, elde edilen görüntüler üzerinden sürücünün yüz bölgesini tespit ederek, devamında yorgunluk tespiti için gereken davranışsal özellikleri (göz kırpma sayısı, PERCLOS (Göz Kapanma Oranı), esneme, FOM (Esneme Frekans) ve kafa düşme durumu belirleyip; Adaboost, Destek Vektör Makinesi (SVM), Tek Görevli ve Çok Görevli çalışan iki farklı (İki Çıkışlı, Üç Çıkışlı) Konvolüsyonel Sinir Ağı (ConNN) ve Derin İnanç Ağları (DBN) modelleri ile sürücünün yorgunluk durumuna/derecesine karar vermektir.

Tez kapsamında, literatürde yer alan ZJU, YawDD, Nthu-DDD veri kümelerinin yanı sıra, bu çalışma kapsamında hazırlanan ve literatüre kazandırılan KouBM-DFD olarak adlandırılan özgün veri kümesi de sürücünün yorgunluk durumuna/derecesine karar verme amaçlı kullanılmıştır. Bu kapsamda; Senaryo 1, Senaryo 2, Senaryo 3, Senaryo 4, Senaryo 5 ve Senaryo 6 olmak üzere farklı 6 çalışma yapılmıştır. En iyi doğruluk oranı (accuracy); Senaryo 4'te İki Çıkışlı Çok Görevli ConNN Model-1 kullanılarak YawDD veri kümesi üzerinde %99,18 olarak elde edilmiştir. Aynı model KouBM-DFD veri kümesi üzerinde de denenerek %98,88 doğruluk oranı sağlanmıştır. Elde edilen sonuçlar değerlendirildiğinde; KouBM-DFD veri kümesinin literatürde yer alan diğer veri kümeleri gibi kullanılacak yöntemler için güvenilir veriler sağlayabileceği açıktır. Ayrıca yapılan deneysel çalışmalarda, DBN modeli kullanılmış olup KouBM-DFD veri kümesi üzerinde DBN-göz için %84, DBN-ağız için %80 doğruluk oranı elde edilmiştir.

Anahtar Kelimeler: Derin Ağlar, Derin İnanç Ağları, Konvolüsyonel Sinir Ağları, Sürücü Yorgunluk Durum/Derecesi, Sürücü Yorgunluğu Tespiti.

DETECTING DRIVERS' FATIGUE AND DEVELOPMENT OF A PREDICTION SYSTEM BASED ON DEEP LEARNING

ABSTRACT

There are many reasons for traffic accidents. Accidents are caused by driver defects as well as environmental conditions. One of the most important of these flaws is that the driver is driving in a drowsiness/ fatigue state. Because fatigue/drowsiness is physically natural, it is an obvious problem that is difficult to prevent. For this reason, researchers and vehicle developing companies develop systems that detect and predict drivers' fatigue/drowsiness during driving.

In parallel with the developments in information technologies, smart vehicle development studies have also upgraded in recent years. As part of smart vehicle designs, driver fatigue detection systems have also come to the forefront because these systems support safe vehicle driving. These systems, which provide safe driving support, detect the driver's fatigue status / degree according to the data obtained by following the behavior and physical condition of the driver during driving.

The aim of this study is; By monitoring the driver with fixed camera (s) in daylight conditions (in daylight conditions), detecting the driver's face region through the images obtained, then behavioral characteristics required for fatigue detection (blink number, PERCLOS (The Percentage of Eyelid Closure), determining the yawning, FOM (Frequency Of Mouth) and head situation); Adaboost, Support Vector Machine (SVM), two different (Two Outputs, Three Outputs) Convolutional Neural Network-ConNN and Deep Belief Network (DBN) models to determine the fatigue state/degree of the driver.

Within the scope of the thesis, besides the ZJU, YawDD, Nthu-DDD data sets in the literature, the original data set called KouBM-DFD, which was prepared within the scope of this study and added to the literature, was used to decide the fatigue status / degree of the driver. In this context; 6 different studies were conducted, namely Scenario 1, Scenario 2, Scenario 3, Scenario 4, Scenario 5, and Scenario 6. Best accuracy rate; In Scenario 4, it was achieved as 99.18% on the YawDD dataset by using Two Outputs Multitasking ConNN Model-1. The same model was tested on the KouBM-DFD dataset, achieving 98.88% accuracy. When the results obtained are evaluated; It is clear that the KouBM-DFD dataset can provide reliable data for the methods to be used, like other datasets in the literature. In addition, in the experimental studies, DBN model was used and an accuracy rate of 84% for DBN-eye and 80% for DBN-mouth was obtained on the KouBM-DFD dataset.

Keywords: Deep Learning, Deep Belief Networks, Convolutional Neural Network, Driver's Fatigue Status / Degree, Driver Fatigue Detection.

GİRİŞ

Günümüzde, birçok ulaşım aracı akıllı sistemlerle donatılıp, otonom ya da yarı-otonom haline getirilmektedir. Ancak, mevcut teknoloji ve altyapılar henüz yeterli değildir. Aracı süren/kontrol eden/yöneten insan faktörüdür. Bu nedenle, araç sürücüleri hala sürüş güvenliğinde en önemli rolü oynamaktadır. Doğrudan ya da dolaylı olarak araç sürücülerinden kaynaklanan kazaların sebepleri arasında, “yorgunluk ve/veya uykululuk hali” önemli yer tutar. Bu sebeple, sürücü yorgunluk ve/veya uykululuk hali tespit ve tahmini literatürde çalışılan önemli bir açık problemdir.

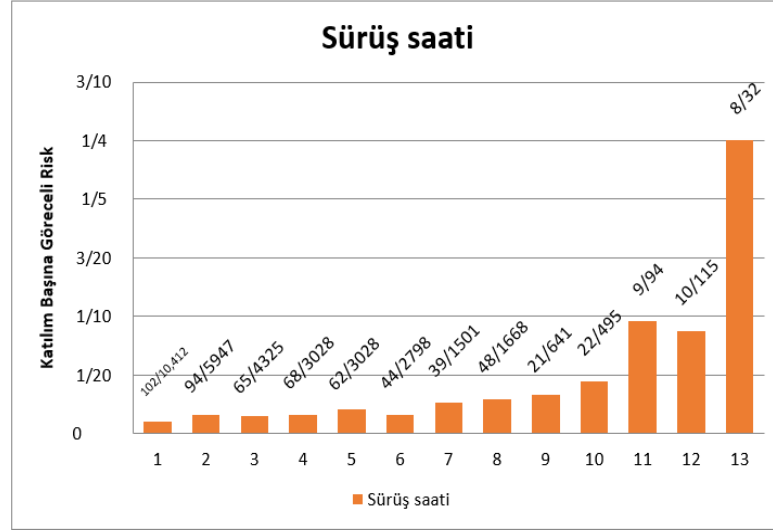
Dünya Sağlık Örgütü'nün istatistiksel verilerine göre (Facts, 2019), trafik kazalarında her yıl milyonlarca insan hayatını kaybetmektedir. İstatistiklere göre ölümcül kazaların büyük çoğunluğu, sürücü yorgunluğu ve dikkatsizliğinden kaynaklanmaktadır. Amerikan Otomobil Derneği tarafından yayınlanan raporda (Tefft, 2020), tüm kazaların %7'sine ve ölümcül trafik kazalarının ise %21'ine, yorgun sürücülerin sebebiyet verdiği belirtilmektedir. ABD'de 2003 yılında 4010 sürücü ile yapılan bir röportajda (Royal, 2003), sürücülerin %37'si hayatlarında bir noktada araba kullanırken başının düştüğünü ve bu sürücülerin %29'unun yıl içinde bu problemi yaşadığını bildirmişlerdir. 2006 yılında Kanada'nın Ontario eyaletinde 750 sürücü ile yapılan ankette (Hammoud ve diğ., 2008; Vanlaar ve diğ., 2000), sürücülerin yaklaşık olarak %60'ı, bazen uykulu ya da yorgun olduklarında araba sürmeyi kabul ederken, %15'i ise araç kullanırken sürüş sırasında başının düştüğünü bildirmiştir. ABD Ulusal Karayolu Trafik Güvenliği İdaresi (National Highway Traffic Safety Administration-NHTSA) (URL-2, 2018), yalnızca ABD'de 2005-2009 yılları arasında yıllık toplam ölümcül kazaların %2,2 ile %2,6 sının sürücü yorgunluğundan kaynaklandığını belirtmektedir. Sadece maddi zarara yol açan kazalar bu sonuçlara dahil edilmemektedir. 2009 yılında yayımlanan rapora göre (Facts, 2019), yaklaşık 30.000 yaralanma kazası meydana gelmiştir ve bu yaralanmaların %2'si sürücü yorgunluğundan kaynaklanmaktadır. Yıllara göre (URL-1, 2021) sürücü yorgunluğu sebebi ile meydana gelen kaza sayıları Tablo 1.1'de verilmiştir.

Tablo 1.1. Sürücü yorgunluğu sebebi ile meydana gelen kaza sayıları (URL-1, 2021).

	Toplam kaza	Uykulu sürücülerin dahil olduğu kaza	Oran
2013	43.965	1231	%2.8
2014	45.138	1309	%2.9
2015	48.770	1268	%2.6
2016	52.400	1310	%2.5
2017	52.240	1306	%2.5
2018	50.875	1221	%2.4

2017 yılında Foundation for Traffic Safety'nin yaptığı çalışmada (URL-2, 2018), bir haftada sürücülerin %42,4'ünün altı saatten daha az uyuyarak en az bir veya daha fazla gün araba kullandıkları belirtilmiştir. 10 sürücüden 3'ü (%30,8), çok yorgun olduklarında bile son bir ay içinde gözlerini açık tutmakta zorlandıkları halde araba kullandıklarını belirtmişlerdir. Ülkemizde yapılan araştırmaya göre (Sönmez, 1999), ağır vasıta sürücülerinin en önemli kaza nedenleri arasında %17,10'unun yorgun ve uykusuz araç kullanma kaynaklı olduğu tespit edilmiştir. Uyku nedenli trafik kazalarını belirlemedeki zorluklar nedeniyle (kazaya yorgunluğun sebep olup olmadığı ya da sebep oluyorsa yorgunluğun derecesini belirlemek gibi) 2000 yılından sonraki kaza istatistiklerinde kaza tespit tutanaklarından çıkartılmıştır ve anket yöntemine geçilmiştir (Sönmez, 1999; Baş, 2015).

Sürücüler sürüş biçimlerindeki değişmeyi ya da yorulduklarını fark etmeden önce yorgunluk ve uykusuzluk tarafından etkilenebilirler. Sürücüler yorgun ya da uykusuz olduklarında; herhangi bir olay karşısında tepki verme süreleri uzamaktadır. Sürücü yorgunluğuna bağlı kazalarda, sürücülerin kazayı önlemeye yönelik tepki/manevra süresi gecikir. Bu nedenle yorgunluk kaynaklı kazalar, daha şiddetli olmaktadır. Araç sürüş süresi arttıkça, sürücü yorgunluğu da artar. Şekil 1.1'de (URL-3, 2011), araç sürüş süresi artışının sürücü yorgunluğu üzerindeki etkisi ve ilgili kazaların yüzdesi arasındaki ilişki verilmiştir. Her çubuk grafiğinin üzerinde verilen ilk sayı, 1991 ve 2002 yılları arasındaki, her bir sürüş saati için sürücü yorgunluğu kaynaklı ölümcül kazalara karışan büyük kamyon sayısını temsil ederken, ikinci sayı ise aynı sürüş saatindeki tüm ölümcül kazalara karışan toplam büyük kamyon sayısını temsil etmektedir.



Şekil 1.1. Saat artışının sürücü yorgunluğuna etkisi (URL-3, 2011)

NHTSA'ya göre uyuklu araç sürüşünün, alkollü araç sürüşü kadar tehlikeli olduğu gösterilmiştir (URL-3, 2011). Bu nedenle, sürücü yorgunluğunu otomatik olarak tespit ve tahmin eden sistemler, ölümcül kazaların önlenmesi için gereklidir. Sürücü yorgunluk tespitinde, bir sürücünün yorgun/uyuklu olarak sınıflandırılabilmesi için olası durumlar (eylemler):

- Zikzak sürüş, şerit sürüklenme,
- Başka bir araca sürtmek,
- Öndeki araca çok yakın sürüş,
- Gezinme veya bağlantısız düşünceler,
- Tekrarlanan ve uzun süren esnemeler,
- Gözleri açık tutma zorluğu,
- Sürekli göz kırpma
- Sürücünün başını düzgün tutamaması

şeklinde verilebilir (Oliveria, 2018). Sürücü yorgun/uykululuk hali tespiti amaçlı teknolojiler; fizyolojik tabanlı, araç tabanlı ve davranışsal tabanlı olarak sınıflandırılabilir.

Fizyolojik tabanlı ölçümler, fizyolojik sinyallere veya bilgisayar görme yöntemlerine dayanan uyuşukluğu tespit etmektedir. Beyin dalgaları, kalp hızı ve solunum hızı, sürücü yorgunluğunun saptanması için kullanılan bazı fizyolojik sinyallerdir (Cobb ve

diğ., 1983; Xun ve diğ., 2009). Fizyolojik sinyaller ile elde edilen veriler sonucunda yüksek doğruluk oranı elde edilse bile, kullanılan donanımlar genellikle sürücü ile fiziksel temas gerektirir. Bu teknolojiler kullanıldığında yorgunluk ölçümü için, sürücüden içinde sensör barındıran bir nesneyi giymesi ya da vücudunun çeşitli yerlerine yapıştırması beklenir. Bu durum sürücü açısından rahatsızlığa neden olabilir, bu nedenle fizyolojik tabanlı ölçümler gerçek zamanlı sistemlerde genellikle tercih edilmemektedir. Araç tabanlı ölçümlerde, sürücünün direksiyon tutuş değişimi, araç hızındaki ve araç yönündeki değişim dikkate alınarak sürücü yorgunluğu/uykululuk hali tespit edilir. Bu ölçümlerde, sürücünün sürüş sırasındaki yorgunluğu ile aracı nasıl kullandığı arasında bir çıkarım yapılır (Feng ve diğ., 2009; Ingre ve diğ., 2006). Günümüzde, bazı otomobil üreticileri Volkswagen (TelematicsNews, 2012), Mercedes (MERCEDES Attention Assist™) (Mbusa, 2017), Volvo (Driver Alert) (Mashko, 2015), Nissan (Iizuka ve diğ., 1985) ve Renault (C Artaud ve diğ., 2014; Sahayadhas ve diğ., 2012), araç tabanlı ölçüm teknolojilerini kullanmaktadırlar. Ancak bu teknolojilerde araç donanımlarından gelen bilgiler girdi olarak kullanıldığından, sürücü yorgunluğu tespiti sınırlı durumlarda çalışmaktadır. Bu tür sistemler yolun geometrik karakteristiğine bağlıdır. Yol durumu (otoyol, virajlı yol,.. vb.) ve iklim koşulları, bu sistemler için yanıltıcı olabilir. Diğer bir ifadeyle bu sistemlerde, aracın sadece otoyol üzerindeki durumu dikkate alındığından, sürücü yorgunluk/uykululuk hali tespiti oldukça sınırlıdır (Artaud ve diğ., 2014). Şu an bu sistemler kullanılsa dahi; bu sistemlerin güvenilirliği, doğruluğu ve duyarlılığı belirsizliğini korumaktadır (Hartley ve diğ., 2000, Triyanti ve diğ., 2017, Li ve diğ., 2017).

Bu eksiklikler, üçüncü bir yaklaşım olan davranışsal tabanlı ölçüm sistemlerinin geliştirilmesinin önünü açmıştır. Davranışsal tabanlı ölçümlerde, bilgisayarlı görme ve görüntü işleme yaklaşımları ile sürücünün davranışlarına odaklanılarak, sürücü yorgunluğu/uykululuğu tespit edilir. Bu ölçümlerde, sürücünün yüz, göz, ağız ve baş durumlarının karakteristik değişimleri odak noktası kabul edilir (Cobb ve diğ., 1983) (Scherer ve diğ., 2010). Sistemde, temelde göz kırpma sayısı, gözün kapalı kalma süresi, esneme, ağız açıklığı gibi durumlar analiz edilir (Freund ve diğ., 1996; Viola ve diğ., 2001; Ji ve diğ., 2004; Kang ve diğ., 2013; Pratama ve diğ., 2017; Zhang ve diğ., 2017). Göz kapanma oranını ifade eden PERCLOS sürücü uyku yorgunluk seviyesi tespiti için en çok kullanılan parametredir (Ciresan ve diğ., 2011; Junaedi ve

diğ., 2018; Darshana ve diğ., 2014; Xie ve diğ., 2012). Yapılan ölçümler için, fizyolojik tabanlı ölçümlerin aksine, sürüş esnasında sürücüyü rahatsız etmeyecek şekilde konumlandırılmış kamera(lar) aracılığıyla; sürücünün davranışsal takibi yapılır. Ayrıca bu sistemlerin tespit başarısı, araç tabanlı ölçümlerin aksine; iklim koşulları ve yol durumu değişikliklerinden etkilenmez. Bunun yanı sıra, maliyeti düşük ve doğruluk oranı yüksek sistemlerdir.

Yukarıda verilen ölçüm sistemlerinin olumlu ve olumsuz yönleri değerlendirildiğinde; bu tez çalışmasında, sürücü yorgunluk/uykululuk tespiti için davranışsal tabanlı ölçüm kullanılmıştır. Bu sayede araç sürücüsünün dikkatini dağıtacak donanımların kullanımının aksine, sürücüyü rahatsız etmeyecek şekilde konumlandırılmış kamera(lar) aracılığıyla yorgunluk/uykululuk tespiti amaçlanmaktadır.

Davranışsal tabanlı ölçüme dayanan, Bosch (Bosch, 2017) ve NVIDIA (Nvidia, 2017) gibi otomobil tedarikçileri sürücü yorgunluk/uykululuk hali algılama sistemleri geliştirmektedir. Önümüzdeki birkaç yıl içinde, araç sürüş görevini otomatik sürüş sistemlerine bırakan otonom sistemlerin, kademeli olarak devreye alınması beklenmektedir. Bu durum da sürücü yorgunluğu/uykululuğu tespitinin, otomatik sürüş sistemleri için de önemli bir açık problem olduğunu kanıtlar (Walch ve diğ., 2017; N.H.T.S., 2018).

“Herhangi bir ön işleme adımı yapılmadan, gün ışığı koşullarında (gündüz koşullarında) sürücüyü sabit kamera(lar) ile izleyip, elde edilen görüntüler üzerinden sürücünün yüz bölgesindeki davranışsal özelliklerin değişimi; geliştirilen özgün modeller ile sınıflandırılarak, yorgunluk parametreleri yardımıyla sürücünün yorgunluk/uykululuk hali, gerçek zamanlı sistemler üzerinde de yüksek başarımla tespit edilebilir” yargısı tezin üzerine kurulduğu hipotezdir. Bu hipotez, tez kapsamında geliştirilen Senaryo 1, Senaryo 2, Senaryo 3, Senaryo 4, Senaryo 5 ve Senaryo 6 çalışmaları üzerinde yapılan testler ve elde edilen sonuçlar ile kanıtlanmıştır.

Tez çalışmasının temel amacı; göz, ağız ve kafa düşme durumu gibi çoklu durumları dikkate alan davranışsal tabanlı ölçüme dayanan, gerçek zamanlı sistemlerle uyumlu yorgunluk/uykululuk hali tespit ve tahmin sisteminin geliştirilmesidir.

Bu temel amaç kapsamında aşağıda verilen işlerin gerçekleştirilmesi hedeflenmiştir:

- Literatürde bulunan veri kümelerinin tez çalışması için kullanılabilir şekilde etiketlenmesi ve düzenlenmesi,
- Bu tez çalışması için hazırlanan ve KouBM-DFD olarak adlandırılan yeni ve özgün bir görüntü veri kümesinin oluşturulması: Gönüllü sürücülerden yorgun ve yorgun değilken elde edilecek ve evrensel olarak kullanılabilir veri kümesinin oluşturulması (9 farklı ülkeden/etnik gruplardan kişiler),
- Araç sürüş esnasında, sürücülerin yüz bölgelerinin tespit ve takibinin yapılması,
- Sürücü yorgunluk/uykululuk tespiti için gereken göz kırpma sayısı, PERCLOS, esneme, FOM, ağız açıklığı ve kafa düşme durumlarının tespiti,
- Adaboost, SVM, Tek Görevli ve Çok Görevli çalışan farklı iki (İki Çıkışlı, Üç Çıkışlı) ConNN modeli ve DBN modelinin tasarlanması ve kodlanması,
- Sürücü yorgunluk durumu/derecesi tahmini: Yorgunluk durumu/derecesi için kullanılan sınıf ve etiket değerleri:
 - Sürücünün uykuya dalma durumu: çok yorgun (2)
 - Sürücünün yorgunluk belirtisi gösterme durumu: az yorgun (1)
 - Sürücünün herhangi bir yorgunluk belirtisi göstermediği durum: yorgun değil (0)
- Tek Görevli ve Çok Görevli çalışan farklı iki (İki Çıkışlı, Üç Çıkışlı) ²ConNN modeli ve DBN modellerinin gerçek zamanlı olarak NVIDIA Jetson TX2 gömülü sistem modülü üzerinde test edilmesi,
- Tüm sonuçların literatürdeki çalışmalarla karşılaştırılması

Tez kapsamında geliştirilen yeni ve özgün sürücü yorgunluk/uykululuk hali tespit ve tahmin sistemi farklı 6 senaryo modeli içermektedir. Bu modellerin başarımını test etmek amaçlı da kullanılan ve KouBM-DFD olarak adlandırılan yeni ve özgün bir görüntü veri kümesi literatüre kazandırılmıştır. Tez çalışması; Sürücü Yorgunluğu, Kullanılan Sınıflandırma Yöntemleri, Kullanılan Gömülü Sistem, Sürücü Yorgunluğu Tespiti Çözüm Yaklaşımları ve Tez Kapsamında Gerçekleştirilen Senaryolar, Sonuçlar ve Öneriler olmak üzere farklı 5 bölümden oluşmaktadır. Tez çalışmasının “Sürücü Yorgunluğu” başlıklı bölümünde, sürücü yorgunluğu ile ilgili literatürde yer

² * Bu tez çalışmasında **Konvolüsyonel Sinir Ağları (Convolutional Neural Network)** kısaltılması ConNN olarak verilmiştir. CNN ya da CoNN olarak kısaltılmamıştır. CNN uzun süredir Hücresel Sinir Ağları (Cellular Neural Network) tanımının kısaltılması olarak ve CoNN uzun süredir İşbirlikçi Sinir Ağları (Cooperative Neural Networks) sinir ağlarının kısaltılması olarak literatürde yer almaktadır.

alan çalışmalar kapsamlı olarak verilmiştir. Ayrıca, tez çalışmasının özgün değeri ve literatüre olan katkıları bu bölümde ayrı bir alt bölüm olarak verilmiştir. “Kullanılan Sınıflandırma Yöntemleri” başlıklı bölümde ise, tez çalışmasına ait matematiksel alt yapı verilmiştir. Tez çalışmasında kullanılan Adaboost, SVM, ConNN ve DBN modelleri alt başlıklı bölümlerde detaylandırılmıştır. “Kullanılan Gömülü Sistem” başlıklı bölümde ise, tez çalışmasında kullanılan NVIDIA Jetson TX2 gömülü sistem modülü hakkında kapsamlı bilgi verilmiştir. “Sürücü Yorgunluğu Tespiti Çözüm Yaklaşımları ve Tez Kapsamında Gerçekleştirilen Senaryolar” başlıklı bölümde, tez çalışmasında geliştirilen sürücü yorgunluk/uykululuk tespiti sistemine ait kapsamlı bilgiler ve tez kapsamında gerçekleştirilen senaryolar detaylandırılmıştır.

“Sonuçlar ve Öneriler” başlıklı bölümde, yapılan çalışmalardan elde edilen sonuçlar analiz edilerek, önerilen yapıların literatüre ve gelecekte yapılacak çalışmalara sağlayabileceği katkılar ve ileriki çalışmalar tartışılmıştır.

Bu doktora tez çalışması kapsamında geliştirilen tüm yapılar ve çalışmalar özgün değere sahiptir. Senaryo 4’te verilen ve literatüre kazandırılan, davranışsal tabanlı ölçüme dayanan özgün Çok Görevli ConNN sinir ağı modeli, literatürde yer alan ConNN modelinin geliştirilmiş ve daha yüksek sınıflandırma başarımı sağlaması özelliğiyle iyileştirilmiş halidir. Literatürde yer alan çalışmalar incelendiğinde, önerilen model yapısında benzer bir çalışmaya rastlanmamıştır. Bunun yanı sıra, tez kapsamında KouBM-DFD olarak adlandırılan yeni ve özgün bir görüntü veri kümesi literatüre kazandırılmıştır. Tez çalışmasının çıktıları ele alındığında, bu tez çalışması oldukça özgün bir değere sahiptir. Bu nedenle, tezin literatüre ve gelecekte yapılacak çalışmalara önemli katkılar sağlayacağı öngörülmektedir.

1. SÜRÜCÜ YORGUNLUĞU

Yorgunluğun uyanıklık ve uyku arasındaki geçici süre olduğu ve eğer kesintisiz olursa uykuya neden olabileceğini belirtilmektedir. Grandjean, yorgunluğu azaltılmış verimlilik ve genel çalışma isteksizliği ile işaretlenmiş bir durum olarak tanımlamaktadır (Grandjean, 1979; Grandjean, 1988). Brown, yorgunluğu görevi yerine getirmeye devam etmekten vazgeçme olarak tanımlarken, kişinin yorgunluk durumunun farkına varmasından sonra çalışmaya devam etmesi durumunda, verimliliğin de düştüğünü belirtmiştir.

Yorgunluk; fiziksel ve zihinsel yorgunluk olarak ikiye ayrılır. Zihinsel yorgunluğun psikolojik olduğuna inanılırken, fiziksel yorgunluğun kas yorgunluğu ile eşanlamlı olduğu düşünülmektedir (Brown, 1994). İnsanlar yorgun veya uykuluyken araba kullandıklarında, bu durum genellikle “sürücü yorgunluğu” veya “uykulu sürüş” olarak ifade edilir. Uykululuk hali, uyanık olma ile uykuda olma arasındaki geçiş durumudur ve bu sırada uyanıklıkta bir azalma görülür. Bu durum araç sürüşü gibi sürekli dikkat gerektiren durumlarda kazalara yol açar (Oliveria, 2018).

Bu tez çalışmasında sürücü yorgunluk/uykululuk hali tespit ve tahmin sistemi geliştirilmiştir. Bu kapsamda, sürücü yorgunluğunun tespiti ile ilgili literatürde yer alan çalışmalar kapsamlı olarak incelenmiştir.

Araştırmacıların sürücü yorgunluğu üzerine yaptıkları çalışmalar 90’lı yılların sonu 2000’li yılların başlarına dayanmaktadır (Hamouda ve diğ., 1995; Eriksson ve diğ.,1997; Ji ve diğ., 2002; Ji ve diğ., 2004; Lee ve diğ., 2008; Chau ve diğ., 2012). Uykululuk doğal bir fizyolojik ihtiyaç ile ilişkilidir, bu yüzden ortadan kaldırılması mümkün değildir (Colic ve diğ., 2014). Bu nedenle, araştırmacılar çalışmalarında sürücü yorgunluğunu engellemek yerine; sürücünün yorgun/uykululuk halinde olup olmadığını izleme, tespit etme ve uyarma yöntemleri üzerine çalışmalar yapmıştır (Kim ve diğ., 2014).

Literatürdeki çalışmalara bakıldığında, araç sürüş videolarından davranışsal özelliklerin çıkarımı amaçlı yüz, göz veya ağız tespiti için Viola – Jones algoritması, Bağlı Bileşen Analizi (Connected Component Analysis), Haar Sınıflandırıcılar, Hough Dönüşümü veya Gabor Filtresi kullanılmıştır (Hong ve diğ., 2007; Fan ve diğ., 2008; Sahayadhas ve diğ., 2012; Danisman ve diğ., 2010; Mandal ve diğ., 2017; Diddi ve diğ., 2015; Talukder, 2011, Ali ve diğ., 2019). Davranışsal özelliklerin çıkarımı için en çok kullanılan teknikler; Geometrik kısıtlamalar, Dalgacık Paket Ayırıştırma, Gabor Dalgacıkları, Temel Bileşen Analizi veya Ayrık Dalgacık Dönüşümüdür (Horn ve diğ., 2008; Fan ve diğ., 2008). Bazı çalışmalarda yorgunluk durumu; eşik seviyelerine göre sınıflandırılırken, bazı çalışmalarda ise SVM, Adaboost algoritması, bulanık sınıflandırıcı, Bayes Ağları, Rastgele Orman ve Doğrusal Diskriminant Analizi (Linear Discriminant Analysis-LDA) gibi sınıflandırıcılar kullanılmıştır (Ji ve diğ., 2004; Fan ve diğ., 2008; Danisman ve diğ., 2010; Abtahi ve diğ., 2011; Daza ve diğ., 2011; Sacco ve diğ., 2012; Sahayadhas ve diğ., 2012; Luo ve diğ., 2013; Mbouna ve diğ., 2013; Lin ve diğ., 2013; Darshana ve diğ., 2014; Manu ve diğ., 2017; Mandal ve diğ., 2017; Kır Savaş ve diğ., 2017; Lyu ve diğ., 2016; Kır Savaş ve diğ., 2018; Ali ve diğ., 2019). Literatürde yer alan göz tespiti amaçlı yapılan çalışmalar incelendiğinde; Hong ve diğ. (2007) çalışmalarında, sürücü göz konumunu hesaplamak için, Haar sınıflandırıcılar ile yüz tespiti yaparak sürücünün yorgunluk durumunu belirlemek için gözün açıklık ve kapalılık durumlarının oranlarından yararlanmışlardır. Lalonde ve diğ. (2007) çalışmalarında, insanların göz kırpma hareketini tespit etmek amaçlı geliştirdikleri uygulamada kızılötesine yakın aydınlatmada elde edilen çok düşük kontrastlı görüntülere GPU tabanlı, gerçek zamanlı göz kırpma dedektörü gerçekleştirmişlerdir. Bu dedektör ile, veri toplama ve analiz işlemlerini gerçekleştirmektedirler. Fan ve diğ. (2008) çalışmalarında, göz kırpma durumunu değerlendirerek sürücünün yorgunluk seviyesini belirlemişlerdir. Gözlerin sınıflandırılması için AdaBoost, esneme hareketinin tespiti için ise Gabor dalgacık yöntemi ve sınıflandırılması için LDA yöntemi ile kullanılmıştır. İki davranış özelliğini birleştirmek için; Bayes Ağlarına dayalı olasılıklı yöntemi önermişlerdir. Liu ve diğ. (2008) Radyal Tabanlı Fonksiyon (Radial Basis Function-RBF) sinir ağını temel alan göz durumu sınıflandırması için bir algoritma önermişlerdir. Wu ve diğ. (2010) çalışmalarında, göz tespiti için SVM algoritmasından yararlanmışlardır. Göz tespit edildikten sonra, Yerel İkili Örüntü (Local Binary Pattern-LBP) ve SVM

sınıflandırıcı kullanarak, gözün açık ya da kapalı olma durumunu tespit etmişlerdir. Luo ve diğ. (2013) çalışmalarında farklı ışık altındaki sürüş ortamlarında gözlerin kesin konumunu bulmak için AdaBoost algoritmasından yararlanmışlardır. Yorgunluk tespiti için PERCLOS ve gözlerin karakteristik parametrelerini kullanmışlardır. Bahsedilen bu çalışmalarda sürücü yorgunluğunun tespitinde odaklanılan ana etmen göz durum bilgisidir.

Literatürde yer alan sürücü yorgunluk tespiti çalışmalarında göz durum takibinin yanı sıra, ağız ve kafa düşme durumunun da tespiti amaçlı çalışmalar yapılmıştır. Bu çalışmalar incelendiğinde; Lyu ve diğ. (2016) uykululuğu tanımlamak için yüz hizasına dayalı (kafa durum bilgisi) bir model önermişlerdir. Bu modelde etkili yüz tanımlayıcıları çıkarılarak, sürücü yüz durumlarını Rastgele Orman ile sınıflandırmışlardır. F-DDD ve Nthu-DDD veri kümeleri üzerinde modellerinin başarımlarını test etmişlerdir. Zhang ve diğ. (2017) çalışmalarında, Uzun-Kısa Vadeli Bellek (LSTM) Ağlar ile sürücü esneme tespit sistemi önermişlerdir. Sistem uzamsal görüntü özelliklerini çıkarmak için CNN ve zamansal özellikleri analiz etmek için LSTM ağlardan yararlanmışlardır. Kır Savaş ve diğ. (2017) çalışmalarında, kendi oluşturdukları veri kümesi üzerinde yorgunluğu iki sınıfa ayırarak, göz kırpma analizi için PERCLOS ölçütünü kullanmışlardır. Kır Savaş ve diğ. (2018) çalışmalarında, sürücü yorgunluğu tespiti için farklı davranış modellerini incelemişlerdir. Sürücü yorgunluğu tespiti için seçilen davranışları, kendi oluşturdukları veri kümesi üzerinde SVM algoritması ile sınıflandırmışlardır. Ali ve diğ. (2019) çalışmasında, yüz tanıma için Haar sınıflandırıcıları kullanmışlardır. Önerdikleri sürücü yorgunluk tespit sistemi, sürücünün esneme durumuna bağlıdır. Sistem Raspberry Pi 3 üzerinde gerçek zamanlı olarak test edilmiştir.

Son yıllarda, sürücü yorgunluğu ile ilgili özellikleri elde etmek ve sınıflandırmak için derin öğrenme algoritmaları kullanılmaya başlanmıştır (Dwivedi ve diğ., 2014; Zhang ve diğ., 2015; Lai ve diğ. (2015), Yan ve diğ., 2015; Park ve diğ., 2016; Yuen ve diğ., 2016; Yan ve diğ., 2016; Zhang ve diğ., 2017; Huang ve diğ., 2018; Tayibnapis ve diğ., 2018; Miah ve diğ., 2019; Kır Savaş ve diğ., 2020; Zhao ve diğ., 2020). Derin öğrenme yapısı çoklu soyutlama yapısı ile verinin temsillerini öğrenmek için bir araya getirilmiş çoklu işleme katmanlarından oluşmaktadır (Lecun ve diğ., 2015). Doğru özelliklerin seçimi, sınıflandırma aşaması için oldukça önemlidir. Bu bağlamda derin

öğrenme algoritmaları performans değerlendirmesi için katkı sağlayacağından görüntü sınıflandırmak için tercih edilebilir bir yöntem olmuştur. En çok kullanılan derin öğrenme modellerinden biri olan CNN, görüntüyü daha basit bir şekilde işleyebilmekte ve daha başarılı sonuçlar elde edebilmektedir (Yan ve diğ., 2016; Yuen ve diğ., 2016; Park ve diğ., 2016). CNN'lerin bir yararı da aynı sayıda gizli birimle tamamen bağlı ağlardan daha az sayıda eğitim gerektirmeleri ve daha az sayıda parametreye sahip olmalarıdır. Literatürdeki sistemler genellikle sürücünün yorgun olduğu her durum için ayrı eğitim ve test aşamalarına sahiptir ve daha sonra birleştirilmektedir.

Literatürde yer alan CNN tabanlı sürücü yorgunluğu tespiti yapan çalışmalar incelendiğinde; Dwivedi ve diğ. (2014) çalışmalarında, çeşitli gizli yüz özelliklerini ve karmaşık doğrusal olmayan özellik etkileşimlerini açıkça yakalamak için, CNN modelini önermişlerdir. Sürücünün, uykulu veya uykusuz olarak sınıflandırılması için softmax katmanı kullanılmıştır. Zhang ve diğ., (2015) çalışmalarında, yüz dedektörü, burun dedektörü, burun izleyicisi ve esneme detektöründen oluşan esneme tespit sistemi önermişlerdir. Kalman filtresi ile özel bir açık kaynak Takip, Öğrenme ve Algılama yöntemini birleştirip burun izleme algoritması geliştirmişlerdir. Son olarak, burun izleme güven değeri, ağız köşeleri etrafındaki gradyan özellikleri ve yüz hareket özellikleri gibi özellikler CNN modeli ile sınıflandırılmıştır. Lai ve diğ. (2015) çalışmalarında, yüz işaret noktalarının tespiti için derin öğrenme yöntemlerini kullanmışlardır. Yüz işaretlerini belirlemek için CNN ve Özyinelemeli Sinir Ağları (RNN) yöntemleri kullanılmıştır. Önerilen sistem üç bölümden oluşmaktadır; ilk bölümde evrişimsiz özellik haritaları çıkarılmıştır, ikinci bölümde ise bu evrişimsiz özellik haritaları üstünde ek bir konvolüsyonel katmanla yüz işaret noktalarının başlangıç koordinatları tahmin edilmiştir, üçüncü bölümde ise evrişimsiz özellik haritaları ve ilk yüz işaret noktaları girdi olarak kullanılarak, yüz işaret noktalarının koordinatları, birden çok LSTM bileşenlerinden oluşan tekrarlayan bir ağ ile hesaplanmıştır. Yan ve diğ., (2016) çalışmalarında, önceden tanımlanmış sürüş duruşlarını otomatik olarak öğrenmek ve tahmin etmek için CNN modeli kullanmışlardır. Yazarlar çalışmalarında, CNN modeli ile ilk önce seyrek filtreleme (sparse filtering) adı verilen denetimsiz bir öğrenme yönteminde önce eğitim yapıp daha sonra sınıflandırma yapmışlardır. Çalışmada, sınıflandırma için Southeast

Üniversitesi sürüş veri kümesi kullanılmıştır. Yuen ve diğ., (2016) çalışmalarında, sürücü kabininde bulunan zorlukları göz önünde bulundurarak yüz algılama, yüz işareti lokalizasyonu ve kafa düşme tahmin yöntemini uygulamak için CNN yapılarına dayanan bir sistem önermişlerdir. Park ve diğ. (2016) çalışmalarında, Kırmızı-Yeşil-Mavi (Red-Green-Blue-RGB) giriş videosu verildiğinde sürücü özellikleri öğrenmek ve yorgunluğu tespit etmek amaçlı derin uykululuk algılama ağı olarak adlandırdıkları bir mimari önermişlerdir. Bu ağ, üç derin ağıdan oluşmaktadır (AlexNet, VGG-FaceNet ve Flow-Net). Deneysel sonuçlar Nthu-DDD veri kümesi ile test edilmiştir. Sınıflandırmayı kendi hazırladıkları veri kümesi üzerinde gerçekleştirmişlerdir. Reddy ve diğ., (2017) çalışmalarında, eşikleme yöntemi kullanarak bir bölütleme algoritması önermişlerdir. Çalışmada ağız bölgesi içerisinde maksimum alana sahip olan bölümlenmiş bölge, esneme çerçevesi olarak YawDD veri kümesi üzerinde sınıflandırılmıştır. Junaedi ve diğ., (2018) çalışmasında, sürücü yorgunluğunu tespit etmek amaçlı, her karede gözü algılayan bir model önermişlerdir. Tespit edilen gözden, sol ve sağ gözlerin iris bölgelerini kullanarak PERCLOS hesaplanmış, modelin başarısı YawDD video veri kümesi üzerinde test edilmiştir. Huang ve diğ., (2018) çalışmasında, CNN modeline dayanan FDCN olarak adlandırılan model önermiştir. Modelin başarısı Wild (CEW) ve ZJU veri kümeleri üzerinde test edilmiştir. Miah ve diğ., (2019) çalışmalarında, gözün durumuna bağlı yorgunluk tespiti için model önermişlerdir. Gözlerin konumu ve ortalama göz konumunun mesafesine bağlı göz durumu tespit edilmiştir. Çalışmada gözün kapalı olduğu durumda uyku hali durumunu belirlemek amaçlı, gözün kapalı kalma süresi dikkate alınmıştır.

CNN modeli görüntü ve video sınıflandırmada yaygın olarak uygulanan etkili bir derin öğrenme yöntemidir (Lecun ve diğ., 1998). Ancak, CNN parametreleri çok fazladır, bu da hesap yükünün artmasına neden olur. Pratik uygulamalarda, CNN tabanlı sistemler yüksek sistem maliyetlerine yol açar, bu nedenle literatüre bakıldığında son yıllarda derin öğrenme modellerinden DBN modelinin kullanımı artmıştır (Mohamed ve diğ., 2012; Zhao ve diğ., 2015; Hua ve diğ., 2015; Hajinoroozi ve diğ., 2015; Weng ve diğ., 2016; Zhao ve diğ., 2018; Jabbar ve diğ., 2018; Ma ve diğ., 2018; Zheng ve diğ., 2019).

Literatürde yer alan DBN tabanlı sürücü yorgunluğu tespiti yapan çalışmalar incelendiğinde; Weng ve diğ., (2016) çalışmalarında, araba kazalarını en aza

indirgemek ve önlemek için uykulu sürücü uyarı sistemleri geliştirmişlerdir. Yazarlar, uykululuk tespiti için Hiyerarşik Zamansal Derin İnanç Ağı (HTDBN) modelini kullanmışlardır. DBN'lerin üzerine iki sürekli gizli Markov modeli oluşturarak; yüz hareketleri arasındaki etkileşimli ilişkileri modellemeyi ve tespit etmeyi amaçlamışlardır. Modelin başarımı, Nthu-DDD veri kümesi üzerinde test edilmiştir. Zhao ve diğ., (2018) çalışmalarında, DBN tabanlı sürücü yorgunluğu tahmin sistemi geliştirmişlerdir. Yüksek çözünürlüklü kamera kullanılarak çekilen videolar üzerinden yüzde bulunan göz işaretlerini belirlemişler ve buna bağlı yorgunlukla ilgili çıkarımlar yapmışlardır. Jabbar ve diğ., (2018) çalışmalarında, çok katmanlı algılayıcı sınıflandırıcılara dayalı yorgunluk algılama sistemi önermişlerdir. Zheng ve diğ. (2019) çalışmalarında, lokomotif sürücüleri üzerinde yorgunluk tespiti için bir DBN tabanlı model önermişlerdir. Lokomotif sürücünün üst gövdesindeki 12 özellik noktasının koordinatları OpenPose çerçevesi kullanılarak çıkarılmıştır.

Literatürde gerçek zamanlı sürücü yorgunluğu tespiti yapan çalışmalar incelendiğinde, NVIDIA'nın Jetson modelleri en yaygın kullanılan gömülü platformlar arasında yer almaktadır (Salehian ve diğ., 2017; Reddy ve diğ., 2017; Gu ve diğ., 2018; Zhang ve diğ., 2019; Borghi ve diğ., 2017; Mittal, 2019). Salehian ve diğ., (2017) çalışmalarında, gerçek zamanlı görüntü işleme ve bilgisayarlı görme tekniklerini kullanarak yorgunluk takibi gerçekleştirmişlerdir. Çalışmada, göz kırpma eyleminin algılanması amaçlı, şekil analizi ve histogram analizi paralel olarak gerçekleştirilmiştir. Önerilen model, gerçek zamanlıya yakın performans ve yüksek doğruluk oranı elde edebilmek için, NVIDIA Jetson TK1 üzerinde test edilmiştir. Reddy ve diğ. (2017) çalışmalarında, derin öğrenmeye dayalı gerçek zamanlı yorgunluk tespitine yönelik bir yaklaşım önermişlerdir. Çalışmada gerçek zamanlı, yüksek doğruluk ve düşük maliyetli sonuç elde edebilmek amaçlı, model Jetson TX1 üzerinde test edilmiştir. Borghi ve diğ. (2017) çalışmalarında, kafa düşme durumu tahmini, sürücünün davranış ve dikkatini analiz etmek için bir model önermişlerdir. Modelde; 5 konvolüsyon katmanlı, 3 maksimum havuzlu katmanlı ve 2 LSTM modüllü bir RNN kullanılmıştır. Model, NVIDIA Jetson TK1 üzerinde test edilmiştir. Sürücü yorgunluk tespitinden sadece kafa düşme durumu takibinin yeterli olmadığı gözlemlenmiştir. Gu ve diğ., (2018) çalışmalarında, yorgunluk tespit sistemi için çok amaçlı bir hiyerarşik CNN şeması sunarak çok ölçekli havuzlama içeren CNN modeli önermişlerdir. Yorgunluğu tespit

etmek amaçlı PERCLOS ve FOM ölçütleri kullanılmıştır. Bu çalışmada Kafa durum bilgisinin incelenmediği gözlemlenmiştir. Model, NVIDIA Jetson TX2 gömülü platformu üzerinde test edilmiştir. Zhang ve diğ., (2019) çalışmalarında, yorgunluk sürüş tanıma amaçlı, uzay-zamansal özellik dizisine dayanan gerçek zamanlı bir model önermişlerdir. Yazarlar çalışmayı IBUG, LFPW and HELEN veri kümeleri üzerinde eğiterek, sistemi NVIDIA Jetson TX2 gömülü platformu üzerinde test etmişlerdir.

Literatürde verilen çalışmalar incelendiğinde; literatürde halka açık olarak yayınlanan yeteri kadar veri kümesinin bulunmadığı gözlemlenmiştir. Ayrıca yapılan çalışmaların yoğunluklu olarak sadece göz, ağız ya da kafa durum bilgilerine ayrı ayrı odaklandığı gözlemlenmiştir. Bu durum bilgilerinin ayrı ayrı incelenmesi sürücü yorgunluk tespitinde her zaman doğru sonuçlar vermeyebilir. Göz, ağız ya da kafa durum bilgilerinin beraber incelenmesi de büyük önem arz etmektedir.

Bu eksiklikler dikkate alındığında; sürücü yorgunluk/uykululuk hali tespitinin literatürde önemli bir açık problem olduğu gözlemlenmiştir. Bu tez çalışmasında, geçici/sürekli fiziksel özelliklerin, yorgunluk/uykululuk hali üzerindeki etkisinin araştırılması ve araştırma sonucunda yorgun ve/veya uykusuz sürücülerin tespit edilmesi amaçlanmıştır. Bu kapsamda, önerilen davranış tabanlı ölçüme dayanan tahmin sisteminde, özellikle gerçek zamanlı videolar üzerinde sürücünün otomatik olarak takip edilmesi ve sürücünün yüz ifadelerinden yorgun/uykulu ifadelerin ayırt edilmesi hedeflenmiştir. Tez çalışması sonucunda elde edilen ve literatüre kazandırılan özgün çıktılar aşağıdaki gibi ifade edilebilir;

- Ulusal ve Uluslararası Nitelikte Sürücü Yorgunluğu Tespitinde Kullanılabilecek Özgün Yerli Veri Kümesi: Sürücü yorgunluğu ile ilgili tüm yorgunluk durumlarını (çok yorgun, az yorgun, yorgun değil) içeren, farklı etnik gruplardaki sürücü görüntülerinden oluşan, gerçek zamanlı video kayıt veri kümesinin eksikliği önemli bir açıktır. Bu tez kapsamında gönüllü sürücülerden (farklı etnik gruplu) oluşan, KouBM-DFD olarak adlandırılan yeni bir yorgunluk veri kümesi oluşturulmuştur.
- Sürücü Yorgunluğu Tespitinde Kullanılan Çok Görevli Yeni Bir Ağ Modeli: Tez kapsamında yorgunluğun biriktirici bir özelliğe sahip olduğu kabul edilerek, yorgunluğun bir anda oluşmama durumu dikkate alınmış, bu bağlamda tüm

davranışsal tabanlı ölçüm parametreleri de dikkate alınarak yeni ve özgün bir ağ modeli tasarlanmıştır. Tez çalışmasında, çoklu durumlar/semptomlar (birbiri ile ilişkili olabilecek olan göz hareketleri, ağız hareketleri, kafa düşme durumu ve bunların alt hareketleri) dikkate alınmıştır.

Tez çalışması sonucunda elde edilen ve literatüre kazandırılan ara çıktılar aşağıdaki gibi ifade edilebilir.

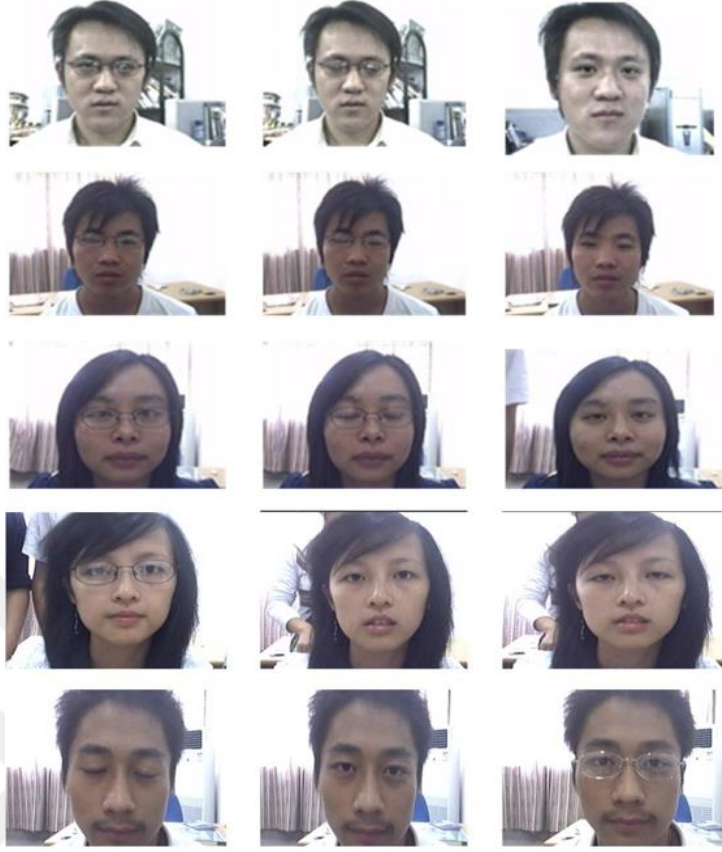
- Sürücü Yorgunluğu Tespitinde Kullanılan Tek Görevli DBN Ağ Modeli: Tez çalışmasında DBN ağ modeli ile sürücü yorgunluğu tahmini yapılırken eğitim ve test görüntüleri üzerinde herhangi bir ön işleme yapılmadan doğrudan yüz verilerinin tamamı kullanılmıştır.
- Gerçek Zamanlı Gömülü Sistem Jetson TX2 Üzerinde Çok Görevli ConNN ve DBN Modelinin Gerçeklenmesi: Tezde önerilen sürücü yorgunluk/uykululuk tahmin sisteminin, gerçek zamanlı uygulamasının test edilmesi amaçlı NVIDIA Jetson TX2 gömülü platformu kullanılarak, ConNN ve DBN modelleri test edilmiştir.

1.1. Kullanılan ve Oluşturulan Veri Kümeleri

1.1.1. ZJU göz kırpma video veri kümesi

ZJU video veri kümesi, 20 farklı kişiden toplamda 80 adet video kaydı alınarak oluşturulmuştur. Videolar Logitech Pro 5000 kamera ile .avi formatında çekilmiştir. Veri kümesi; kişilerin gözlüksüz - önden görünüm, ince kenarlı gözlük takan-önden görünüm, siyah çerçeve gözlük takan-yukarı doğru bakan ve gözlüksüz-yukarı doğru bakan görüntülerini içerecek şekilde 4 farklı senaryodan oluşturulmuştur. Her katılımcıdan, dört senaryo için normal hızda refleks olarak gözünü kırpması istenmiştir. Her bir video, konfigürasyon için 30 fps (Frame Per Second - Saniyedeki Kare Sayısı) ve 320×240 boyutunda yaklaşık beş saniye sürmektedir.

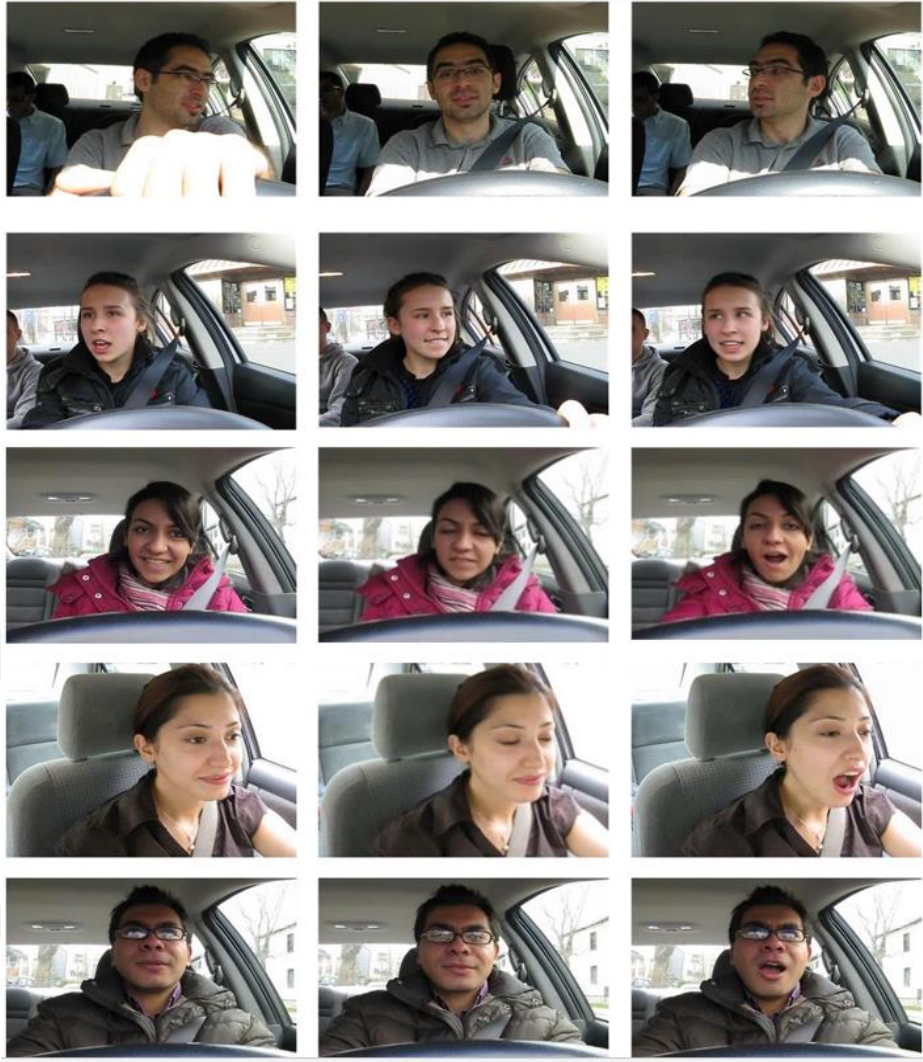
Videolardaki göz kırpma sayıları 1 ile 6 arasında değişmektedir. Veri kümesinde toplamda 255 adet göz kırpma hareketi bulunmaktadır (Pan ve diğ., 2007). Şekil 1.2'de ZJU göz kırpma video veri kümesinden örnekleri gösterilmiştir.



Şekil 1.2. ZJU video veri kümesinden örnek görüntüler

1.1.2. YawDD esneme tespiti video veri kümesi

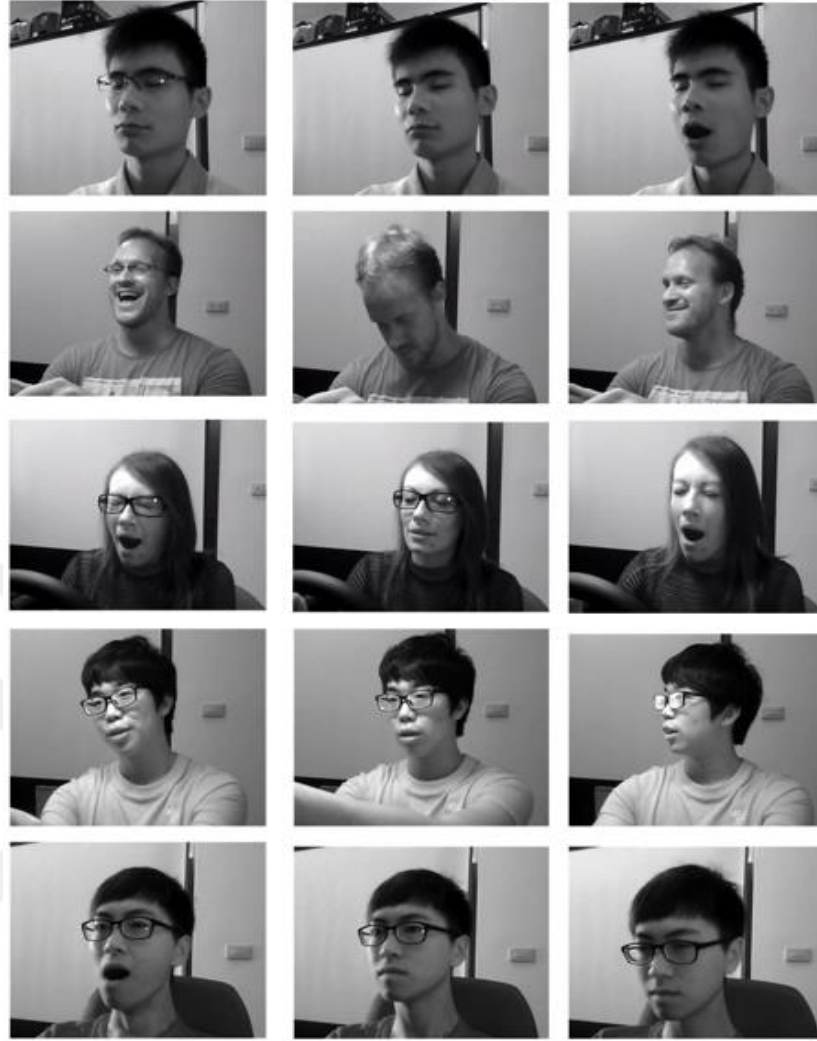
YawDD video veri kümesi (Abathi ve diğ., 2014), araç içi kamera ile kaydedilen, gerçek araç içindeki sürücülerin (erkek ve kadın, gözlüklü/güneş gözlüklü/gözlüksüz, farklı etnik kökenler) konuşmama (sessiz), konuşma, şarkı söyleme ve esneme senaryolarını gerçekleştirdikleri videolardan oluşur. YawDD veri kümesi öncelikle esneme tespiti için algoritma ve modellerin test edilmesinde kullanılmak üzere oluşturulmuştur. Videolar doğal ve değişken aydınlatma koşullarında çekilmiştir. Veri kümesinde, arabanın ön aynasının altına bir kamera yerleştirilmiştir. Her katılımcının üç / dört videosu bulunmaktadır. Her video sessiz, konuşma / şarkı söyleme ve esneme gibi farklı ağız durumlarına sahiptir. Bu veri kümesi, farklı etnik gruplardaki erkek ve kadın sürücülerin gözlüklü ve güneş gözlüklü durumlarını içeren 322 adet videodan oluşmuştur. Şekil 1.3'te YawDD veri kümesinden örnekler gösterilmiştir.



Şekil 1.3. YawDD video veri kümesinden örnek görüntüler

1.1.3. Nthu-DDD sürücü tespiti video veri kümesi

Nthu-DDD video veri kümesi (Weng ve diğ., 2016), NTHU Computer Vision Lab tarafından oluşturulan 36 farklı kişinin sürüş esnasında uykulu videolarını içeren veri kümesidir. Veri kümesinin tamamı normal sürüş, esneme, yavaş göz kırpma hızı, uykuya dalma, gülme gibi çeşitli durumları içeren sürüş senaryolarından oluşmuştur. Tüm veri kümesinin toplam süresi yaklaşık 9 buçuk saattir. Eğitim veri kümesi, farklı 5 senaryoya (gündüz gözlüksüz, gündüz gözlüklü, gece gözlüksüz, gece gözlüklü, güneş gözlüklü) sahip 18 konu içerir. Değerlendirme ve test veri kümeleri, farklı senaryolar altında uykulu ve uykulu olmayan durumları içeren 90 sürüş videosu içerir. Şekil 1.4'te Nthu-DDD veri kümesinden örnekleri gösterilmiştir.



Şekil 1.4. Nthu-DDD video veri kümesinden örnek görüntüler

1.1.4. KouBM-DFD: Kocaeli üniversitesi bilgisayar mühendisliği sürücü yorgunluk tespiti veri kümesi

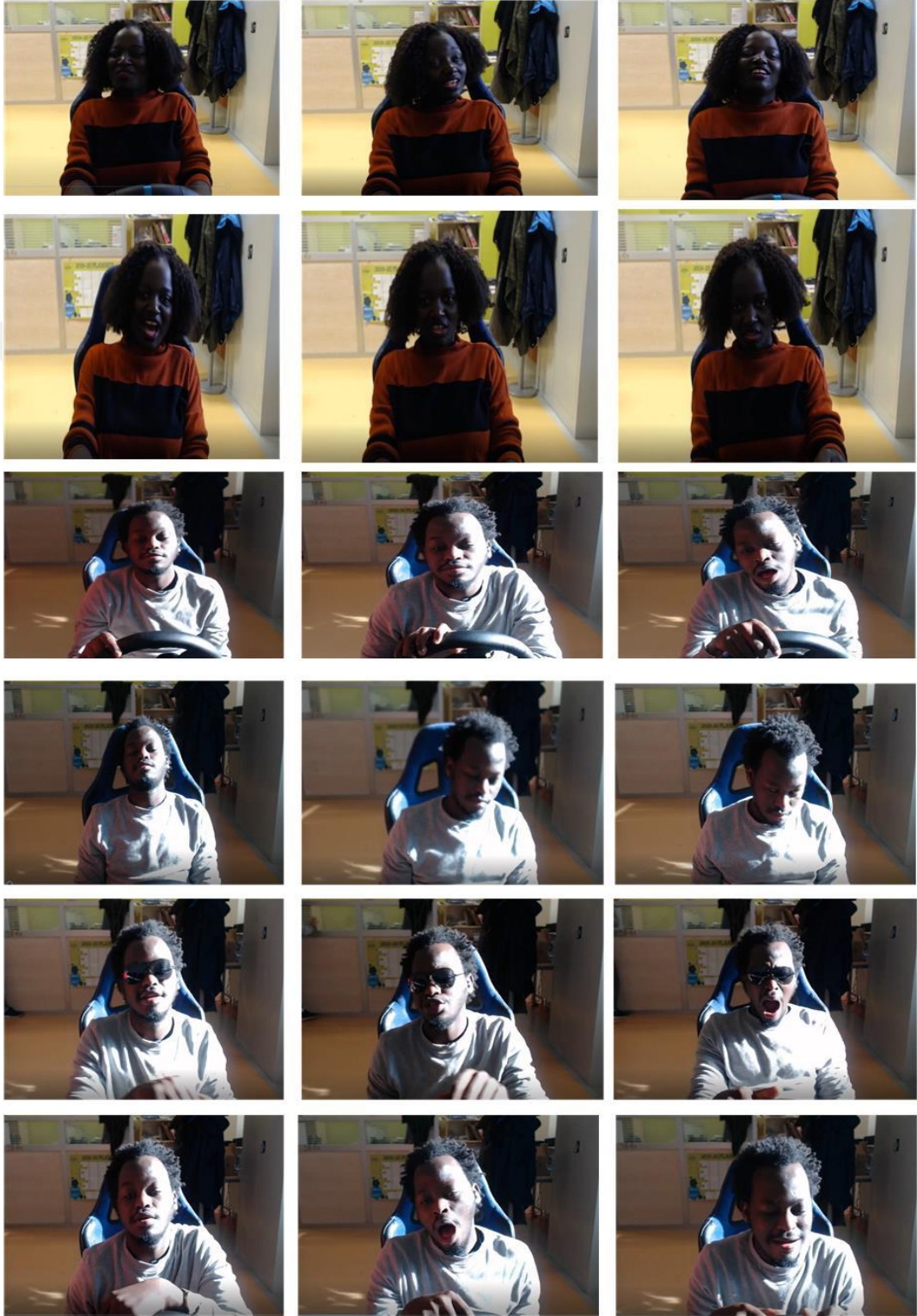
KouBM-DFD (Kocaeli Üniversitesi Bilgisayar Mühendisliği Sürücü Yorgunluk Tespiti) yorgunluk tespiti için algoritma ve modellerin test edilmesinde kullanılmak üzere, göz tanıma ve izleme, ağzı tanıma ve izleme ve sürücü uykululuk durumlarını tahmin etmek için kullanılan çeşitli yüz karakteristiğine sahip video veri kümesidir. KouBM-DFD veri kümesi farklı milletlerden (farklı etnik gruplu) olan kadın ve erkek sürücülerden alınan videolar ile oluşturulmuştur. Bu videolar Kocaeli Üniversitesi Görüntü İşleme ve Bilgisayar Görüsü Araştırma Laboratuvarında simülasyon ortamında çekilmiştir. Her veri kümesinin çekimi için farklı senaryolar ve farklı 40 konu kullanılmıştır. Videolar farklı türlerde yorgun ve yorgun olmayan aktivitelerle

gündüz ve gece dahil olmak üzere farklı durumlardan oluşur. Veri kümesinin tamamı (eğitim, değerlendirme, test veri kümeleri dahil), normal sürüş, esneme, yavaş göz kırpmaya hızı, uykuya dalma, gülme gibi çeşitli gerçek ya da simüle edilmiş videolardan içerir. Videolar 30 fps, 640×480 çözünürlüğünde ve ses verisi olmadan çekilmiştir. Gönüllü sürücüler veri kümesi oluşturmak için hazırlanan bir oyun simülasyon koltuğunda oturup ve simüle edilmiş direksiyon, vites ve pedallarla bir sürüş oyunu oynarlar. Bir deneyci tüm deneyi takip ederek videoları düzenlemiştir. Yorgun ve yorgun olmama durumlarının her ikisi için gönüllülerin her biri sırasıyla ortalama 1, 5 ve 20 dakika süreyle ön kamera (kameranın dikiz aynasına takılacağı düşünülmüştür) ve yan kamera (kameranın ön konsolun soluna monte edileceği düşünülmüştür) tarafından kayda alınmıştır. Yorgunlukla ilişkili semptomlar; esneme, başını sallama ve yavaş göz kırpmaya hızıdır. Yorgunlukla ilgili olmayan semptomlar ise konuşma, gülme ve her iki tarafa da bakmadır. Tez çalışması kapsamında oluşturulan özgün veri kümesinde, toplamda ön kameradan alınan 233 video ve yan kameradan alınan farklı 117 video yer almaktadır. Farklı durum ve semptomları içeren bilgiler EK-A'da verilmiştir. Tez çalışmasında sürücü simülasyonlarında kullanmak için Playseat sürücü yarış koltuğu, Logitech C922, Pro Stream kamera (1080P), Logitech G29 direksiyon, G29vites ve G29 pedal seti, Samsung 21.5 inç LED Monitör Doktora Tezleri Destekleme Projeleri A-2-2 projesi desteği ile satın alınmıştır. Veri kümelerinin oluşturulduğu simülasyon ortamı Şekil 1.5'te gösterilmiştir.

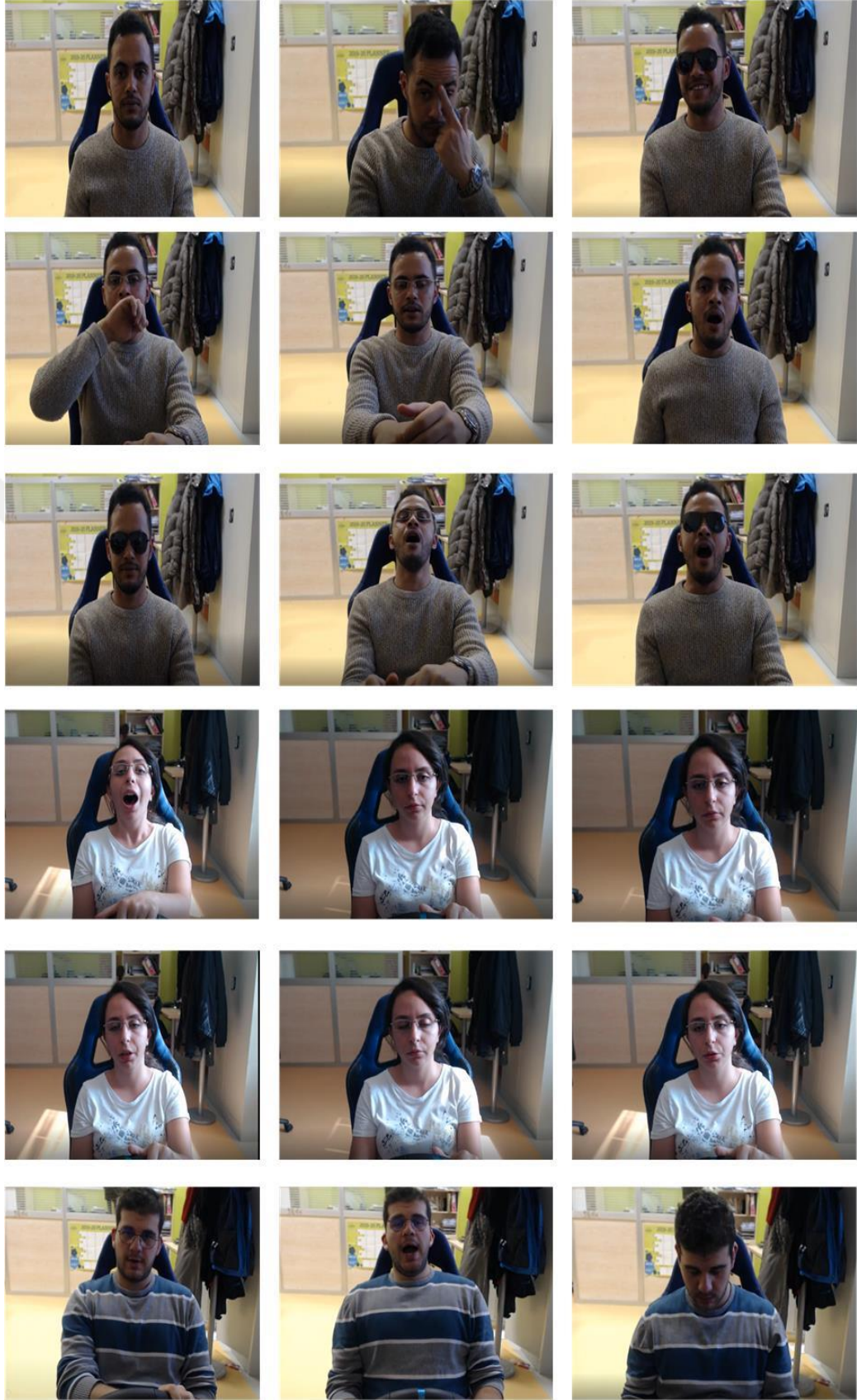


Şekil 1.5. Simülasyon ortamı

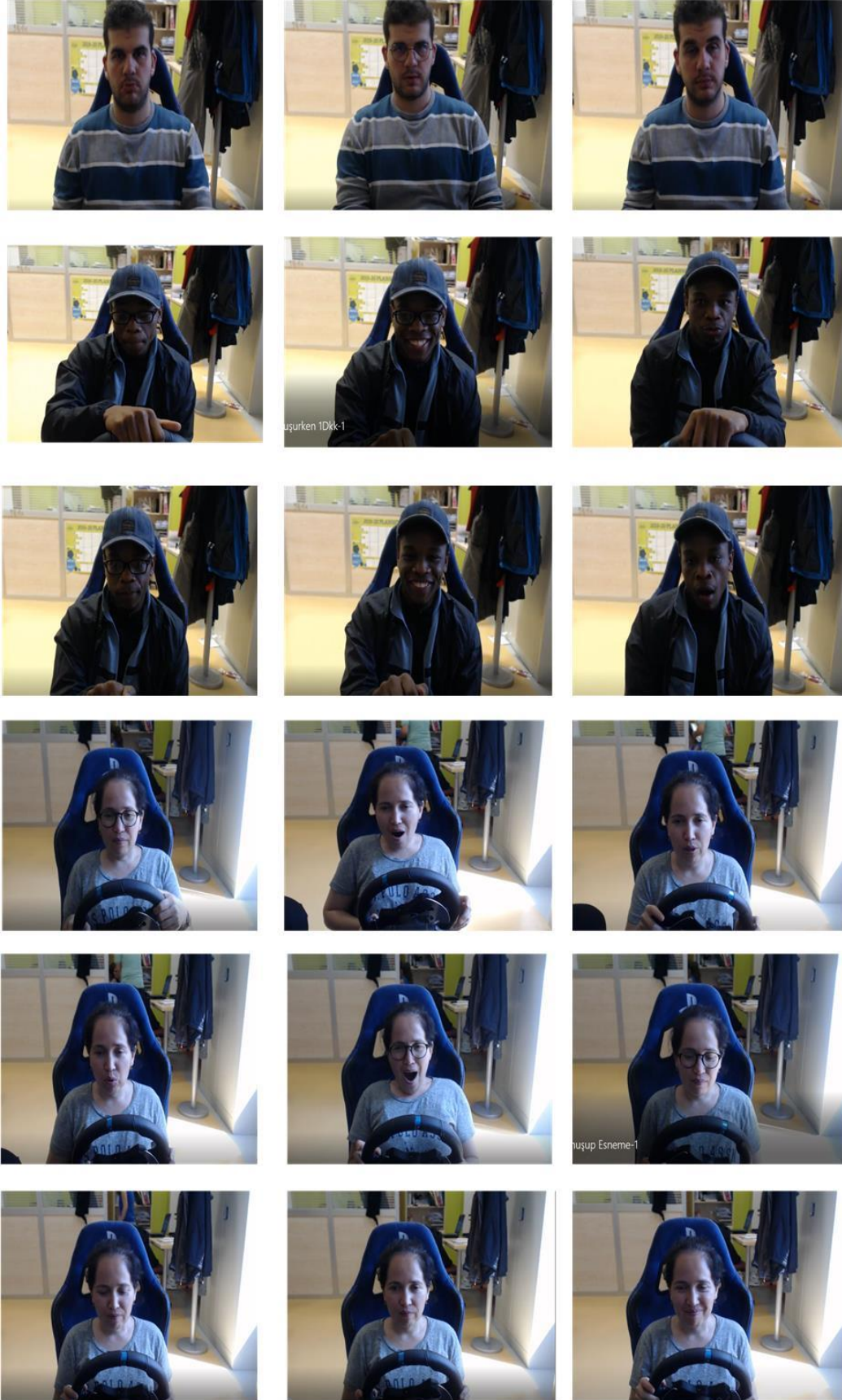
Şekil 1.6’da aracın ön kamerası ile toplanan KouBM-DFD veri kümesi örnekleri, Şekil 1.7’de ise aracın yan kamerası ile toplanan KouBM-DFD veri kümesi örnekleri verilmiştir.



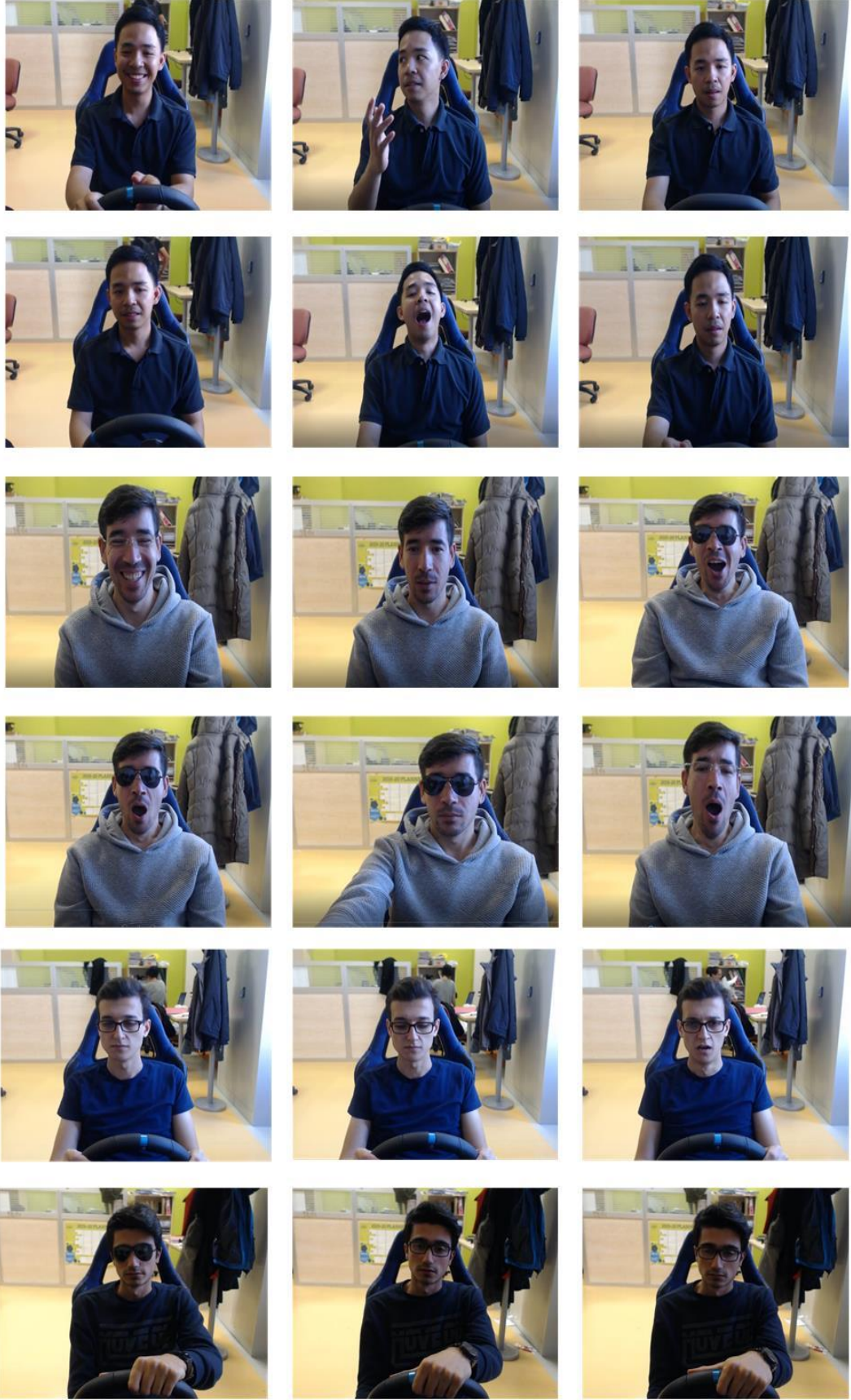
Şekil 1.6. Aracın ön kamerası ile toplanan KouBM-DFD veri kümesinden örnek görüntüler



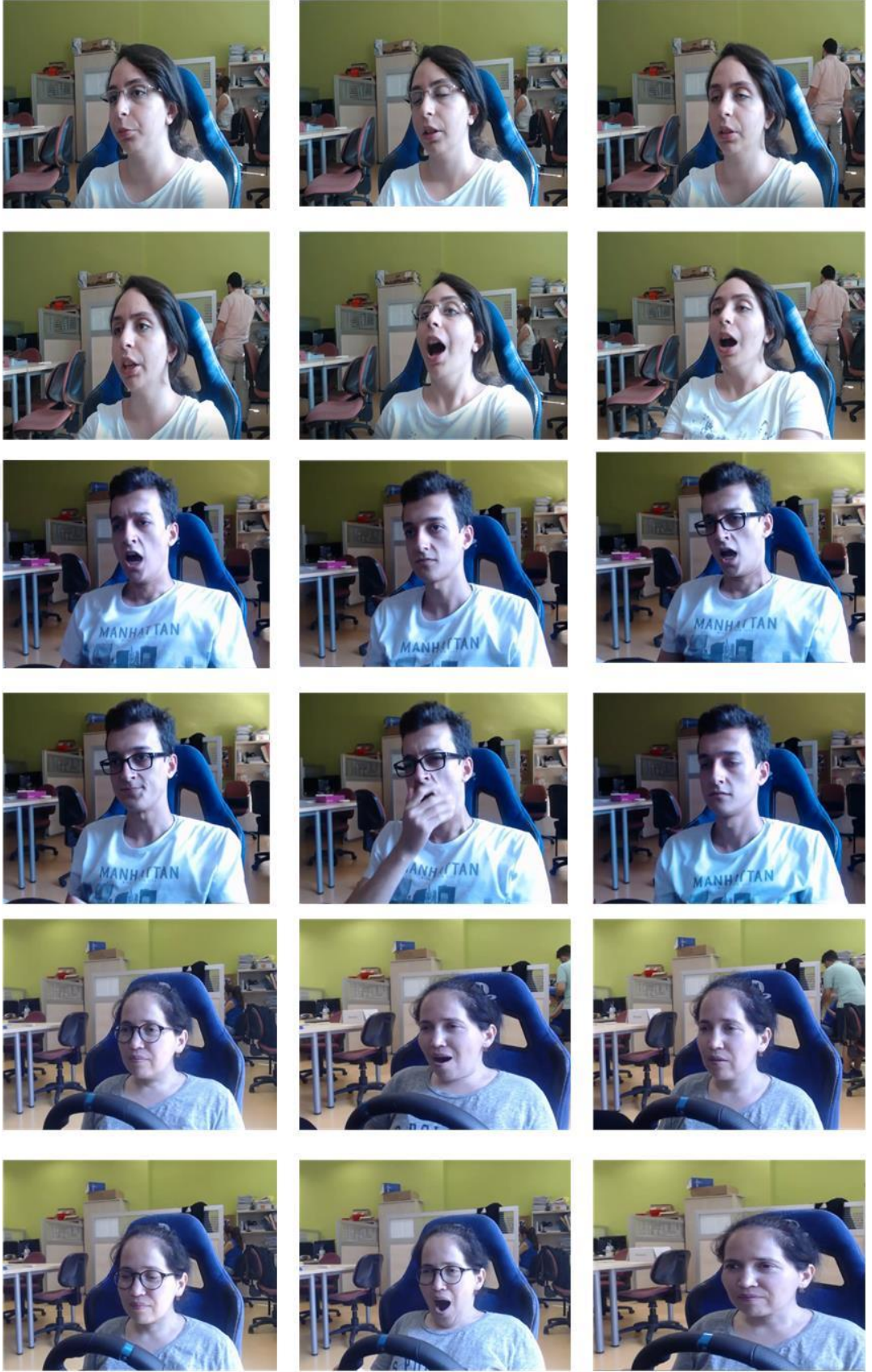
Şekil 1.6. (Devam) Aracın ön kamerası ile toplanan KouBM-DFD veri kümesinden örnek görüntüler



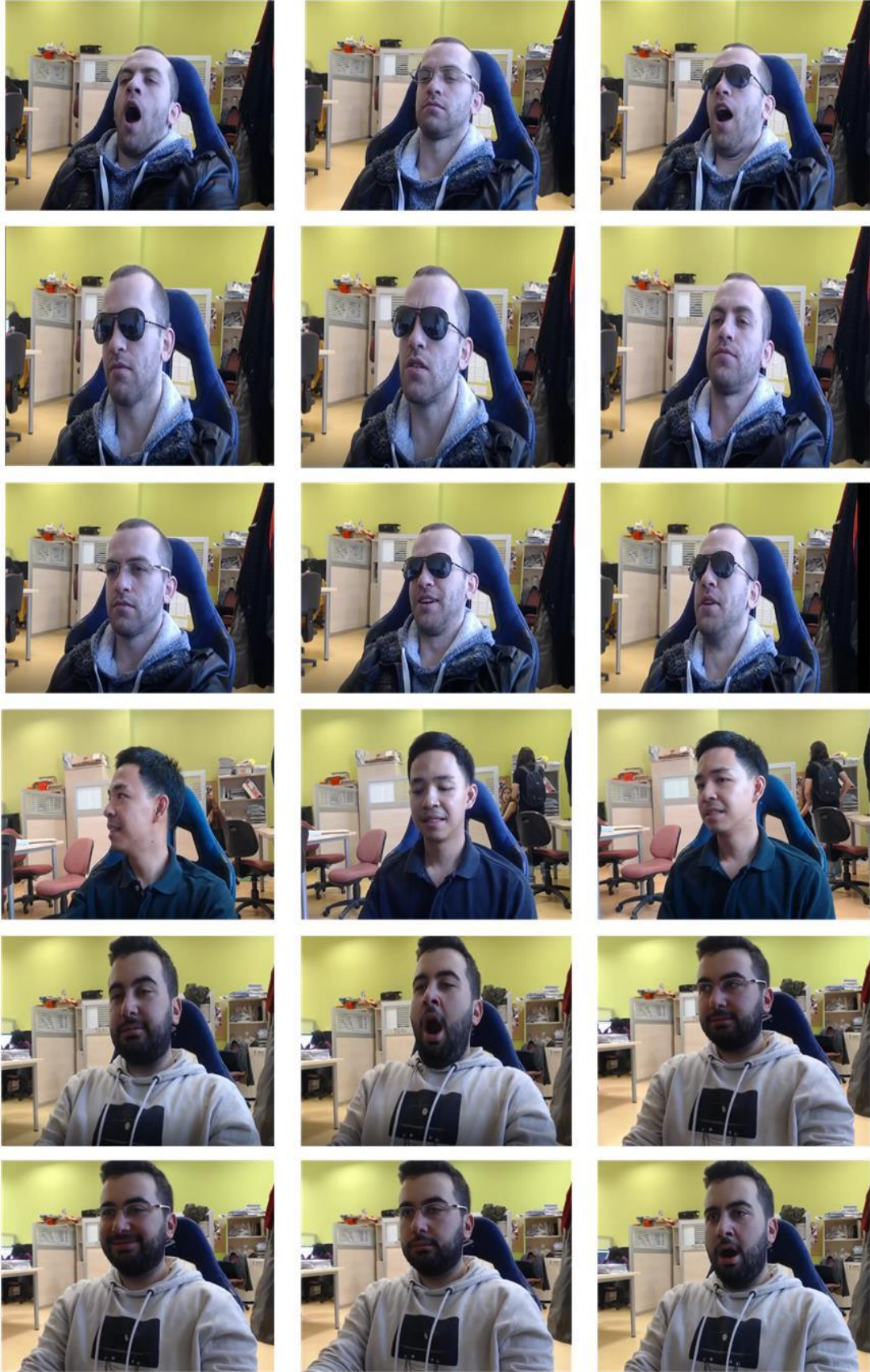
Şekil 1.6. (Devam) Aracın ön kamerası ile toplanan KouBM-DFD veri kümesinden örnek görüntüler



Şekil 1.6. (Devam) Aracın ön kamerası ile toplanan KouBM-DFD veri kümesinden örnek görüntüler



Şekil 1.7. Aracın yan kamerası ile toplanan KouBM-DFD veri kümesinden örnek görüntüler



Şekil 1.7. (Devam) Aracın yan kamerası ile toplanan KouBM-DFD veri kümesinden örnek görüntüler

Tez çalışmasının bundan sonraki bölümünde tez kapsamında kullanılan kütüphaneler hakkında detaylı bilgi verilmiştir.

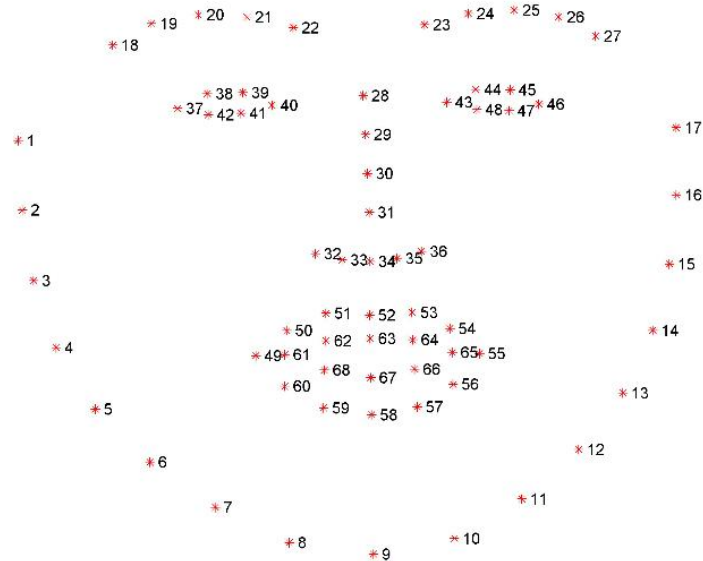
1.2. Kullanılan Kütüphaneler

Tez kapsamında kullanılan Dlib, OpenCV, Tensorflow, Pytorch kütüphaneleri ile ilgili bilgiler aşağıdaki kısımlarda verilmiştir. Bu kütüphaneler ile ilgili kapsamlı bilgiler Ek B’de sunulmuştur.

1.2.1. Dlib

Dlib kütüphanesi (Dlib, 2019) C++ programlama dili ile geliştirilmiş ve 2002 yılından bu yana geliştirilmeye devam eden içerisinde makine öğrenimi, derin öğrenme ve bilgisayarlı görü algoritmalarını barındıran açık kaynak kodlu bir kütüphanedir. C++ ve Python uygulama programı arayüzü sayesinde de Python programlama dili ile uygulama geliştirilebilir. Dlib Raspberry, Tinkerboard gibi gömülü donanımlar veya Android, IOS gibi mobil platformlar üzerinde çalışabilir.

Dlib kütüphanesinde önceden eğitilmiş yüz işareti detektörü, yüz yapılarıyla eşleşen p_τ noktalarıyla temsil edilmektedir. p_τ burada yüzün konumunu tahmin etmek için kullanılan koordinat çiftine (x, y) karşılık gelmektedir. Dlib yüzey göstergesi öngörücüsünün eğitildiği 68 nokta iBUG 300-W veri kümesinin bir parçasıdır. Şekil 1.8’de Dlib kütüphanesi ile bulunan yüz noktaları gösterilmiştir.



Şekil 1.8. Dlib kütüphanesi ile bulunan yüz noktaları

1.2.2. OpenCV

Intel tarafından geliştirilen OpenCV, örüntü tanıma ve görüntü işleme konularında yaygın olarak kullanılır (Hong ve diğ., 2007; Raman ve diğ., 2018). Aynı zamanda OpenCV bir açık kaynak kütüphanesidir ve C / C ++, Java ve Python arayüzleri C dilinde yazılmış olup, birçok platformda (Windows, Linux vb.) çalışır.

1.2.3. TensorFlow

Google tarafından uzun yıllardır geliştirilen TensorFlow kodları 2015 yılında herkesin erişimine açılmıştır. TensorFlow açık kaynak kodlu bir derin öğrenme kütüphanesidir. TensorFlow platform farketmeksizin hesaplamaları, bir veya birden fazla CPU, GPU kullanarak kodlamaya olanak sağlar. Temelinde Python kullanarak geliştirilen bu framework, Python 'ın yanısıra Javascript, R, Swift gibi birçok dili destekler.

1.2.4. Pytorch

PyTorch makine öğrenme algoritmaları için kapsamlı destek sunan bilimsel hesaplama yapısıdır. C/CUDA uygulaması temelini kullanan verimli bir yapıya sahiptir. Derin öğrenme tabanlı araştırmalarda en yaygın kullanılan platformlardan biridir. Güçlü bir GPU desteğiyle vektör hesaplamaları esnek bir şekilde ve hızlı olarak gerçekleştirir.

2. KULLANILAN SINIFLANDIRMA YÖNTEMLERİ

Bu bölümde, sürücü yorgunluğu/uykululuğu tespiti için kullanılan veri kümeleri üzerinde doğru sınıflandırma yapmak amaçlı kullanılan sınıflandırıcılardan bahsedilmiştir. Belirli bir uygulama için tek tip bir sınıflandırma yöntemi yoktur, her sınıflandırıcının güçlü ve zayıf yönleri bulunmaktadır. Sınıflandırıcı performansını veriler/veri kümeleri belirler. Bu nedenle en iyi yaklaşım, farklı modelleri farklı ve aynı veri kümeleri üzerinde değerlendirmektir. Bu çalışmada, farklı dört sınıflandırıcı kullanılmıştır: Adaboost sınıflandırıcısı (Viola ve diğ., 2001), SVM sınıflandırıcısı (Diddi ve diğ., 2015; Mbouna ve diğ., 2013), ConNN modeli (Yan ve diğ., 2016; Yuen ve diğ., 2016; Park ve diğ., 2016) ve DBN (Zhao ve diğ., 2017; Ma ve diğ., 2018; Zheng ve diğ., 2019) modelidir. Adaboost ve SVM algoritmaları ile ilgili ayrıntılı bilgiler Ek C’de verilmiştir

2.1. Adaboost

Adaboost, Viola ve Jones (Viola ve diğ., 2001) tarafından elde edilen peş peşe zayıf sınıflandırıcıları eğitmek için kullanılmıştır. Freund ve Schapire (Freund ve diğ., 1996) tarafından 1995 yılında önerilen bu yöntem, her bir öznelik üzerinde bir zayıf sınıflandırıcıyı eğiterek, seçtiği bazı zayıf sınıflandırıcıların doğrusal birleşiminden kuvvetli sınıflandırıcının oluşturulmasını sağlar. Eğitim sırasında her bir öznelik üzerinde, yanlış sınıflandırılmış her örneğin ağırlıkları artırılarak yeni sınıflandırıcı oluşturulur. Böylece hata ve ağırlıklar güncellenerek ve sistemin eğitilmesi sağlanmış olur (Friedman ve diğ., 2000).

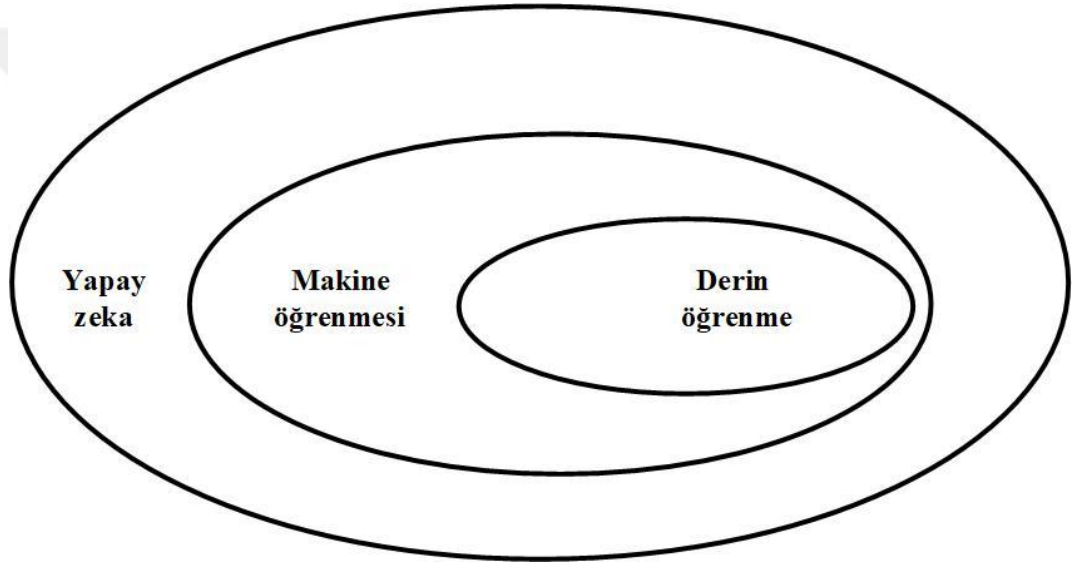
2.2. Destek Vektör Makineleri (SVM)

Sınıflandırma başta olmak üzere kümeleme ve bağlanım problemlerinde kullanılan bir makine öğrenmesi modelidir (Vapnik, 1995). SVM modelinin amacı, hedef değişkene ait sınıfları birbirinden en uygun şekilde ayıracak üst düzlemi tespit etmektir. Literatürde sürücü yorgunluk/uykululuk tespiti için tercih edilen algoritmalarından biridir (Diddi ve diğ., 2015; Mbouna ve diğ., 2013; Sacco ve diğ., 2012; Rivera ve

diğ., 2013; Alioua ve diğ., 2016).

2.3. Derin Öğrenme

Derin öğrenme, yapay zekâ geliştirmede kullanılan en popüler yaklaşımlardan biridir. Derin öğrenme yöntemi, giriş olarak verilen veri kümesi ile sonuçları tahmin etmeyi sağlayan ve birden fazla katmandan oluşan bir makine öğrenme yöntemidir. Yapay zeka, makine öğrenmesi ve derin öğrenme birbiri ile ilişkili ancak anlamları birbirinden farklıdır. Bu 3 kavramın birbiri ile ilişkisi aşağıdaki Şekil 2.1’de verilmiştir (Goodfellow ve diğ, 2016).



Şekil 2.1. Derin öğrenme kavramlarının birbiri ile ilişkisi (Goodfellow ve diğ, 2016)

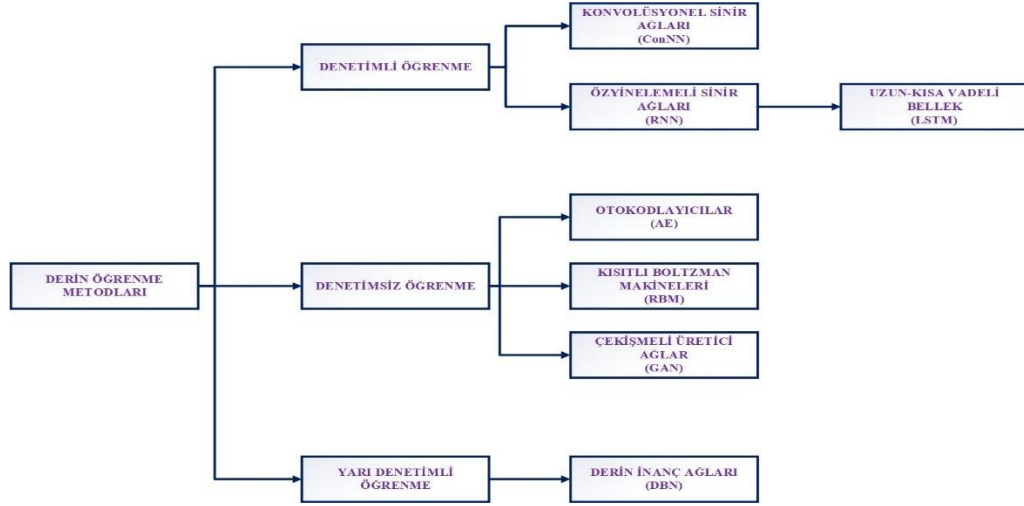
Yapay zeka sistemleri, kendisine verilen görevleri yerine getirebilmek için insan zekasını taklit eden sistemlerdir. Bu sistemler topladıkları bilgilere göre kendini sürekli günceller ve geliştirir. Geniş kapsama sahip olan Yapay zeka sistemleri, oluşabilecek problemlere göre alt kümelere ayrılmıştır. Bu alt kümeler de kendi problemini çözmek için ayrı bir çalışma alanı (planlama ve zamanlama, makine öğrenmesi, doğal dil işleme, vb. gibi) haline gelmiştir. Makine öğrenmesi ise bir veri kümesi üzerinde matematiksel ya da istatistiksel işlemler yaparak tahminlerde bulunur veya sınıflandırılan sistemleri bilgisayar vasıtası ile modeller (Goodfellow ve diğ, 2016). Yukarıdaki bilgiler ışığında Yapay Zeka, ‘tüm bu kavramların genel teknolojisi’, makine öğrenimi de onu destekleyen, hesaba dayalı yöntemlerinin

uygulanmasıdır. Derin öğrenme ise bir makine öğrenmesi yöntemidir. Derin öğrenme bilgisayarların basit kavramlardan daha karmaşık kavramlar kurmasına olanak sağlar (Goodfellow ve diğ., 2016). Derin öğrenmede doğru özelliklerin seçimi sınıflandırma aşaması için önemlidir (Lecun ve diğ., 2015). Özellik çıkarma için bir çok doğrusal olmayan işlem birimi katmanı kullanır. Derin öğrenme yöntemlerinin şu anda var olan makine öğrenmesi yöntemlerinden farkı; büyük miktardaki veriyi aynı anda kullanabilmesi ve bu büyük veriyi işleyebilecek çok yüksek hesaplama gücü olan donanımlara ihtiyaç duymasındır. Son yıllarda özellikle örüntü ve görüntü işleme alanlarında etiketli veri sayısı milyonlarla ifade edilir. Ekran kartı temelli hesaplama gücünün önemli olduğu alanlarda derin öğrenme yöntemleri oldukça yaygınlaşmıştır (Rende ve diğ., 2016). Büyük miktarlarda etiketlenmiş eğitim verilerinden özellik saptama yapabilen sistemler oluşturmak için çok seviyeli “derin” sinir ağlarının kullanılması önemli derecede yatırım ve araştırmanın yapıldığı bir alandır.

Örüntü, görüntü, ses ve konuşma analizi dahil olmak üzere çeşitli uygulamalarda daha hızlı ve performansı yüksek sonuçlar elde etmek için GPU’lu (Grafik İşlemci Ünitesi) sistemler tercih edilir. Özellikle yüksek veri ve hesaplama gücü olanaklarına sahip Microsoft, Google, Facebook, Baidu, NVIDIA gibi büyük teknoloji firmaları tarafından derin öğrenme konuları üzerinde çalışmalar yürütülmektedir. Aynı zamanda bu firmalar kendi Derin öğrenme yazılım kütüphanelerini geliştirmekte ve bunları geliştirici topluluklara da açarak bu alandaki ilerlemeye de destek olmayı amaçlamaktadırlar.

Derin öğrenme metotları öğrenme durumlarına göre denetimli, denetimsiz ve yarı denetimli öğrenme olarak 3 bölüme ayrılır.

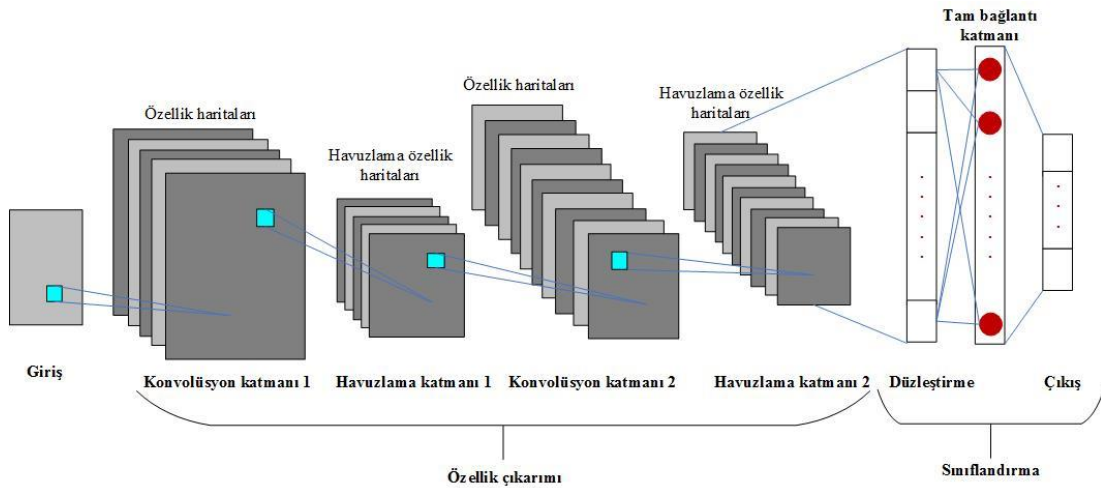
ConNN, RNN denetimli öğrenmeye, DBN yarı denetimli öğrenmeye, oto-kodlayıcılar, Kısıtlı Boltzman Makineleri (Restricted Boltzman Machine-RBM), Çekişmeli Üretici Ağlar (Generative Adversarial Network - GAN) denetimsiz öğrenmeye (Goodfellow ve diğ., 2016; Wu ve diğ., 2020) örnek olarak gösterilebilir. Şekil 2.2’de derin öğrenme metotları verilmiştir. Yapılan tez çalışmasında sürücü yorgunluğu tespiti için derin öğrenme modellerinden olan ConNN ve DBN kullanılmıştır.



Şekil 2.2. Derin öğrenme metotları

2.4. Konvolüsyonel Sinir Ağları (ConNN)

Konvolüsyonel Sinir Ağları ya da tez çalışmasında ConNN olarak adlandırılan Konvolüsyonel ağlar veriyi işlemek için kullanılan özel bir tür sinir ağıdır. ConNN'ler pratik uygulamalarda (görüntü sınıflandırma, bölütleme, nesne tespiti, vb.) oldukça başarılı sonuçlar vermiştir (Lecun ve diğ., 2015; Krizhevsky ve diğ., 2012). ConNN, bir veya daha fazla konvolüsyonel, havuzlama katmanı ve bu katman(lar)dan sonra standart çok katmanlı bir sinir ağı katmanlarından oluşur (Lecun ve diğ., 2015). ConNN modeli özellik çıkarımı ve sınıflandırma isimli iki kısımdan oluşur. Bu kısımlar içinde konvolüsyon, aktivasyon, havuzlama, düzleştirme, tam bağlantı ve çıkış katmanları bulunur. Genel ConNN modeli (Yan ve diğ., 2015) Şekil 2.3' de gösterilmiştir.



Şekil 2.3. Genel ConNN modeli (Yan ve diğ., 2015)

2.4.1. Konvolüsyon katmanı

ConNN modelinde konvolüsyon katmanı uygulanan işaretin işleme alındığı ilk katmandır. Bu tez çalışmasında giriş bir görüntüdür. Konvolüsyon katmanı işaretin özelliklerini algılamaktan sorumlu katmandır. Görüntü durağan/sabit olma özelliğindedir. Bu özelliğinden dolayı bir görüntünün içinde bulunan bazı kısımların (düşük/yüksek seviyeli) istatistiklerinin diğer bölümlerle aynı olduğu kabul edilir. Kısaca görüntünün bir kısmında bulunan bu özellikler, görüntünün diğer kısımlarında da bulunabilirler. Bu katman, görüntüde bulunan bu düşük ve yüksek seviyeli özellikleri çıkarmak için görüntüye bazı filtreler uygular (Goodfellow ve diğ., 2016). Bu filtreleme işlemine konvolüsyon işlemi denilir ve bu işlem sonucunda özellik haritası (aktivasyon haritası) oluşur. Konvolüsyon katmanında kullanılan filtreler birbirinden farklı boyutlarda ya da yapılarında olabilir.

Konvolüsyon işlemi, birim dürtü yanıtı $l(k)$ olarak bilinen bir sistemin, $x(k)$ giriş işaretine karşılık üreteceği $s(k)$ çıkış işaretini zaman uzayında bulmaya yarayan bir işlemdir (Hsu, 1995). Konvolüsyon işlemi yıldız işareti ile gösterilir ve bir boyutlu işaretler için konvolüsyon işlemi Eşitlik (2.1)'deki (Goodfellow ve diğ., 2016) gibi gösterilir.

$$s(k) = x(k) * l(k) \quad (2.1)$$

Burada x giriş işaretini ve l konvolüsyon çekirdeğini (kernel) ifade eder. Zaman indeksi k sadece tamsayı değerleri aldığıında, ayrık konvolüsyon Eşitlik (2.2)'deki gibi (Goodfellow ve diğ., 2016) gösterilir.

$$s(k) = x(k) * l(k) = \sum_{m=-\infty}^{\infty} x(m)l(k-m) \quad (2.2)$$

Görüntüler için girdi bir veri dizisidir ve görüntü boyutu arttıkça bu veri dizisi boyutu da artmaktadır. Bu sebeple çok boyutlu işaretler için konvolüsyon işlemi Eşitlik (2.3)'teki (Goodfellow ve diğ., 2016) gibidir.

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i-m, j-n) \quad (2.3)$$

İki boyutlu I görüntü girdisi ve K iki boyutlu filtredir. Konvolüsyon işlemi sırabagimsiz işlemdir. Bu sebeple Eşitlik (2.4)'te verilen (Goodfellow ve diğ., 2016) denklemlerle eşdeğerdir.

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i-m, j-n)K(m, n) \quad (2.4)$$

Genellikle makine öğrenmesi işlemlerinde Eşitlik (2.4)'ün kullanımı daha kolaydır. Çünkü n değerine göre geçerli m değerleri aralığında daha az varyasyon vardır. m değerinin artmasıyla, girdinin indeksi de artar, ancak filtre indeksi azalır. Filtreyi çevirmenin nedeni konvolüsyon işleminin sıra bağımsız özelliğini kullanmaktır. Bu da filtreyi girdiye göre çevirdiğimiz anlamına gelir. Filtre simetrik olduğunda, konvolüsyon ile aynı olan çapraz korelasyon olarak adlandırılan fonksiyon elde edilir ve bu fonksiyonun tanımı Eşitlik (2.5)'te (Goodfellow ve diğ., 2016) verilmektedir.

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i+m, j+n)K(m, n) \quad (2.5)$$

Birçok makine öğrenmesi algoritmasında kullanılan konvolüsyon yapıları simetrik formda olduğundan dolayı, konvolüsyon işlemleri aynı zamanda korelasyon işlemlerine karşılık gelmektedir (Goodfellow ve diğ., 2016).

Konvolüsyon katmanında kaydırma işlemi sırasında filtre boyutu kadar adımlama (stride) ve dolgu (padding) parametreleri de önemlidir. Boşluk parametresi ise kenar değerlerinin işleme nasıl katılacağını ifade etmektedir. Her bir parametre iki boyutlu giriş matrisinde ayrı bir özelliğe odaklanmaktadır.

Adımlama (Stride); görüntü girdisi üzerinde filtrenin kaç piksel kaydırılacağını sayıdır. Adım sayısının artırılması kaydırma adımının daha büyük olmasını sağlar; bu da daha fazla bilginin görmezden gelinmesi anlamına gelir ve sonuçta elde edilecek özellik haritası boyutu daha da küçülecektir.

Dolgu (Padding) İşlemi; konvolüsyon işleminde girdi görüntüsünün kenar bölgelerinde kalan piksel değerlerini de hesaba katabilmek için dolgu (dolgulama) işleminden faydalanılmaktadır. Bu durumda girdi görüntüsünün dış kısmı "0" piksel değerleri ile doldurulur. Bu işleme dolgu işlemi denilir.

Konvolüsyon işlemi örneği Şekil 2.4'te gösterilmiştir. Burada 3×3 boyutundaki giriş görüntüsü (I) üzerine (verilen örnekte giriş görüntüsünün her elemanını işleme katmak için, giriş görüntüsü üzerinde dolgulama işlemi yapılmıştır) 3×3 boyutunda bir filtre (K) uygulanması sonunda, çıkışta 3×3 boyutunda özellik haritası (S) Eşitlik (2.3)'te verilen hesaplama ile elde edilir.

Giriş Görüntüsü (3×3) (I)			Filtre (3×3) (K)		
1	2	3	-1	-2	-1
4	5	6	0	0	0
7	8	9	1	2	1

Giriş Görüntüsü (5×5) (I)				
0	0	0	0	0
0	1	2	3	0
0	4	5	6	0
0	7	8	9	0
0	0	0	0	0

Özellik Haritası		
-13	-20	-17
-18	-24	-18
13	20	17

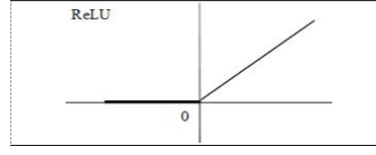
Şekil 2.4. Konvolüsyon işlemi örneği

2.4.2. Aktivasyon katmanı

Aktifleştirme katmanı olarak da bilinen aktivasyon katmanı genellikle konvolüsyon katmanlarından sonra kullanılır. ConNN modeli bu katmandan önce doğrusal bir yapıdadır ve tek bir sinir ağı (perceptron) gibi davranır. Doğrusal yapıların doğrusal olmayan bir duruma dönüşümü aktivasyon fonksiyonları ile olur. Aktivasyon fonksiyonları doğrusal olmayan problemlerin çözümüne yardımcı olurlar ve böylece model daha hızlı öğrenir. ConNN yaklaşımlarında (Krizhevsky ve diğ., 2012; Krause ve diğ., 2014; Simonyan ve diğ., 2014; Zhang ve diğ., 2014) en iyi sonuç Düzleştirilmiş Doğrusal Birim Katmanı (Rectified Linear Units Layer-ReLU) fonksiyonu ile elde edilir. ReLU aktivasyon fonksiyonu Eşitlik (2.6)'da (Glorot ve diğ., 2011) verilmiştir.

$$ReLU(x) = \max(0, x) \quad (2.6)$$

ReLU fonksiyonu negatif girdiler için 0 değerini alırken, x pozitif girdiler için x değerini almaktadır. Şekil 2.5'te ReLU fonksiyon örnek gösterimi verilmiştir.



Şekil 2.5. ReLU fonksiyonu

2.4.3. Havuzlama katmanı

Havuzlama katmanları her bir özellik haritasının boyutunu azaltan ancak önemli bilgileri koruyan bölümdür. Bu katman sayesinde ağırlık hesaplaması hafifler, görüntü üzerindeki gürültüler azalır ve modelin ezberlemesi önlenir. Havuzlama katmanında uygulanan işlem sonucunda üretilen görüntünün boyutu (UGB) Eşitlik (2.7), Eşitlik (2.8), Eşitlik (2.9) ve Eşitlik (2.10)'da verilmiştir.

$$UGB = W_2 \times H_2 \times D_2 \quad (2.7)$$

$$W_2 = \frac{(W_1 - F)}{A + 1} \quad (2.8)$$

$$H_2 = \frac{(H_1 - F)}{A + 1} \quad (2.9)$$

$$D_2 = D_1 \quad (2.10)$$

Burada F filtre boyutu ve A adım sayısıdır. D_1 ve D_2 sırasıyla girdi ve çıktı görüntüsünün derinliğine karşılık gelmektedir. Genellikle kullanılan en popüler havuzlama yöntemi maksimum havuzlama yöntemidir (Liu ve diğ., 2008). Maksimum havuzlama; özellik haritasındaki komşuluk değerlerinden en büyük değeri alarak yeni matrise aktarmaktadır. Maksimum havuzlama kenarlar gibi en önemli özellikleri çıkarmaktadır. Ortalama havuzlama; özellik haritasındaki komşuluk değerlerinin ortalama değerlerini alarak yeni matrise aktarmaktadır. L2-boyutlu havuzlama; özellik haritasında bulunan karelerinin ortalama değerleri alınarak yeni matrise aktarmaktadır.

2.4.4. Düzleştirme katmanı

Sinir ağları, giriş verileri tek boyutlu bir vektördür. Düzleştirme katmanı Tam Bağlantılı Katman'ın girişindeki verileri hazırlar.

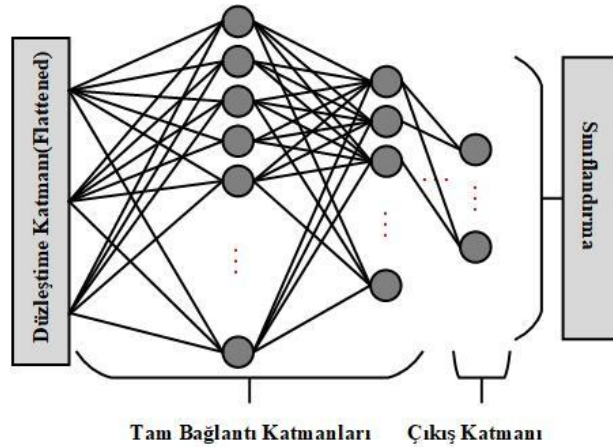
2.4.5. Tam bağlantı katmanı

Tam bağlantı katmanı bir ya da birden çok konvolüsyon ve havuzlama katmanından sonra bulunmaktadır. Tam bağlantı katmanda, tüm katlanma katmanlarından gelen bileşik ve toplanmış en önemli bilgiler bulunur. Bu katmanda sınıflandırma işlemi yapılır. Sınıflandırma işlemi için düzleştirme katmanında gelen vektördeki özellikler iyileştirilir ve ağırlıklar güncellenir. Böylece her bir etiketin özellik vektörlerinden, son olasılık değeri hesaplanır ve girdi görüntüsü sınıflandırılır (Liu ve diğ., 2008). Tam bağlantı katmanının çalışması Eşitlik (2.11) ve Eşitlik (2.12)'de (Toğaçar ve ark., 2020) verilmiştir.

$$Z_i^{(l)} = \sum_{j=1}^{M^{(l-1)}} w_{i,j}^{(l-1)} y_j^{(l-1)} \quad (2.11)$$

$$Y_i^{(l)} = f(Z_i^{(l)}) \quad (2.12)$$

M girdi verisinin boyutu, $(l-1)$ tam bağlantı katmanı ve $Y_i^{(l)}$ i. özellik haritasıdır (Toğaçar ve ark., 2020). Şekil 2.6'da tam bağlantı katmanı gösterilmiştir.

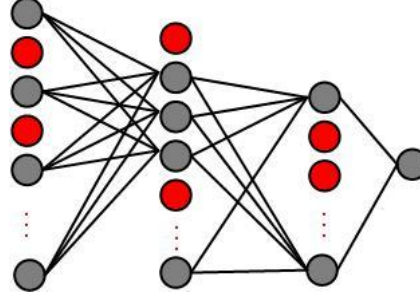


Şekil 2.6. Tam bağlantı katmanı

2.4.6. Seyreltme (DropOut) Katmanı

ConNN'de eğitim işlemi büyük veriler ile yapıldığı için model bazen öğrenme yerine ezberleme yapar. Ezberleme durumunu engellemek için ConNN modeline Seyreltme (DropOut) katmanı eklenir (Hinton ve diğ., 2014). Bu katman model içinde rastgele bazı düğümlerinin işleme katılmamasını sağlar. Böylece ezberleme işleminin önüne

geçilir. Şekil 2.7’de Seyreltme katmanı verilmiştir. Burada kırmızı olarak verilen düğümler işleme alınmayan ve gri olarak verilen düğümler üzerinde işlem yapılan düğümlerdir.



Şekil 2.7. Seyreltme katmanı

2.4.7. Çıkış katmanı

Çıkış katmanı tam bağlantılı katmandan sonra gelmektedir. Bu katmanın çıkış değeri, sınıflandırması yapılacak nesne sayısına eşittir. Bu katmanda farklı sınıflandırıcılar kullanılmaktadır.

Literatürde farklı ConNN modellerinde en sık kullanılan sınıflandırıcı Softmax sınıflandırıcısıdır. Softmax sınıflandırıcısının çıktısında, nesne sayısı kadar 0-1 aralığında değer üretilir. Model için üretilen güçlü tahmin için, vektörde 1'e yakın tek bir girişe sahipken, kalan girişler 0'a yakındır. Zayıf tahmin için ise aşağı yukarı eşit derecede birden fazla olası etiket bulunur (Buduma, 2017).

Eşitlik (2.13)'te (Wani ve ark., 2020) Softmax işlemi verilmiştir.

$$f(x_j) = \frac{e^{x_j}}{\sum_{k=1}^n e^{x_k}} \quad (2.13)$$

Burada x girdi, n sınıf sayısı ve k toplamının indeksidir.

2.5. Derin İnanç Ağları

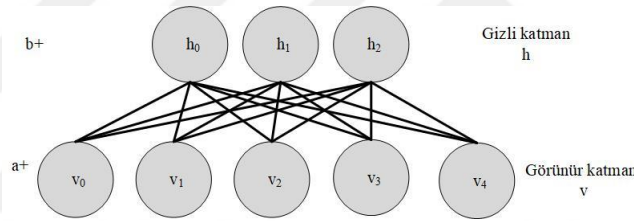
2006 yılında, Hinton ve arkadaşları olasılık tabanlı model kullanılarak veri ve etiketlerin ortak dağılımında RBM üzerinde hesaplama yapan bir model önermiştir

(Hinton ve diğ., 2006). Önerilen bu RBM yapıları birleştirilerek Derin İnanç Ağları (DBN) yapısı oluşturulmuştur (Hinton ve diğ., 2006; Zheng ve diğ., 2019).

2.5.1. Kısıtlı boltzman makineleri

Geoffrey Hinton tarafından icat edilen, RBM sınıflandırma, bağlanım, özellik öğrenme ve konu modelleme için kullanılan bir algoritmadır.

RBM ler iki katmanlı bir stokastik (rastlantısal) ağıdır. Görünür katman v ve gizli katman h şeklinde iki katmandan oluşur. Her görünür düğüm, öğrenilecek olan veri kümesindeki özellikleri tutar. Örneğin, girdi görüntü olduğunda bir görünür düğüm, görüntüdeki piksel değeridir (Le ve diğ., 2008; Salakhutdinov ve diğ., 2007). Görünür katmanda 5 düğümü ve gizli katmanda 3 düğümü olan basit yapıda bir RBM modeli Şekil 2.8’de gösterilmiştir.



Şekil 2.8. RBM modeli

RBM’ler enerji tabanlı modellerdir. Bir RBM enerji fonksiyonu ile tanımlanır. Enerji fonksiyonu Eşitlik (2.14)’te (Hinton, 2010) verilmiştir.

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i \in v} a_i v_i - \sum_{j \in h} b_j h_j - \sum_{i,j} v_i h_j \mathbf{W} \quad (2.14)$$

Burada \mathbf{W} ağırlık matrisi, \mathbf{a} ve \mathbf{b} görünür ve gizli katmanlar için kullanılan yanlılık vektörleridir. RBM modeli bu enerji fonksiyonu ile her görünür ve gizli vektör çifti (\mathbf{v}, \mathbf{h}) için olasılıksal değer verir. Olasılık fonksiyonu, Eşitlik (2.15)’de (Hinton, 2010) verilmiştir.

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v}, \mathbf{h})} \quad (2.15)$$

Burada Z normalizasyon sabitidir ve Eşitlik (2.16)’da (Hinton, 2010) verilmiştir.

$$Z = \sum_{\mathbf{v}, \mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})} \quad (2.16)$$

Z sabiti, görünür ve gizli vektör tüm olası çiftleri toplanarak elde edilmektedir. Buna bağlı olarak görünür vektör olasılığı, tüm olası gizli vektörlerin toplamıdır ve Eşitlik (2.17)'de (Hinton, 2010) verilmiştir.

$$p(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})} \quad (2.17)$$

RBM modelinde, eğitim aşamasında, eğitilen görüntünün enerjisi düşürülür ve diğer görüntülerin, özellikle de düşük enerjiye sahip olanların enerjisi ağırlıklar ve yanlılık değerleri güncellenerek yükseltilir. Böylece eğitim işlemi gerçekleştirilmiş olur. Eşitlik (2.18)'de (Hinton, 2010) eğitim formülü verilmiştir.

$$\frac{\partial \log p(\mathbf{v})}{\partial w_{ij}} = \langle v_i h_j \rangle_{veri} - \langle v_i h_j \rangle_{model} \quad (2.18)$$

Köşeli parantezler ($\langle \rangle$), simge içerisinde belirtilen dağılımdaki beklentileri belirtmek için kullanılır. Bu dağılıma göre en iyi sonucu bulmak için eğitim verileri üzerinde güncelleme yapılır (gradyan azalma olasılığı hesaplanır.) Eşitlik (2.19)'da öğrenme adımına göre güncelleme formülü (Hinton, 2010) verilmiştir.

$$\Delta \mathbf{W} = \alpha \left(\langle v_i h_j \rangle_{veri} - \langle v_i h_j \rangle_{model} \right) \quad (2.19)$$

Denklemden verilen α öğrenme adımı ifade etmektedir. RBM eğitimindeki en önemli noktalardan biri gizli birimler arasında doğrudan bağlantı olmadığı için $\langle v_i h_j \rangle_{veri}$ yanlılık içermeyen verilerin elde edilme zorluğudur. Bunun için de RBM modelinde rastgele seçilen bir eğitim görüntüsü üzerinde, her bir gizli birimin ikili durumlarının belirlenmesi için \mathbf{v} ye bağlı olasılık bulunur. Eşitlik (2.20)'de (Hinton, 2010) gizli birim güncellenme formülü verilmiştir.

$$P(h_j = 1 | \mathbf{v}) = \sigma(b_j + \sum_i v_i \mathbf{W}_{ij}) \quad (2.20)$$

Bir RBM'deki görünür birimler arasında doğrudan bağlantı olmadığından, gizli bir vektör verildiğinde, görünür bir birimin ikili durumlarının belirlenmesi için \mathbf{h} ye bağlı olasılık bulunur. Eşitlik (2.21)'de (Hinton, 2010) görünür birim güncellenme formülü verilmiştir.

$$P(v_i = 1 | \mathbf{h}) = \sigma(a_i + \sum_j h_j \mathbf{W}) \quad (2.21)$$

$\sigma()$ aktivasyon fonksiyodur ve genellikle Sigmoid fonksiyonu olarak seçilir.

2.5.2. Kısıtlı boltzman makinelerinin öğrenmesi

Hinton (Hinton, 2002) RBM modeli için bir öğrenme yöntemi önermiştir. Görünür birimlerin durumlarını bir eğitim vektörü haline getirir ve daha sonra, gizli birimlerin ikili durumlarının tümü, Eşitlik (2.20) kullanılarak paralel olarak hesaplanır. Yeniden yapılandırma olarak adlandırılan (“reconstruction”) işlem ile gizli birimler için ikili durumlar seçildikten sonra, Eşitlik (2.17) ve Eşitlik (2.21) ile verilen işlemlerle her v_i olasılığı 1 olarak belirler. Eşitlik (2.22)'de (Hinton, 2010) yeniden yapılandırmaya bağlı oluşan hata verilmiştir.

$$\Delta \mathbf{W} = \alpha \left(\langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{recon} \right) \quad (2.22)$$

Yeniden yapılandırma, sürekli/ayrık bir değeri bir giriş örneğiyle ilişkilendirmek yerine orijinal girdinin olasılık dağılımını tahmin etmesi nedeniyle bağlanım veya sınıflandırmadan farklıdır. Kısaca öğrenme aşamasında model aynı anda birden fazla değeri tahmin etmeye çalışır. Böylece bir sınıflandırma probleminde meydana gelen ayrımcı öğrenme yerine üretken öğrenme yapılır (girdi bilgilerinin etiketlerle eşleme işlemi). RBM modelinde öğrenme kuralı, Kontrastsal Diverjans (Hinton, 2002) adı verilen bir fonksiyon ile sağlanmaktadır.

Diverjans, bir nokta için tanımlı bir kavramdır. Bir noktaya giren ya da bir noktadan çıkan akı miktarını ile ölçülür. Akının genişleme oranı (pozitif diverjans) ya da akının daralma oranı (negatif diverjans) olarak da belirlenmiştir.

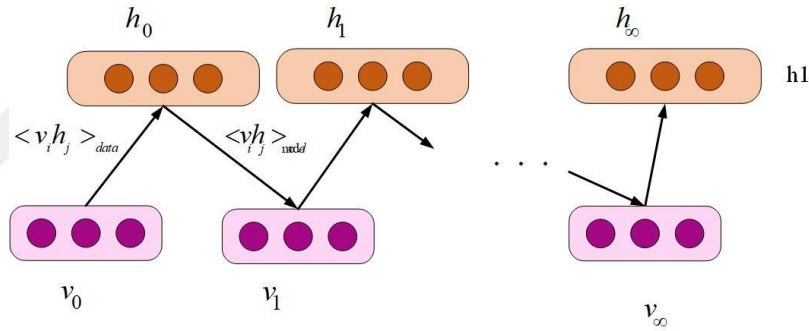
Kontrastsal Diverjans algoritması pozitif ve negatif faz olarak ikiye ayrılır. Pozitif fazda, gizli katmanların ikili durumları, ağırlıkların ve görünür katmanların olasılıkları

hesaplanır. Eğitim verilerinin olasılığı arttığı için bu faza pozitif faz denilmiştir. Negatif fazda ise RBM tarafından üretilen örneklerin olasılığı azalmaktadır. Eşitlik (2.23)'te (Hinton, 2010) Kontrastal Diverjans işlemi verilmiştir.

$$\mathbf{CD}_z(W, \mathbf{v}^{(0)}) = -\sum_{\mathbf{h}} p(\mathbf{h} | v_z) \frac{\partial E(\mathbf{v}_z, \mathbf{h})}{\partial W} + \sum_{\mathbf{h}} p(\mathbf{h} | \mathbf{v}_z) \quad (2.23)$$

CD_z değişkeni, Gibbs örneklemesinin, z , tam adımını kullanır. RBM modeli simetrik yapıdadır ve bütün bir tabaka sadece bir kez örneklendiğinden RBM modelinde Gibbs metodu kullanılmaktadır (Hinton, 2010).

Kontrastal Diverjans ve RBM öğrenme algoritmaları için ayrıntılı matematiksel bilgiye (Kim, 2013)'den ulaşılabilir. Şekil 2.9'da RBM eğitiminde kontrastal diverjans algoritmasında veri ve ters/negatif faz gösterilmiştir.



Şekil 2.9. RBM eğitiminde kontrastal diverjans (Kim, 2013)

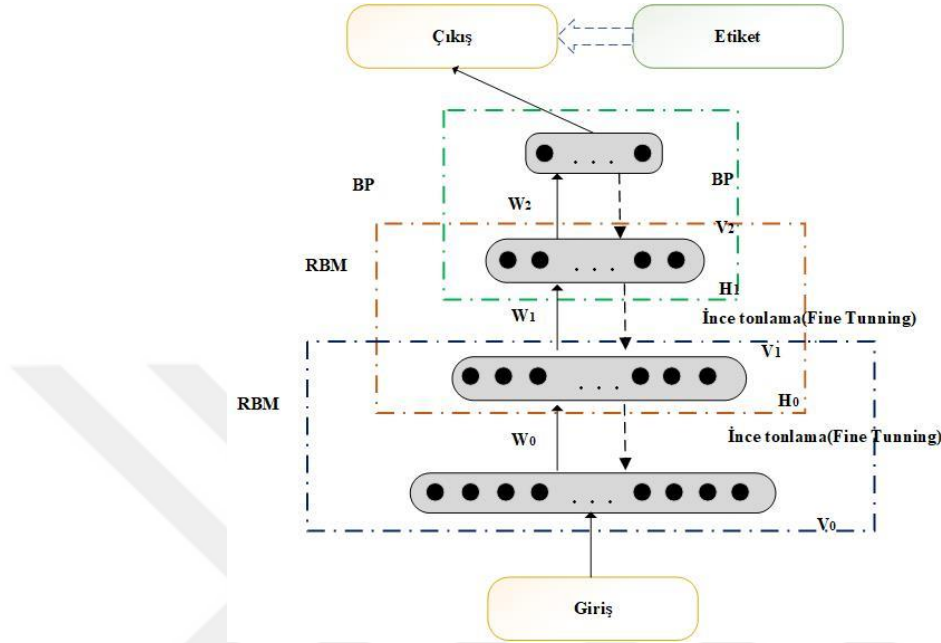
2.5.3. Derin inanç ağları

DBN yarı denetimli bir öğrenme modelidir. Birden fazla RBM modelinin birleşmesiyle DBN modeli oluşur. Her RBM katmanı hem önceki hem de sonraki katmanla bağlıdır. DBN öğrenme algoritması için ayrıntılı matematiksel bilgiye Kim (2013)'ten ulaşılabilir.

DBN modelinde θ öğrenme parametresi, RBM tarafından öğrenilen hem gizli vektörlere bağlı görünür katman üzerindeki dağılımla $p(\mathbf{v} | \mathbf{h}, \theta)$, hem de gizli vektör dağılımı ile $p(\mathbf{h} | \theta)$ güncellenir. Görünür katman vektörü olasılığı, $\mathbf{p}(\mathbf{v})$, Eşitlik (2.24)'te (Mohamed ve diğ., 2009) verilmiştir.

$$p(\mathbf{v}) = \sum_h p(\mathbf{h} | \theta) p(\mathbf{v} | \theta) \quad (2.24)$$

Şekil 2.10'da genel DBN modeli (Hua ve diğ.,2015) gösterilmiştir.



Şekil 2.10. Genel DBN modeli (Hua ve diğ.,2015)

Eğitim aşamasında θ öğrenme parametresi güncellenerek öğrenme sağlanır. Her RBM bloğunda Eşitlik (2.24)'te verilen görünür katman olasılığı güncellenir ve θ öğrenme parametresi gizli katman bilgilerine göre güncellenir ve öğrenme devam eder. Böylece DBN modeli öğrenme işlemini tamamlar (Mohamed ve diğ., 2009, Hinton ve diğ., 2006). Eğitilen DBN modeli; sınıflandırma işlemi için doğrudan ya da iyileştirilerek kullanılabilir. Bu sebeple Hinton ve diğ. (2006) DBN eğitimi bittikten sonra performansı artırmak için İnce Tonlama/İnce Ayar (Fine-Tuning) yöntemini önermişlerdir.

İnce tonlama /İnce Ayar (Fine-Tuning) yöntemi modelde bulunan eğitim hatalarını daha da azaltmak için kullanılır. Tüm ağırlık matrisi yeniden güncellenir. Bu sayede eğitim seti örneklerindeki eğitim hataları azaltılır. İnce ayar yöntemi, her RBM modelinin tüm katmanlarını tek bir model olarak ele alır ve böylece RBM'lerin tüm ağırlıklarının güncellenmesi tek bir iterasyonla gerçekleşir. Bu şekilde, eğitilmiş DBN hızlı bir değişime uğramadan eğitim hataları azaltılır. Eşitlik (2.25)'te DBN modelinde

en çok tercih edilen sınıflandırma yöntemlerinden biri olan Softmax sınıflandırıcı işlemi (Hinton, 2010) verilmiştir.

$$p = \sigma(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + e^0} \quad (2.25)$$

DBN’de eğitilmiş ikili birimleri sınıflandırma kuralı, standart ikili birimler kuralıyla hemen hemen aynıdır. Tek fark her durumun olasılıklarının hesaplanması ve örneklerin alınmasıdır. DBN’deki Softmax sınıflandırıcısı ile ilgili ayrıntılı matematiksel bilgiye Hua ve diğ. (2015)’ten ulaşılabilir.



3. KULLANILAN GÖMÜLÜ SİSTEM

Tek kartlı bilgisayar sistemleri kişisel bilgisayarlardan farklı olarak, basit bir mimari ile yüksek performans gösterebilen sistemlerdir (Süzen ve diğ., 2020). Donanımsal ve yazılımsal geliştirme olanakları sunan ve CPU/GPU içeren çok çeşitli tek kartlı bilgisayar sistemleri (Jetson Nano / TX2, Raspberry Pi, BeagleBoard, Asus Tinker Board vb.) mevcuttur. Tek kartlı bilgisayar sistemleri otonom araç üretiminden, akıllı ev sistemlerine kadar birçok alanda kullanılmaktadır (Yen ve diğ., 1998).

3.1. NVIDIA Jetson

NVIDIA Jetson, NVIDIA'nın ürettiği bir gömülü sistem modelidir. Bu model içerisinde Jetson Nano, TK1, TX1, TX2 vb. gibi farklı amaçlar için kullanılan birçok model bulunur. Jetson'lar düşük güç tüketimi olan sistemlerdir ve makine öğrenimi uygulamalarını hızlandırmak için tasarlanmışlardır (Mittala ve diğ., 2019). Jetson platformu sayesinde; sınıflandırma, örüntü (görüntü/ses/konuşma) tanıma gibi işlemler başka bir platforma ihtiyaç duyulmaksızın; geliştirme kiti üzerinde bulunan yapay zeka ile gerçekleşir. Ayrıca bu platform kullanıcıya güçlü paralel hesaplama olanağı sunar. Böylece bant genişliği kullanımı ve paket gecikmeleri önlenir. Tablo 3.1'de Jetson platformuna ait Jetson Nano / Jetson TX2 / Tam yapay zeka kitlerinin karşılaştırması (Nvidia, 2017) verilmiştir.

Tablo 3.1. Jetson Nano / Jetson TX2 / Tam yapay zeka kiti karşılaştırması (Nvidia, 2017)

	Jetson Nano	Jetson TX2	Tam Yapay Zeka Kiti
Performans	472 GFLOPs	1.3 TFLOPs	24× 1 TFLOPs,
GPU	NVIDIA Maxwell w/ 128 CUDA cores @ 921 MHz	256 NVIDIA CUDA NVIDIA Pascal mimari@ 1300MHz	6,144 GPU CUDA cores with NVIDIA® Pascal™ Maxwell™
CPU	Quad-Core ARM Cortex-A57 64-bit @ 1.42 GHz	2 çekirdekli Denver2 64 bit CPU (@ 2GHz) ve 4 Çekirdekli ARM A57 kompleks	Quad-Core ARM Cortex-A57 @ 2GHz + Dual-Core NVIDIA Denver2 @ 2GHz
Bellek	4 GB LPDDR4 @ 1600MHz, 25.6 GB/s	8GB 128-bit LPDDR4	8GB 128-bit LPDDR4 @ 1866Mhz, 59.7 GB/s
Bağlanılabilirlik	Gigabit Ethernet / M.2 Key E	Gigabit Ethernet	Gigabit Ethernet, 802.11ac WLAN

Tablo 3.1.(Devam) Jetson Nano / Jetson TX2 / Tam yapay zeka kiti karşılaştırması (Nvidia, 2017)

	Jetson Nano	Jetson TX2	Tam Yapay Zeka Kiti
Ekran	HDMI 2.0 and eDP 1.4	2× DSI, 2× DP 1.2 / HDMI 2.0 / eDP 1.4	2× DSI, 2× DP 1.2 / HDMI 2.0 / eDP 1.4
USB	4× USB 3.0, USB 2.0 Micro-B	USB 3.0 + USB 2.0	1× USB 3.1, 2× GbE
Video kodlayıcı	H.264/H.265 (4Kp30)	2×2K@30 (HEVC)	H.264/H.265(4Kp 60)
Video kod çözücü	H.264/H.265 (4Kp60, 2× 4Kp30)	2×2K@30 12 bit destek	H.264/H.265 (4Kp60)
Kamera	MIPI CSI port	12 lanes MIPI CSI-2, D-PHY 1.2 (30 Gbps)	MIPI CSI port
Depolama	16 GB eMMC	32GB eMMC	32GB eMMC
Güç	5W-10W	7.5W-15W	7.5W-15W

Tablo 3.1'deki veriler değerlendirildiğinde; Jetson TX2 modülünün gerçek zamanlı görüntü işleme çalışmalarında Jetson Nano kartından daha güçlü performansa ve Tam Yapay Zeka Kiti modülü kadar hızlı GPU kullanımına sahip olduğu görülmektedir.

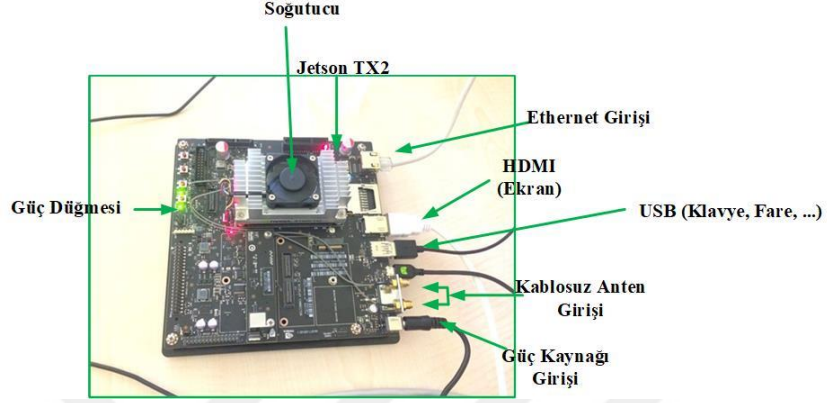
3.2. Jetson TX2

Jetson TX2, NVIDIA firması tarafından özellikle yapay zeka uygulamaları için geliştirilmiş gömülü sistem platformlarından biridir (Nvidia, 2020). İki çekirdekli ARM işlemcisini ve 256 CUDA çekirdekli GPU'yu destekler ve böylece Jetson TX2 platformu kullanıcıya yüksek performansta işlem gücü sağlar (Braun, 2018). Tablo 3.2'de Jetson TX2 kartının; GPU, CPU, bellek, depolama, video kodlama, video kod çözümü, bağlanabilirlik, kamera, ekran, boyut ve mekanik bilgileri verilmiştir.

Tablo 3.2. Jetson TX2 teknik özellikleri

Teknik Özellikler	
GPU	256 NVIDIA CUDA çekirdekli™ NVIDIA Pascal™
CPU	İki çekirdekli Denver 2 64 bit CPU ve dört çekirdekli ARM A57 kompleks
Bellek	8 GB 128-bit LPDDR4
Depolama	32 GB eMMC 5.1
Video Kodlama	2× 4K @ 30 (HEVC)
Video Kod Çözümü	2× 4K @ 30, 12-bit destek
Bağlanabilirlik	Gigabit Ethernet
Kamera	12 lanes MIPI CSI-2, D-PHY 1.2 (30 Gbps)
Ekran	HDMI 2.0 / eDP 1.4 / 2× DSI / 2× DP 1.2
Boyut	87 mm×50 mm
Mekanik	Termal Transfer Plakalı (TTP) 400 pinli konektör

Tablo 3.2'deki veriler dikkate alındığında Jetson TX2 gömülü sistem kartının kullanıcıya yeterli depolama alanı sağladığı, ihtiyaç durumunda ağ bağlantısına izin verdiği ve oldukça yüksek performans sağladığı görülmüştür. Aktif çalışan Jetson TX2 kartı Şekil 3.1'de verilmiştir.



Şekil 3.1. Çalışan Jetson TX2 kartının görüntüsü

4. SÜRÜCÜ YORGUNLUĞU TESPİTİ ÇÖZÜM YAKLAŞIMLARI VE TEZ KAPSAMINDA GERÇEKLEŞTİRİLEN SENARYOLAR

Tezin bu bölümünde, tez çalışmasında geliştirilen sürücü yorgunluk/uykululuk tespiti sistemine ait kapsamlı bilgiler ve tez kapsamında gerçekleştirilen senaryolar (Senaryo 1, Senaryo 2, Senaryo 3, Senaryo 4, Senaryo 5 ve Senaryo 6) detaylandırılmıştır. Tez kapsamında gerçekleştirilen senaryolar literatürde yer alan veri kümeleri ve bunun yanı sıra KouBM-DFD olarak adlandırılan yeni ve özgün bir görüntü veri kümesi ile test edilmiştir. Test aşaması için kullanılan bilgisayar/platform bilgileri aşağıda verilmiştir.

- ASUS dizüstü bilgisayar: NVIDIA GeForce GT540M, CUDA 1GB, Intel Core i7, 4 GB DDR3, Ubuntu 16.04 İşletim Sistemi.
- AWS makinesi (AWS Amazon platformu): vCPUs 16, G3, NVIDIA Tesla M60 GPUs, Hafıza 8 GiB of GPU, ve H.265 (HEVC) 1080p30 akışını (maksimum 10 tane) ve H.264 1080p30 (maksimum 18 tane) akışını destekleyen bir donanım kodlayıcı.
- Python3.6 ve Pythorch kütüphanesi.
- Logitech HD 1080p kamera

4.1. Yorgunluk Parametreleri

Tez çalışmasında sürücü yorgunluk/uykululuk tespiti için davranışsal tabanlı ölçüm kullanılmıştır. Bu kapsamda farklı 3 durum bilgisinden yararlanılmıştır. Bunlardan ilki göz durum bilgisidir. Bu durum göz kırpması ve göz kapanma oranına bağlıdır. PERCLOS belirli bir zaman dilimi içinde sürücü gözünün kapalı olduğu durumların toplam duruma bağlı yüzdesini ifade eder. Bir başka deyişle PERCLOS, açık ve kapalı gözlerin sayısına bağlı olarak kapalı gözlerin oranını gösteren yorgunluk analiz yöntemidir (Viola ve diğ., 2001). Eşitlik (4.1)'de PERCLOS ölçütü verilmiştir.

$$P = \frac{N_c}{N_t} \quad (4.1)$$

Burada N_c ise belirli bir zaman dilimi içinde kapalı göz çerçevelerinin sayısını temsil eder. N_t belirli bir zaman dilimi içinde toplam açık ve kapalı göz çerçevelerinin sayısını ifade eder.

Literatürde bir sürücünün normal şartlar altında dakikada yaklaşık 10 kez göz kırptığı tespit edilmiştir. Sürücünün göz kırpmaya frekansı 10 adet/dakikadan düşükse bu yorgunluk tehlikesi olarak algılanabilmektedir. Eğer bir sürücü yorgun değilse, PERCLOS 15-30 adet/dakika ve her göz kırpmaya 0.25-0.3 saniyedir. PERCLOS metriği, literatürde göz durumuna bilgisi dikkate alınarak yapılan sürücü yorgunluğu tespit çalışmalarında kullanılmıştır (Pratama ve diğ., 2017; Manu, 2016; Han ve diğ., 2015; Darshana ve diğ., 2014; Sheng ve diğ., 2013; Xie, 2012; Daza ve diğ., 2011; Danisman ve diğ., 2010; Freund ve diğ., 1996). Yorgunluk hisseden sürücülerde PERCLOS metriği, yorgunluk hissetmeyen sürücülerinden daha yüksek olacaktır çünkü gözler normalden daha uzun ve daha sık kapanmaktadır. Göz açık/kapalı olma durumu (Darshana ve diğ., 2014) Şekil 4.1’de gösterilmiştir.



Şekil 4.1. Göz açık/kapalı olma durumu (Darshana ve diğ., 2014)

Yorgunluk parametrelerinden ikincisi ağız durum bilgisidir. Bu durum esneme sayısı ve esneme frekansına bağlıdır. FOM belirli bir zaman dilimi içinde (belirli bir periyotta) sürücü ağızının açık olduğu durumların, toplam duruma oranının yüzdesi olarak gösterilen orandır. Eşitlik (4.2)’de FOM metriği verilmiştir.

$$N = \frac{M_c}{M_t} \quad (4.2)$$

Burada M_c ise belirli bir zaman dilimi içinde açık ağız çerçevelerinin sayısını temsil eder. M_t belirli bir zaman dilimi içinde toplam açık ve kapalı ağız çerçevelerinin sayısını ifade eder.

Yorgunluk parametrelerinden sonuncusu kafa düşme durum bilgisidir. Kafa düşme durum bilgisi, sürücünün mevcut ilgi odağının güçlü bir göstergesidir. Sürücü yorgunluk/uykululuk sistemlerinde kafa düşme durum bilgisi güvenilir bir ipucu olarak kabul edilmiştir (Anne ve diğ., 1999; Ji ve diğ., 2004; Lee ve diğ., 2008; Ali ve diğ., 2012; Xie ve diğ., 2012; Borghi ve diğ., 2017). Tablo 4.1’de kafa düşme durum bilgisinin sürücü yorgunluk/uykululuk durumuna etkisinin araştırıldığı literatür çalışmaları verilmiştir.

Tablo 4.1. Kafa düşme durum bilgisi

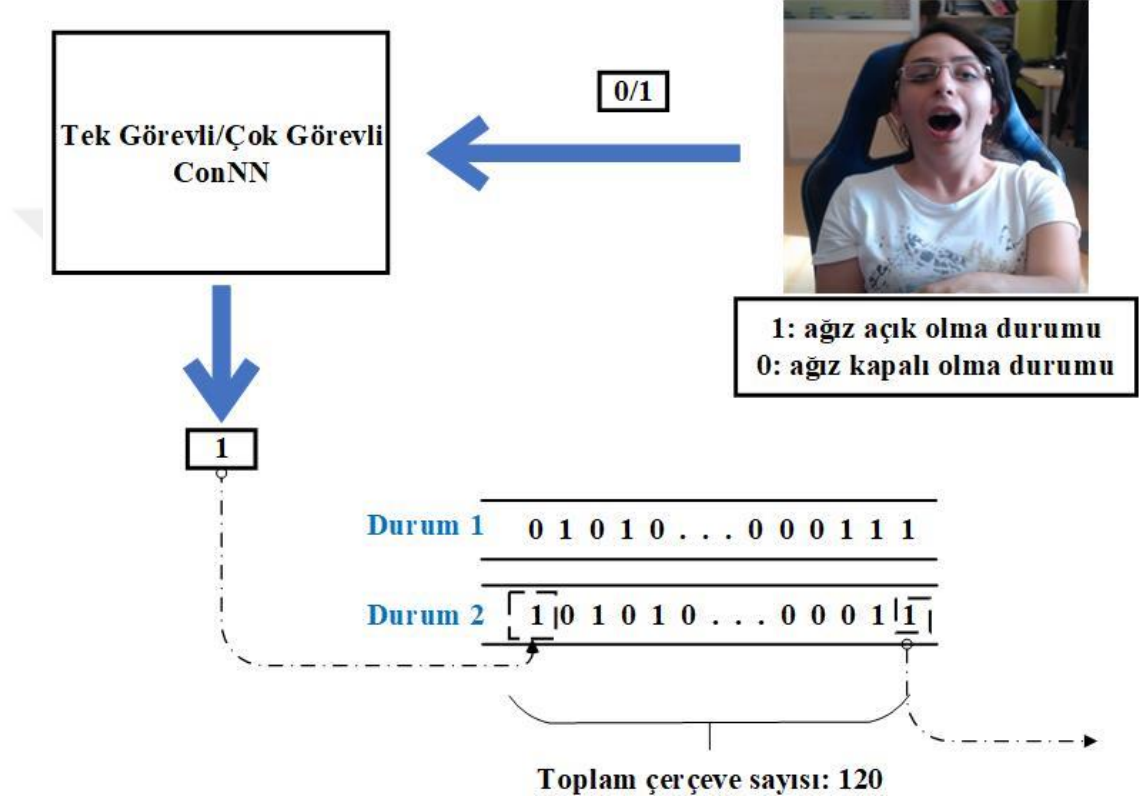
Literatür	Yıl	Tespit	Yöntem
Lavergne ve diğ., 1999	1999	Kafa düşme durum bilgisi	x ve y koordinatları
Ji ve diğ., 2004	2004	Yüz tespiti, kafa düşme durum bilgisi ve göz kırpma tespiti	PERCLOS
Lee ve diğ., 2008	2008	Kafa düşme durum bilgisi	IR Sensor
Ali ve diğ., 2012	2012	Kafa düşme durum bilgisi ve göz kırpma tespiti	x, y ve z koordinatları, PERCLOS
Xie ve diğ., 2012	2012	Kafa düşme durum bilgisi ve göz kırpma tespiti	Adaboost, PERCLOS
Borghi ve diğ., 2017	2017	Kafa düşme durum bilgisi ve göz kırpma tespiti	LSTM, RNN

4.2. Tez Kapsamında Gerçekleştirilen Senaryolar

Tez çalışmasında geliştirilen sürücü yorgunluk/uykululuk tespiti sistemi için farklı 6 senaryo gerçekleştirilmiştir. Tez kapsamında gerçekleştirilen senaryolar, literatürde yer alan veri kümeleri ve bunun yanı sıra KouBM-DFD olarak adlandırılan yeni ve özgün bir görüntü veri kümesi üzerinde test edilmiştir.

Gerçek zamanlı sistemde yorgunluk parametrelerini kayıpsız ve doğru tespit edebilmek için sabit bir frekans aralığı (periyot) gereklidir. Tez çalışması kapsamında bu süre 4 sn (120 çerçeve sayısı) olarak belirlenmiştir. Belirlenen periyotta bulunan çerçeve sayısının sabit olması test aşamasının sürekliliğini sağlar. Tezde kullanılacak frekans aralığı (toplam çerçeve sayısı:120) çalışma içinde sabit tutulur ve her yeni çerçeve işleme girdiğinde en sonda bulunan çerçeve silinir. Böylece, başlangıçta

belirlenen frekans aralığı sabit kalmış olur. Burada elemanları 0 veya 1 olan 120 uzunluklu bir vektör işleme alınmıştır. 0 çerçevede ağzın kapalı olduğunun tespit edildiğini; 1 ise çerçevede ağzın açık olduğunun belirlendiğini gösteriyor. Böylece 120 çerçeveden oluşan giriş 120 elemanlı bir vektöre dönüştürülmüştür. Şekil 4.2’de KouBM-DFD veri kümesinde yer alan örnek bir görüntü (ağız durumunun dikkate alındığı) için çerçeve kuyruğuna görüntünün alınması işlemi ayrıntılı olarak verilmiştir.



Şekil 4.2. Çerçeve kuyruğuna görüntünün alınması

Tezin bu bölümünde tez kapsamında geliştirilen senaryolar karşılaştırmalı olarak anlatılmıştır.

4.2.1. Senaryo 1: Adaboost algoritması ile yorgunluk tahmin sistemi

Senaryo 1’de gündüz koşullarında sürücünün direkt olarak yüzüne dönük bir kamera ile sürücü yorgunluğu tahmin sistemi tasarlanmıştır. Sürücünün yüz bölgesinin tespiti ve takibi için, sınıflandırma ve öznetelik çıkarma işlemlerini aynı anda gerçekleştirebilen Adaboost algoritması kullanılmıştır.

Tasarlanan sistemin blok diyagramı Şekil 4.3’te verilmiştir.



Şekil 4.3. Senaryo 1 blok diyagramı

Eğitim aşamasında pozitif (insan yüzünü içeren görüntüler) ve negatif (insan yüzünün bulunmadığı görüntüler) görüntüler gereklidir. Negatif görüntüler, Huang ve diğ (2007) yılında yaptığı yayında kullanılan veri kümesinden elde edilmiştir. Negatif görüntü örnekleri Şekil 4.4'te verilmiştir.



Şekil 4.4. Negatif görüntü örnekleri

Pozitif görüntüler için ise laboratuvar ortamında kamerayla çekilen videodaki görüntüler kullanılmıştır. 24×24 boyutunda olan pozitif görüntü örnekleri Şekil 4.5'te verilmiştir.



Şekil 4.5. Pozitif görüntü örnekleri

Tüm görüntüler Adaboost Algoritması ile eğitilmiş ve test için gerekli olan .xml dosyaları elde edilmiştir. Elde edilen .xml uzantılı dosyalar OpenCV kütüphanelerinde bulunan sınıflandırıcıları eğitmek için kullanılmıştır. Test aşamasında anlık olarak çekilen video görüntüsü üzerinde bu sınıflandırıcılar kullanılarak yüz bölgesinin tespiti yapılmaktadır. Senaryo 1 için yüz bölgesi tespiti Şekil 4.6'da verilmektedir.



Şekil 4.6. Senaryo 1 için yüz bölgesi tespiti

Yüz bölgesi tespit edildikten sonra göz bölgeleri tespit edilerek görüntüden çıkarılmıştır. Açık ve kapalı göz örnek görüntüleri Şekil 4.7'de verilmiştir.

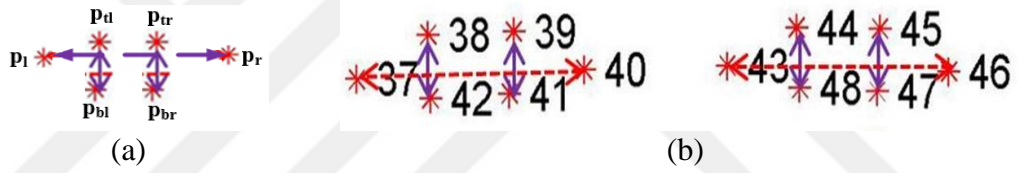


Şekil 4.7. Açık ve kapalı göz örnek görüntüleri

Geliştirilen sistemde sürücü yüzünden alınan görüntünün değerlendirilmesi sonucunda, PERCLOS metriği hesaplanarak sürücünün yorgunluk durumu/derecesi bulunmuştur. Sürücünün uykuya dalma durumu: çok yorgun (1), sürücünün yorgunluk belirtisi göstermediği durum: yorgun değil (0) olarak etiketlenir. Sürücü yorgunluğu yaklaşık %70 doğruluk oranı ile tespit edilmiştir.

4.2.2. Senaryo 2: SVM algoritması ile yorgunluk tahmin sistemi

Senaryo 2’de, SVM tabanlı sürücü yorgunluğu tahmin sistemi geliştirilmiştir. Sürüş simülasyon deneyleri eğitim aşamasında, yaşları 18-30 arasında değişen 10 gönüllü (5 erkek ve 5 kadın) ile gerçekleştirilmiştir. Deneyde, saniyede 320×240 boyutunda 15 kare yakalanmaktadır. Yüz görüntülerinden göz bölgesi ve ağız bölgesi dlib kütüphanesi ile tespit edilmiştir. Gözün konum bilgileri Şekil 4.8’de gösterilmiştir. Eşitlik (4.3) ile göz açıklık oranı (Cech ve diğ., 2016) hesaplanmaktadır.



Şekil 4.8. Göz konum bilgileri, a) göz, b) sağ ve sol göz

$$GAO = \frac{\| p_{tl} - p_{bl} \| + \| p_{tr} - p_{br} \|}{2 \| p_l - p_r \|} \quad (4.3)$$

$\| \ \|$, Öklit boyuna karşılık gelmektedir. Senaryo 2’de, p_{37}, \dots, p_{42} arası belirlenen noktalar ile dlib kütüphanesi ile bulunan sağ göz konum noktaları ve p_{43}, \dots, p_{48} arası belirlenen noktalar ile dlib kütüphanesi ile bulunan sol göz konum noktaları temsil edilir. Eşitlik (4.3) sağ göz için GAO_{sg} ve sol göz için GAO_{sl} ayrı ayrı hesaplanmıştır. Eşitlik (4.4)’te genel göz açıklık oranı hesaplanmıştır.

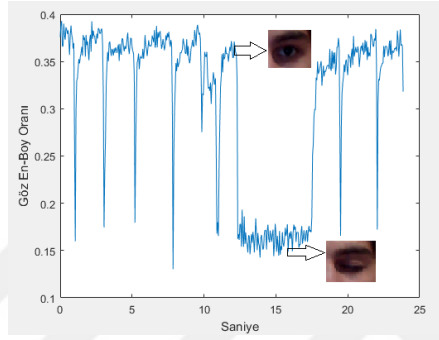
$$GGAO = \frac{GAO_{sg} + GAO_{sl}}{2} \quad (4.4)$$

Genel göz açıklık oranı için, gözün en-boy oranının (aspect ratio) ortalaması eşik değer kabul edilir. Bu genel göz açıklık oranı ($GGAO$), göz kırıldığında 0.24’ün altına

düşmektedir. Bundan dolayı göz kırpma işlemi için eşik değer 0.24 olarak belirlenmiştir. Eşitlik (4.5)'te göz açık ya da kapalı olma durumu hesaplanmıştır.

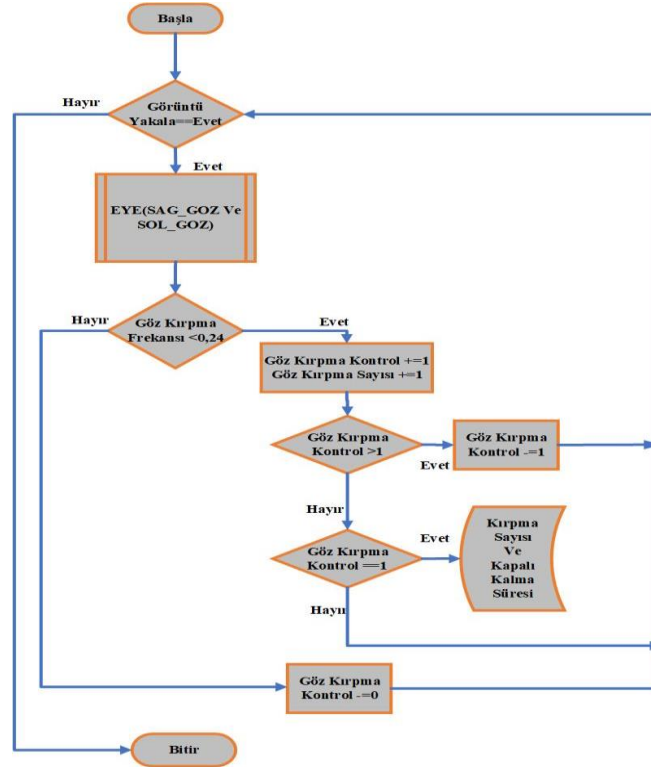
$$goz_etiketi = \begin{cases} 0 & GGAO < 0.24 \\ 1 & aksi\ taktirde \end{cases} \quad (4.5)$$

Eğitim aşamasında göz açıksa "1", göz kapalı ise "0" olarak etiketlenmiştir. Şekil 4.9'da GGAO'nun zamana göre değişimi verilmiştir.



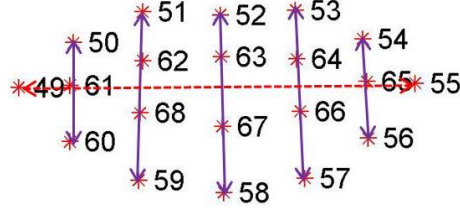
Şekil 4.9. GGAO'nun zamana göre değişimi

Göz kırpma hareketinin akış diyagramı Şekil 4.10'da gösterilmiştir.



Şekil 4.10. Göz kırpma hareketinin akış diyagramı

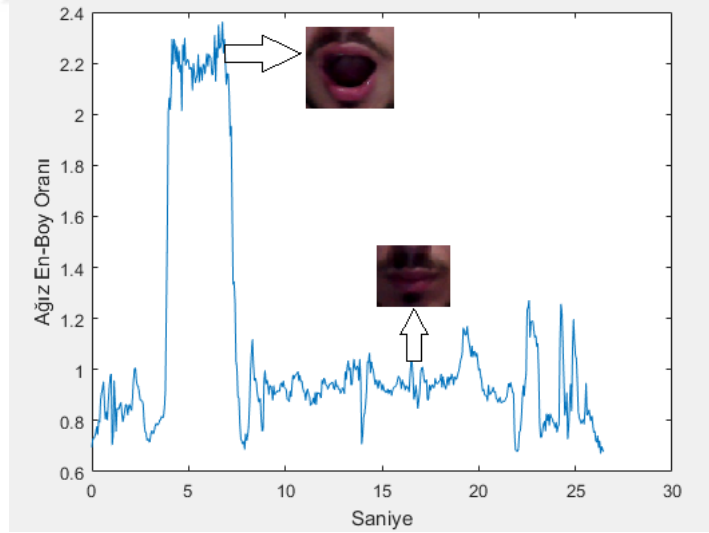
Yüz görüntülerinden ağız bölgesi dlib kütüphanesi ile belirlenmiştir. Ağızın konum bilgileri Şekil 4.11’de gösterilmiştir ve Eşitlik (4.6) ile ağız açıklık oranı (AAO) hesaplanmıştır.



Şekil 4.11. Ağız konum bilgileri

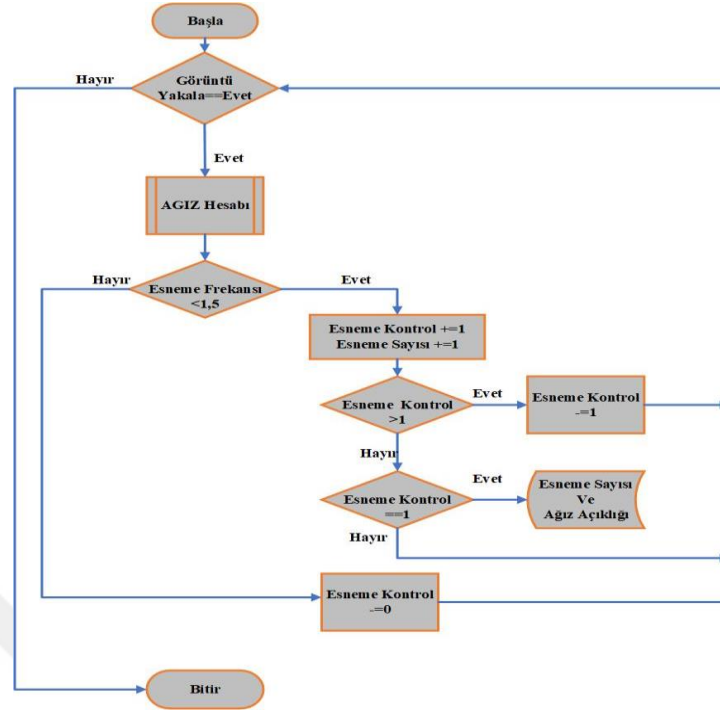
$$AAO = \frac{\|p_{50} - p_{60}\| + \|p_{51} - p_{59}\| + \|p_{52} - p_{58}\| + \|p_{53} - p_{57}\| + \|p_{54} - p_{56}\|}{2 \|p_{49} - p_{55}\|} \quad (4.6)$$

Senaryo 2’de, p_{49}, \dots, p_{68} arası belirlenen noktalar ile dlib kütüphanesi ile bulunan ağız konum noktaları temsil edilir. Esneme tespiti için, ağızın en-boy oranının (aspect ratio) ortalaması eşik değer kabul edilir. Bu ortalama değer, sürücü esnemeye başladığında, 1,5’un üstüne çıkmaktadır. Şekil 4.12’de AAO’nun zamana göre değişimi verilmiştir.



Şekil 4.12. AAO’nun zamana göre değişimi

Eğitim aşamasında ağız görüntüsü üzerinde, açık ağız durumu “1”, kapalı ağız durumu için “0” olarak etiketlenmiştir. Şekil 4.13’te esneme tespitinin akış diyagramı verilmektedir.



Şekil 4.13. Esneme tespiti akış diyagramı

Etiketlenen veriler SVM algoritması ile sınıflandırılmıştır. Sınıflandırma aşamasında k- kat 10 olarak seçilmiştir. Çalışmada, toplamda 713 veri kullanılmıştır. Çalışmadaki veriler yaklaşık %80 eğitim veri kümesi (546 veri kümesi) ve yaklaşık %20 test veri kümesi (167 veri kümesi) olarak iki gruba ayrılmıştır. Veriler, sürücünün uykuya dalma durumu: çok yorgun (1), sürücünün yorgunluk belirtisi göstermediği durum: yorgun değil (0) olarak etiketlenmiş ve sınıflandırma işlemi gerçekleştirilmiştir. Tablo 4.2’de eğitim veri kümesi örneği gösterilmiştir. Tablo 4.3’te test veri kümesi örneği gösterilmiştir.

Tablo 4.2. Eğitim veri kümesi örneği

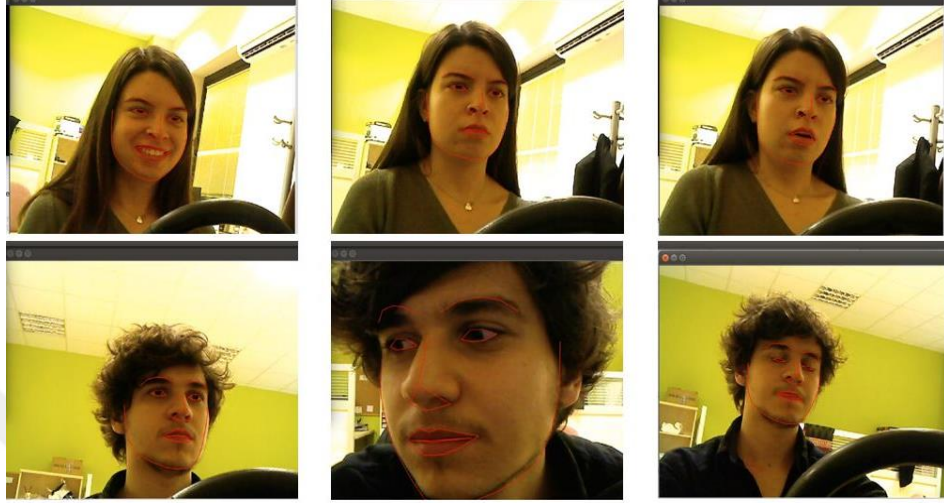
PERCLOS	Esneme sayısı	Kırpma sayısı	Etiket değerleri
0,0331	1	3	1
0,0262	0	2	1
0,0339	2	1	0
0,0705	0	3	1
0,0398	0	3	1

Tablo 4.3. Test veri kümesi örneği

PERCLOS	Esneme sayısı	Kırpma sayısı	Etiket değerleri
0,0230	0	2	1
0,0262	2	2	0
0,0339	0	1	0
0,0705	0	3	1
0,0398	2	3	0

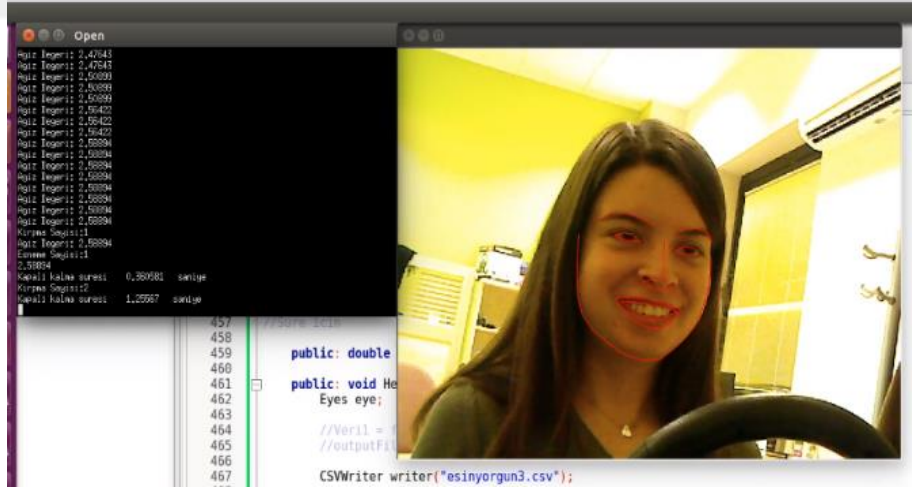
Sistemde her 15 sn’de bir sürücünün yorgun olup olmadığı bilgisi güncellenir. Senaryo 2, %97,8 doğruluk oranı ile çalışmıştır.

Şekil 4.14’te Gerçek zamanlı sürüş simülasyon test örnek görüntüleri gösterilmiştir.



Şekil 4.14. Gerçek zamanlı sürüş simülasyon test örnek görüntüleri

Şekil 4.15’te Yorgunluk tespiti test aşaması örnek gösterimi gösterilmiştir.



Şekil 4.15. Yorgunluk tespiti test aşaması örnek gösterimi

4.2.3. Senaryo 3: Tek Görevli ConNN modeli ile yorgunluk tahmin sistemi

Senaryo 3’te veri kümeleri, eğitim kümesine ve test kümesine ayrılmıştır. Bu senaryoda ZJU, YawDD, Nthu-DDD ve KouBM-DFD veri kümelerinden yararlanılmıştır ve veri kümeleri dağılımı Tablo 4.4’te gösterilmiştir.

Tablo 4.4. Veri kümeleri dağılımı

Veri Kümesi	ZJU	YawDD	Nthu-DDD	KouBM-DFD
Eğitim	12128	51600	28943	26694
Test	3000	11710	4964	5730
Toplam	15128	62910	33907	32424

Matematiksel model tabanlı yaklaşımlarda değiştirilebilir model parametrelerinin (ağırlık, katman, çekirdek/kernel boyutları, çekirdek parametreleri, vb.) seçimi problem çözümünde en önemli adımlardan biridir. Herhangi bir problem için söz konusu model parametreleri için en uygun (optimum) parametrelerin bulunması oldukça zordur. Bu nedenle farklı model parametreleri birçok kez denenerek en uygun parametre değerleri belirlenmeye çalışılır. Bu sebeple uyarlanırlar parametre değiştirme yaklaşımları geliştirmek faydalı olacaktır. Bu senaryoda konvolüsyon katmanlarında bulunan filtre sayısı, adımlama sayısı ve dolgu sayısı hiper parametreleri değiştirilerek optimum parametre değerlerinin bulunması için testler yapılmıştır.

Tek Görevli ConNN modelinde optimum parametre değerlerini elde edebilmek amaçlı, hiper parametre (konvolüsyon adım sayısı) değiştirilerek toplamda farklı 2 model (Model-1 ve Model 2) oluşturulmuştur. Oluşturulan Model-1 üzerinde ise konvolüsyon katmanlarında bulunan filtre sayıları değiştirilerek farklı 3 alt model (Model-1(a), Model-1(b) ve Model-1(c)) tasarlanmıştır. Tasarlanan modellerin ağ yapısı Tablo 4.5'te verilmiş olup değiştirilen hiper parametreler koyu renkli belirtilmiştir.

Tablo 4.5. Tek Görevli ConNN modellerinin ağ yapısı

Model	Katman adı	Katman Tipi	Parametreler
Model-1 (a)	Conv1	Konvolüsyon	5×5, 3 , adım sayısı =1
	A1	Aktivasyon	ReLU
	Pool1	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	Conv2	Konvolüsyon	5×5, 4 , adım sayısı =1
	A2	Aktivasyon	ReLU
	Pool2	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	Conv3	Konvolüsyon	4×4, 5 , adım sayısı =1
	A3	Aktivasyon	ReLU
	Pool3	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
FullyCon1	Tam bağlantı	128, 2d	
Model-1 (b)	Conv1	Konvolüsyon	5×5, 6 , adım sayısı =1
	A1	Aktivasyon	ReLU
	Pool1	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	Conv2	Konvolüsyon	5×5, 8 , adım sayısı =1

Tablo 4.5.(Devam) Tek Görevli ConNN modellerinin ağ yapısı

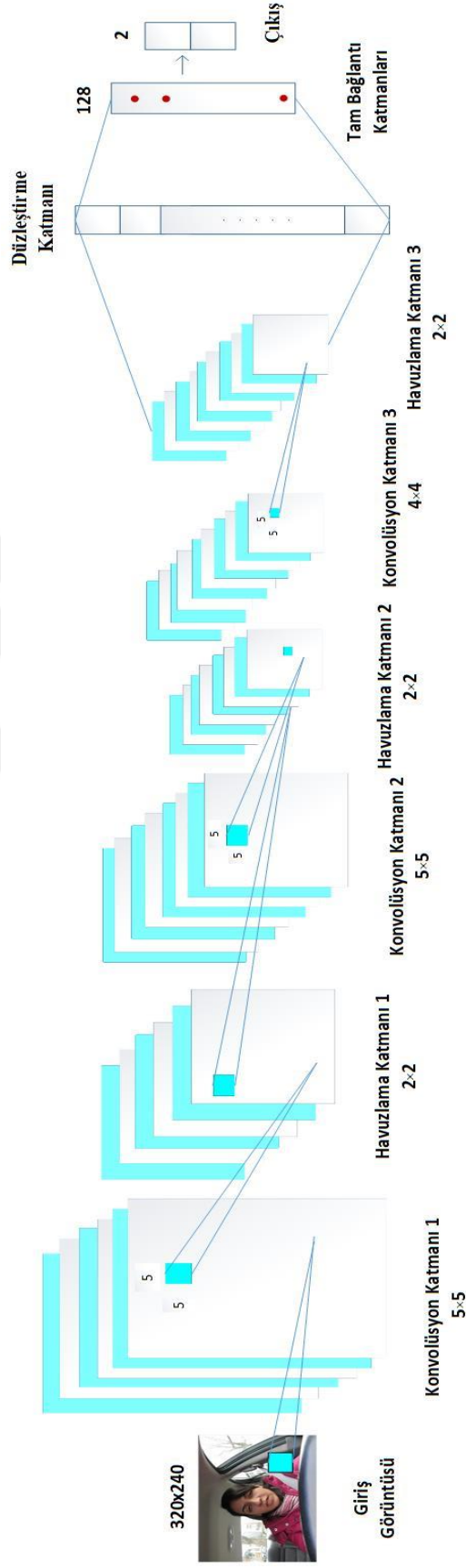
Model	Katman adı	Katman Tipi	Parametreler
Model-1 (b)	A2	Aktivasyon	ReLU
	Pool2	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	Conv3	Konvolüsyon	4×4, 10 , adım sayısı = 1
	A3	Aktivasyon	ReLU
	Pool3	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	FullyCon1	Tam bağlantı	128, 2d
Model-1 (c)	Conv1	Konvolüsyon	5×5, 12 , adım sayısı = 1
	A1	Aktivasyon	ReLU
	Pool1	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	Conv2	Konvolüsyon	5×5, 16 , adım sayısı = 1
	A2	Aktivasyon	ReLU
	Pool2	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	Conv3	Konvolüsyon	4×4, 20 , adım sayısı = 1
	A3	Aktivasyon	ReLU
	Pool3	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
FullyCon1	Tam bağlantı	128, 2d	
Model-2	Conv1	Konvolüsyon	5×5, 6 , adım sayısı = 2
	A1	Aktivasyon	ReLU
	Pool1	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	Conv2	Konvolüsyon	5×5, 8 , adım sayısı = 2
	A2	Aktivasyon	ReLU
	Pool2	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	Conv3	Konvolüsyon	4×4, 10 , adım sayısı = 2
	A3	Aktivasyon	ReLU
	Pool3	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
FullyCon1	Tam bağlantı	128, 2d	

Eğitim aşamasında Rastgele Bayır İniş (SGD) optimizasyon algoritmasından yararlanılmaktadır. SGD parametreleri için başlangıçta parti boyutu 64 ve momentum değeri 0.6 olarak belirlenmiştir. Adım sayısı başlangıçta 0.01 olarak seçilmiştir. Çalışmada kullanılan doğruluk hesabı Eşitlik (4.7) ve Eşitlik (4.8)'de verilmiştir.

$$Doğruluk = \frac{1}{N} \sum_i^N 1(y_i == \hat{y}) \quad (4.7)$$

$$Doğruluk = (TP+TN) / (TP+TN+FP+FN) \quad (4.8)$$

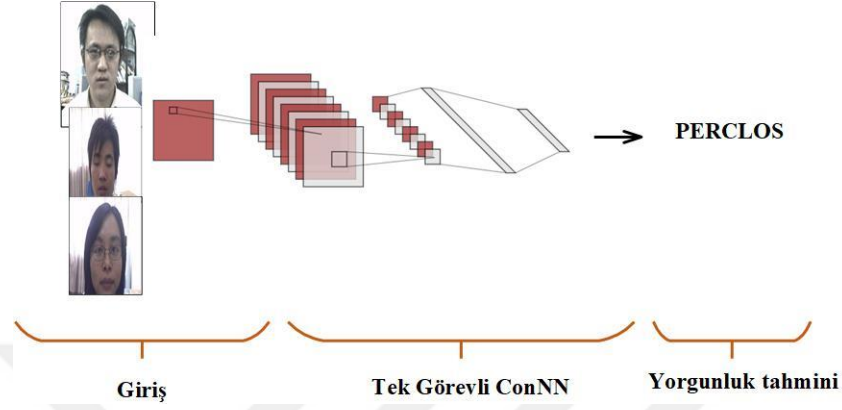
Burada y , hedef değerlerinin (herhangi bir boyutta olan bir dizi içinde tutulan veriler) ve \hat{y} tahmin değerlerinin vektörüne karşılık gelmekte olup, doğruluk değeri ikili sınıflandırma karmaşıklık matrisindeki toplam gerçek pozitif ve gerçek negatif değerlerine göre hesaplanmaktadır. N değeri ise toplam veri sayısına karşılık gelmektedir. Tezde kullanılan Tek Görevli ConNN modeli Şekil 4.16'da gösterilmiştir.



Şekil 4.16. Tek Görevli ConNN modeli

4.2.3.1. Göz durum bilgisine dayanan yorgunluk tahmin sistemi

Göz durum bilgisine dayanan yorgunluk tahmin sisteminde Tek Görevli ConNN modeli kullanılmıştır. Oluşturulan sistemin genel yapısı Şekil 4.17’de gösterilmiştir.



Şekil 4.17. Göz durum bilgisine dayanan sistemin genel yapısı

Göz durum bilgisine dayanan yorgunluk tahmin sisteminde kullanılan Tek Görevli ConNN ağ modelinin eğitimi ve testi; ZJU, YawDD, Nthu-DDD ve KouBM-DFD veri kümeleri üzerinde yapılmıştır. Göz durum bilgisine dayanan Tek Görevli ConNN modeli kullanılarak elde edilen doğruluk oranlarının karşılaştırılması Tablo 4.6’da gösterilmiştir.

Tablo 4.6. Göz durum bilgisine dayanan Tek Görevli ConNN modeli kullanılarak elde edilen doğruluk oranları

Model	Tek Görevli ConNN	Döngü sayısı	ZJU	YawDD	Nthu-DDD	KouBM-DFD
Model-1 (a)	Eğitim doğruluğu	100	%97,96	%99,40	%98,46	%99,35
		200	%98,62	%99,68	%98,52	%99,64
		300	%98,65	%99,76	%98,72	%99,76
		400	%99,93	%99,84	%98,66	%99,82
		500	%99,78	%99,82	%99,05	%99,84
	Test doğruluğu	100	%97,72	%97,38	%97,25	%98,04
		200	%97,28	%97,27	%97,35	%98,22
		300	%98,17	%97,18	%98,64	%98,48
		400	%98,97	%97,67	%98,45	%98,76
		500	%98,75	%97,73	%98,55	%98,81
Model-1(b)	Eğitim doğruluğu	100	%98,83	%99,40	%99,44	%99,17
		200	%98,40	%99,67	%99,56	%99,60
		300	%99,60	%99,75	%99,61	%99,70
		400	%99,50	%99,70	%99,67	%99,84
		500	%99,74	%97,70	%99,73	%99,92
	Test doğruluğu	100	%96,72	%97,80	%98,67	%97,95
		200	%96,97	%97,85	%98,53	%98,27
		300	%96,61	%97,80	%98,61	%97,27
		400	%95,46	%97,55	%98,54	%98,00
		500	%95,98	%97,82	%98,68	%97,52

Tablo 4.6. (Devam) Göz durum bilgisine dayanan Tek Görevli ConNN modeli kullanılarak elde edilen doğruluk oranları

Model	Tek Görevli ConNN	Döngü sayısı	ZJU	YawDD	Nthu-DDD	KouBM-DFD
Model-1(c)	Eğitim doğruluğu	100	%98,69	%99,58	%99,46	%99,23
		200	%99,55	%99,73	%99,61	%99,61
		300	%99,63	%99,86	%99,60	%99,72
		400	%99,90	%99,79	%99,66	%99,83
		500	%99,80	%99,80	%99,70	%99,82
	Test doğruluğu	100	%96,72	%97,71	%98,42	%98,60
		200	%97,17	%97,84	%98,35	%98,78
		300	%97,17	%98,02	%98,33	%98,70
		400	%96,97	%97,80	%98,59	%98,92
		500	%97,22	%97,70	%98,36	%98,92
Model-2	Eğitim doğruluğu	100	%88,13	%96,51	%98,50	%98,23
		200	%97,12	%97,18	%98,85	%99,07
		300	%98,33	%97,57	%98,98	%99,31
		400	%98,62	%98,60	%99,01	%99,43
		500	%98,20	%98,76	%98,18	%99,56
	Test doğruluğu	100	%85,93	%94,49	%97,65	%97,33
		200	%92,93	%95,34	%97,56	%97,90
		300	%94,67	%95,60	%97,89	%97,99
		400	%95,87	%95,44	%96,87	%97,32
		500	%95,22	%95,48	%96,82	%97,86

Tablo 4.6 incelendiğinde 500 döngü sayısı sonunda, test doğruluğu için en iyi doğruluk oranları Model-1(a)'da elde edilmiştir. Bu değerler; ZJU veri kümesi üzerinde %98,75, YawDD veri kümesi üzerinde %97,73, Nthu-DDD veri kümesi üzerinde %98,55, KouBM-DFD veri kümesi üzerinde %98,81 olarak hesaplanmıştır. Göz durum bilgisine dayanan Tek Görevli ConNN modeli kullanılarak elde edilen doğruluk oranlarının literatürde bulunan diğer yaklaşımlarla karşılaştırılması Tablo 4.7'de gösterilmiştir. Elde edilen sonuçların Gu ve diğ., 2018 hariç diğer yaklaşımlardan daha başarılı olduğu tespit edilmiştir.

Tablo 4.7. Doğruluk oranlarının literatürde bulunan diğer yaklaşımlarla karşılaştırılması

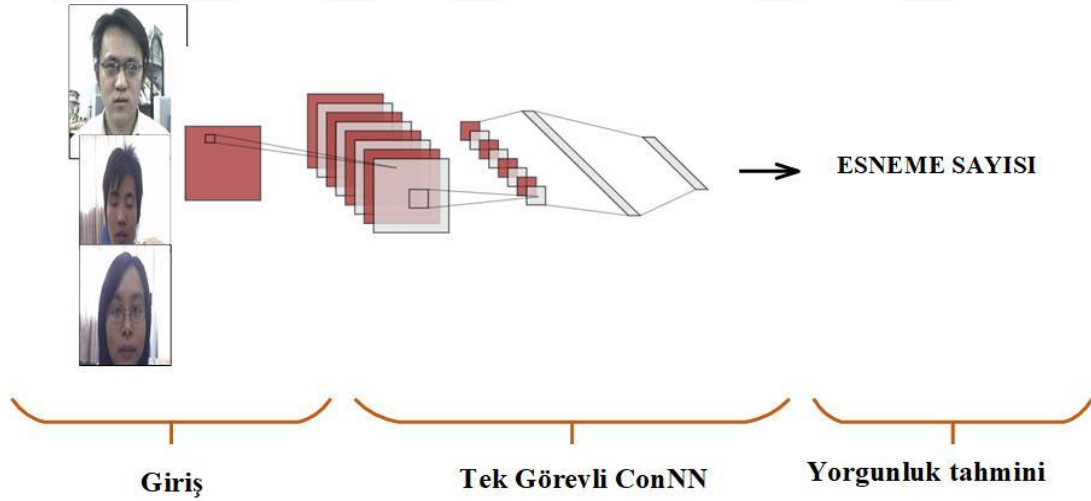
Literatür	Yıl	Yöntem	Veri kümesi	Doğruluk oranı
Luo ve diğ., 2013	2013	Adaboost	Kendi oluşturdukları veri kümesi	%92
Punitha ve diğ., 2014	2014	SVM	Kendi oluşturdukları veri kümesi	%93,5
Yan ve diğ., 2015	2015	CNN	Kendi oluşturdukları veri kümesi	%95,56
Zhang ve diğ., 2017	2017	CNN	IRF,ZJU	%98,83
Li ve diğ., 2018	2018	Kalman Filtre, Ağırlıklı Ortalama Öteleme	Kendi oluşturdukları veri kümesi	%93,19
Huang ve diğ., 2018	2018	FDCN	ZJU,CEW	%94,9

Tablo 4.7. (Devam) Doğruluk oranlarının literatürde bulunan diğer yaklaşımlarla karşılaştırılması

Literatür	Yıl	Yöntem	Veri kümesi	Doğruluk oranı	
Ma, 2018	2018	SVM	Kendi oluşturdukları veri kümesi	% 91	
Gu ve diğ., 2018	2018	MTCNN	EMD,ZJU,CEW	%98,85	
Sharan ve diğ., 2019	2019	CNN	Kendi oluşturdukları veri kümesi	%96,56	
Liu ve diğ., 2019	2019	CNN	DFII, CEW	%96,2	
Zhang ve diğ.,2019	2019	Facial Key Points	Kendi oluşturdukları veri kümesi	%92	
Savaş ve diğ., 2018	2019	SVM	YawDD ve kendi oluşturdukları veri kümesi	%97,93	
Önerilen model	2020	ConNN (Model-1(a))	ZJU, YawDD	%98,25	%97,73
			Nthu-DDD , KouBM-DFD	%98,55	%98,81

4.2.3.2. Ağız durum bilgisine dayanan yorgunluk tahmin sistemi

Ağız durum bilgisine dayanan yorgunluk tahmin sisteminde kullanılan Tek Görevli ConNN eğitimi ve testi; YawDD, Nthu-DDD ve KouBM-DFD veri kümeleri üzerinde yapılmıştır. Oluşturulan sisteminin genel yapısı Şekil 4.18’de gösterilmiştir. Ağız durum bilgisine dayanan Tek Görevli ConNN sınıflandırması ile elde edilen doğruluk oranlarının karşılaştırılması Tablo 4.8’de gösterilmiştir.



Şekil 4.18. Ağız durum bilgisine dayanan sistemin genel yapısı

Tablo 4.8 incelendiğinde 500 döngü sayısı sonunda, test doğruluğu için en iyi doğruluk oranları Model-1(a)’da elde edilmiştir. Bu değerler; YawDD veri kümesi üzerinde %99,35, Nthu-DDD veri kümesi üzerinde %99,25, KouBM-DFD veri kümesi üzerinde %99,01 olarak hesaplanmıştır.

Tablo 4.8. Ağız durum bilgisine dayanan Tek Görevli ConNN modeli ile elde edilen doğruluk oranları

Model	Tek Görevli ConNN	Döngü sayısı	YawDD	Nthu-DDD	KouBM-DFD
Model-1(a)	Eğitim doğruluğu	100	%98,76	%98,86	%99,31
		200	%98,82	%98,90	%99,68
		300	%99,27	%99,09	%99,73
		400	%99,36	%99,25	%99,80
		500	%99,70	%99,30	%99,79
	Test doğruluğu	100	%98,43	%98,71	%97,42
		200	%98,59	%98,86	%98,20
		300	%99,20	%98,78	%98,89
		400	%99,24	%99,05	%98,90
		500	%99,35	%99,25	%99,01
Model-1(b)	Eğitim doğruluğu	100	%98,82	%98,83	%97,26
		200	%99	%98,85	%97,68
		300	%98,92	%98,89	%97,84
		400	%98,93	%99,01	%97,82
		500	%98,90	%98,87	%97,87
	Test doğruluğu	100	%98,47	%98,88	%96,73
		200	%98,58	%98,83	%96,99
		300	%98,60	%98,69	%96,82
		400	%98,65	%98,89	%96,92
		500	%98,57	%98,83	%97,22
Model-1(c)	Eğitim doğruluğu	100	%99,80	%99,83	%99,46
		200	%99,88	%99,87	%99,75
		300	%99,89	%99,91	%99,82
		400	%99,93	%99,90	%99,90
		500	%99,91	%99,32	%99,86
	Test doğruluğu	100	%98,36	%99,78	%98,50
		200	%99,35	%99,74	%98,85
		300	%98,71	%99,21	%98,83
		400	%99,70	%99,23	%98,24
		500	%98,21	%99,28	%98,32
Model-2	Eğitim doğruluğu	100	%97,64	%97,97	%96,85
		200	%97,71	%97,91	%97,25
		300	%97,77	%97,88	%97,39
		400	%97,81	%97,86	%97,40
		500	%97,80	%97,94	%97,73
	Test doğruluğu	100	%97,50	%97,83	%95,81
		200	%97,45	%97,74	%96,71
		300	%97,57	%97,81	%96,59
		400	%97,64	%97,83	%95,90
		500	%97,56	%97,83	%96,57

Ağız durum bilgisine dayanan Tek Görevli ConNN modeli ile elde edilen doğruluk oranının literatürde bulunan diğer yaklaşımlarla karşılaştırılması Tablo 4.9'da gösterilmiştir. Elde edilen sonuçların kullanılan veri kümeleri üzerinde diğer yaklaşımlardan daha başarılı ve güçlü olduğu tespit edilmiştir.

Tablo 4.9. Doğruluk oranlarının literatürde bulunan diğer yaklaşımlarla karşılaştırılması

Literatür	Yıl	Yöntem	Veri kümesi	Doğruluk oranı		
Luo ve diğ., 2013	2013	LBP	Kendi oluşturdukları veri kümesi	%70		
Zhang ve diğ., 2015	2015	CNN	Kendi oluşturdukları veri kümesi, YawDD	%89		
Yan ve diğ., 2015	2015	CNN	Kendi oluşturdukları veri kümesi	%98,22		
Tang ve diğ., 2016	2016	MB-LBP, Adaboost	Kendi oluşturdukları veri kümesi	%98		
Akrouf ve diğ., 2016	2016	Viola ve Jones	YawDD, MiraclHB	%85		
Zhang ve diğ., 2017	2017	CNN	ImageNet, State Farm Distracted Driver Detection, YawDD	%87		
Gupta ve diğ., 2017	2017	TBA, SVM	Kendi oluşturdukları veri kümesi	-		
Wathiq ve diğ., 2017	2017	SVM	Kendi oluşturdukları veri kümesi	%94,32		
Raman ve diğ., 2018	2018	Haar sınıflandırıcı	YawDD	%75		
Gu ve diğ., 2018	2018	MTCNN	EMD, ZJU, CEW	%98,05		
Houssaini ve diğ., 2019	2019	SVM	IBUG 300-W	%83		
Ali ve diğ., 2019	2019	Haar sınıflandırıcı	Kendi oluşturdukları veri kümesi	-		
Yazdi ve diğ., 2019	2019	ROI	VAP_RGBD	%95		
Önerilen model	2019	ConNN (Model-1(a))	YawDD, Nthu-DDD, KouBM-DFD	%99,35	%99,25	%99,01

4.3.4. Senaryo 4: Çok Görevli ConNN ile yorgunluk tahmin sistemi

Senaryo 3’deki gibi optimum çözümlerin elde edilmesi için, Senaryo 4 altında verilen ağ modellerinde (İki çıkışlı çok görevli ConNN ve Üç çıkışlı çok görevli ConNN) kullanılan hiper parametreler (konvolüsyon katmanı adım sayısı ve havuzlama katmanı adım sayısı) değiştirilerek farklı 4 model (Model-1, Model-2, Model-3 ve Model-4) oluşturulmuştur. Model-1 üzerinde ise konvolüsyon katmanlarında bulunan filtre sayıları değiştirilerek (Model-1(a), Model-1(b) ve Model-1(c)) farklı 3 test yapılmıştır. Modellerdeki doğruluk hesabı Eşitlik (4.7)’de verilmiştir.

Senaryo 4 altında verilen ağ modelleri için Rastgele Bayır İniş optimizasyon algoritmasından yararlanılmıştır. Başlangıçta parti boyutu = 64 ve momentum değeri 0.6 olarak belirlenmiştir. Adım sayısı başlangıçta 0.01 olarak seçilmiştir. Tez kapsamında göz durum bilgileri değerlendirilirken gözlerin açık olduğu durum ‘1’

kapalı olduğu durum ‘0’ olarak etiketlenmiştir. Ağız durum bilgileri değerlendirilirken ağzın açık olduğu durum ‘1’, kapalı iken ‘0’ olarak etiketlenmiştir. Kafa bilgileri değerlendirilirken kafanın düz olduğu durum ‘1’, kafanın aşağı düştüğü durum ‘0’ olarak etiketlenmiştir.

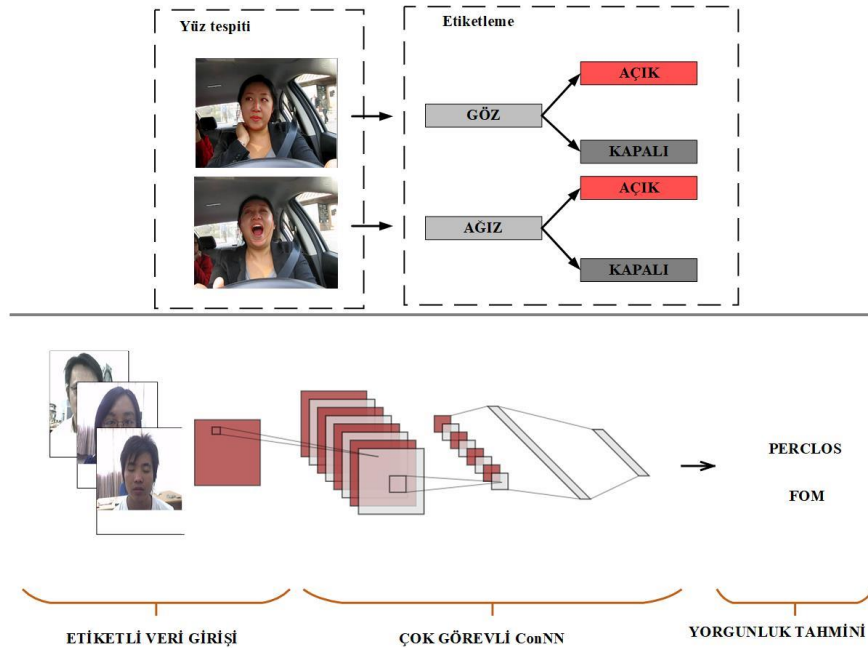
4.3.4.1. İki çıkışlı çok görevli ConNN ile yorgunluk tahmin sistemi

Eğitim aşamasında, veri kümeleri eğitim ve test kümelerine ayrılmıştır. İki Çıkışlı Çok Görevli modelde kullanılan YawDD, Nthu-DDD ve KouBM-DFD veri kümelerinin dağılımı Tablo 4.10’da gösterilmiştir.

Tablo 4.10. Veri kümeleri dağılımı

Veri kümesi	YawDD	Nthu-DDD	KouBM-DFD
Eğitim	51600	28943	26694
Test	11710	4964	5730
Toplam	62910	33907	32424

Eğitim aşamasındaki görüntüler üzerinde göz ve ağız durum bilgileri aynı anda sınıflandırıldığı için YawDD, Nthu-DDD ve KouBM-DFD veri kümelerinin içerisinde bulunan güneş gözlüklü sürücü verileri kullanılamamıştır. Tasarlanan modellerin ağ yapıları Tablo 4.11’de verilmiş olup değiştirilen hiper parametreler koyu renkli belirtilmiştir. Göz ve ağız durum bilgilerine dayanan önerilen sistemin genel yapısı Şekil 4.19’da gösterilmiştir.



Şekil 4.19. Göz ve ağız durum bilgilerine dayanan sistemin genel yapısı

Tablo 4.11. Modellerin ađ yapıları

Model	Katman adı	Katman Tipi	Parametreler
Model-1(a)	Conv1	Konvolüsyon	5×5, 3 , adım sayısı =1
	Pool1	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A1	Aktivasyon	ReLU
	Conv2	Konvolüsyon	5×5, 4 , adım sayısı =1
	Pool2	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A2	Aktivasyon	ReLU
	Conv3	Konvolüsyon	4×4, 5 , adım sayısı =1
	Pool3	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A3	Aktivasyon	ReLU
FullyCon1	Tam bağlantı	(128, 2d), (128, 2d)	
Model-1(b)	Conv1	Konvolüsyon	5×5, 6 , adım sayısı =1
	Pool1	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A1	Aktivasyon	ReLU
	Conv2	Konvolüsyon	5×5, 8 , adım sayısı =1
	Pool2	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A2	Aktivasyon	ReLU
	Conv3	Konvolüsyon	4×4, 10 , adım sayısı =1
	Pool3	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A3	Aktivasyon	ReLU
FullyCon1	Tam bağlantı	(128, 2d), (128, 2d)	
Model-1(c)	Conv1	Konvolüsyon	5×5, 12 , adım sayısı =1
	Pool1	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A1	Aktivasyon	ReLU
	Conv2	Konvolüsyon	5×5, 16 , adım sayısı =1
	Pool2	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A2	Aktivasyon	ReLU
	Conv3	Konvolüsyon	4×4, 20 , adım sayısı =1
	Pool3	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A3	Aktivasyon	ReLU
FullyCon1	Tam bağlantı	(128, 2d), (128, 2d)	
Model-2	Conv1	Konvolüsyon	5×5, 6 , adım sayısı =2
	Pool1	Havuzlama	2×2, maksimum havuzlama, adım sayısı=2
	A1	Aktivasyon	ReLU
	Conv2	Konvolüsyon	5×5, 8 , adım sayısı =2
	Pool2	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A2	Aktivasyon	ReLU
	Conv3	Konvolüsyon	4×4, 10 , adım sayısı =2
	Pool3	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A3	Aktivasyon	ReLU
FullyCon1	Tam bağlantı	(128, 2d), (128, 2d)	
Model-3	Conv1	Konvolüsyon	5×5, 6 , adım sayısı =1
	Pool1	Havuzlama	4×4, maksimum havuzlama, adım sayısı=3
	A1	Aktivasyon	ReLU
	Conv2	Konvolüsyon	5×5, 8 , adım sayısı =1
	Pool2	Havuzlama	4×4, maksimum havuzlama, adım sayısı =3
	A2	Aktivasyon	ReLU
	Conv3	Konvolüsyon	4×4, 10 , adım sayısı =1
	Pool3	Havuzlama	4×4, maksimum havuzlama, adım sayısı =3
	A3	Aktivasyon	ReLU
FullyCon1	Tam bağlantı	(128, 2d), (128, 2d)	

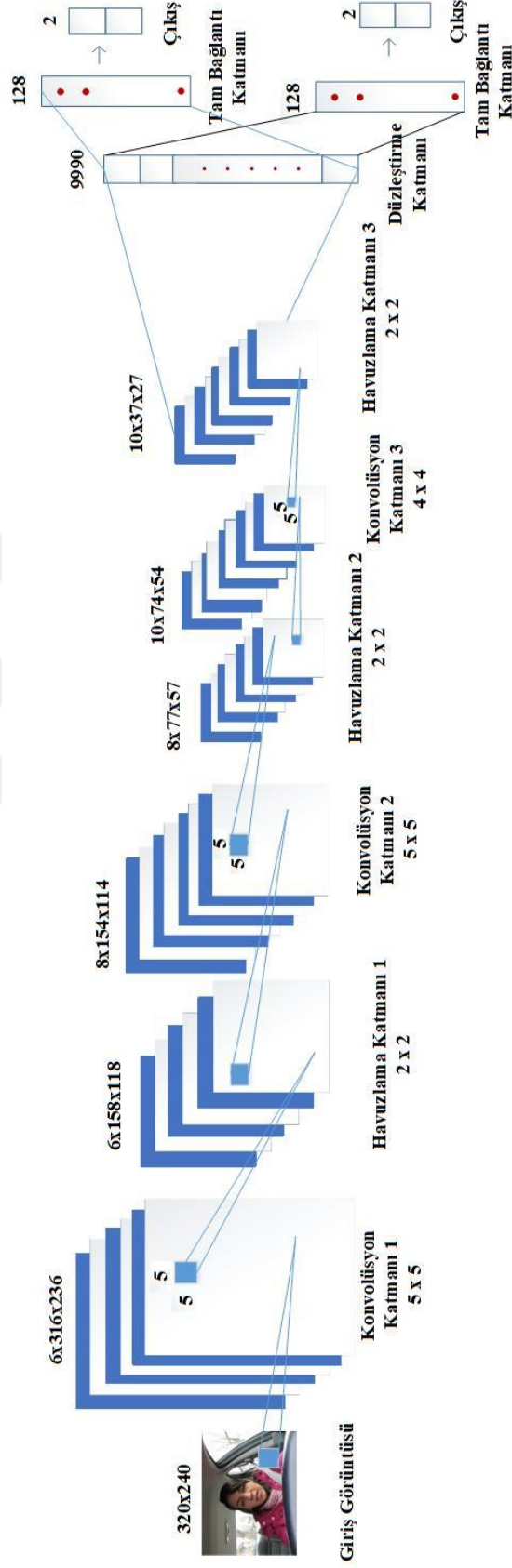
Tablo 4.11. (Devam) Modellerin ağ yapıları

Model	Katman adı	Katman Tipi	Parametreler
Model-4	Conv1	Konvolüsyon	5×5, 6 , adım sayısı=2
	Pool1	Havuzlama	4×4, maksimum havuzlama, adım sayısı=3
	A1	Aktivasyon	ReLU
	Conv2	Konvolüsyon	5×5, 8 , adım sayısı=2
	Pool2	Havuzlama	4×4, maksimum havuzlama, adım sayısı=3
	A2	Aktivasyon	ReLU
	Conv3	Konvolüsyon	4×4, 10 , adım sayısı=2
	Pool3	Havuzlama	4×4, maksimum havuzlama, adım sayısı=3
	A3	Aktivasyon	ReLU
FullyCon1	Tam bağlantı	(128, 2d), (128, 2d)	

İki Çıkışlı Çok Görevli ConNN modeli Şekil 4.20’de verilmiştir. Model-1 için hiper parametreler değiştirilerek Model-1(a), Model-1(b) ve Model-1(c) modelleri oluşturulmuştur. Bu modellerden elde edilen doğruluk oranlarının karşılaştırılması Tablo 4.12’de gösterilmiştir.

Tablo 4.12. Model-1 eğitiminden elde edilen doğruluk oranları

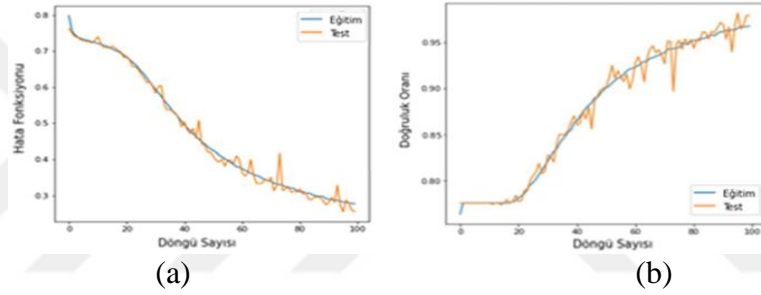
Model	İki çıkışlı Çok Görevli ConNN	Doğruluk Oranı			
		Döngü sayısı	YawDD	Nthu-DDD	KouBM-DFD
Model-1(a)	Eğitim doğruluğu	100	%95,37	%95,72	%96,05
		200	%96,49	%96,81	%96,90
		300	%97,76	%97,20	%97,72
		400	%97,82	%97,48	%97,90
		500	%97,73	%98,62	%98,72
	Test doğruluğu	100	%95,54	%94,74	%95,99
		200	%96,46	%95,01	%96,19
		300	%96,84	%96,72	%97,70
		400	%96,71	%96,35	%97,85
		500	%96,62	%97,70	%97,69
Model-1(b)	Eğitim doğruluğu	100	%98,72	%96,74	%96,82
		200	%99,64	%97,85	%97,50
		300	%99,82	%98,31	%98,36
		400	%99,88	%98,74	%98,71
		500	%99,90	%99,03	%98,94
	Test doğruluğu	100	%98,49	%96,56	%96,78
		200	%98,96	%97,86	%97,48
		300	%98,82	%98,33	%98,06
		400	%99,16	%98,61	%98,55
		500	%99,18	%98,93	%98,88
Model-1(c)	Eğitim doğruluğu	100	%96,52	%95,96	%96,46
		200	%97,21	%96,92	%96,52
		300	%97,58	%96,78	%96,88
		400	%97,86	%96,96	%97,88
		500	%97,80	%97,24	%97,58
	Test doğruluğu	100	%95,45	%95,83	%96,83
		200	%96,48	%95,99	%96,64
		300	%96,27	%96,25	%96,54
		400	%96,98	%96,83	%95,90
		500	%96,86	%96,85	%97,03



Şekil 4.20. İki Çıkışlı Çok Görevli ConNN modeli

Model-1(a), Model-1(b) ve Model-1(c) incelendiğinde; 500 döngü sayısı sonunda, test doğruluğu için en iyi doğruluk oranları Model-1(b)'de elde edilmiştir. Bu nedenle tezde İki Çıkışlı Çok Görevli ConNN modeli için Model-1(b) ağı tercih edilmiştir. 500 döngü sayısı sonunda YawDD test doğruluğu %99,18, Nthu-DDD test doğruluğu %98,93, KouBM-DFD test doğruluğu %98,88 elde edilmiştir.

Göz ve ağız durum bilgisine dayanan Model-1(b)'nin eğitimi ve testi; YawDD, Nthu-DDD ve KouBM-DFD veri kümeleri üzerinde yapılmıştır. Göz ve ağız durum bilgisine dayanan Model-1(b) için (YawDD-100 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.21(a)'da ve göz ve ağız durum bilgisine dayanan Model-1(b) için eğitim ve test doğruluk grafiği Şekil 4.21(b)'de gösterilmiştir. Eğitim ve test hata değerleri Tablo 4.13'te verilmiştir.



Şekil 4.21. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği

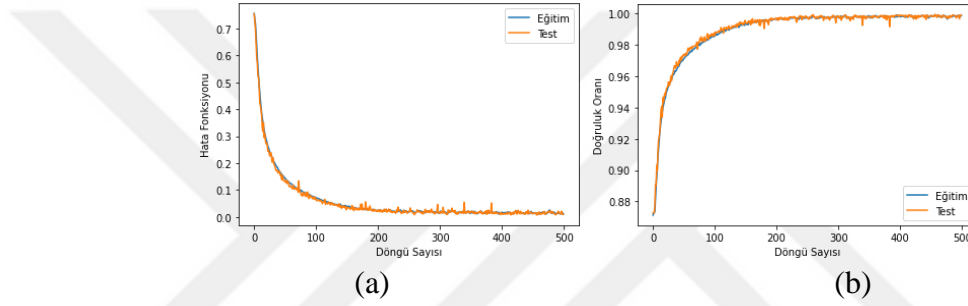
Tablo 4.13. Eğitim ve test hata değerleri (döngü sayısı=100)

Döngü sayısı	Eğitim Hatası	Test Hatası
1	0.7624	0.7983
20	0.6895	0.6819
40	0.4007	0.4048
60	0.3782	0.3993
80	0.3135	0,3094
100	0.1765	0.1556

Tablo 4.13'te verilen eğitim hatası ve test hatası değerleri Şekil 4.21'deki grafiklerde belirtilen 1, 20, 40, 80, 100. döngü sayılarındaki eğitim hatası ve test hatası değerleridir. Şekil 4.21(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı 100'e ulaştığında eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Yapılan ilk modellemede, model 100 döngü sayısına kadar gözlemlenmiş ancak hata fonksiyonuna bakılarak salınımın bir yörüngeye oturmadığı ve modelin öğrenmesinin bitmediği sonucu çıkarılmıştır.

Şekil 4.21(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı 100'e ulaştığında hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Yapılan ilk modellemede, model 100 döngü sayısına kadar gözlemlenmiş ancak doğruluk fonksiyonuna bakılarak bir yörüngeye oturmadığı ve modelin öğrenmesinin bitmediği sonucu çıkarılmıştır.

Göz ve ağız durum bilgisine dayanan Model-1(b) için (YawDD-500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.22(a)'da ve göz ve ağız durum bilgisine dayanan Model-1(b) için eğitim ve test doğruluk grafiği Şekil 4.22(b)'de gösterilmiştir.

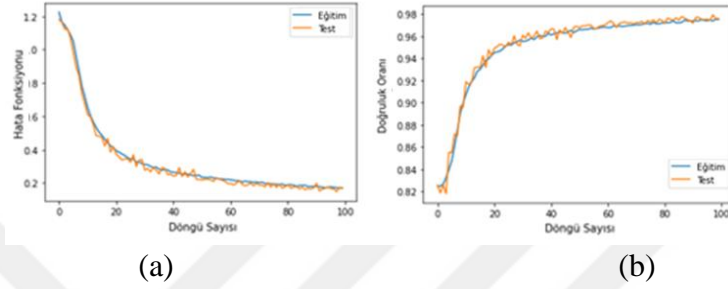


Şekil 4.22. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği

Şekil 4.22(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 100'e ulaşana kadar eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 200'e ulaştığında eğitim ve test hatasının azalmasının yavaşladığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Şekil 4.22(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 100'e ulaşana kadar hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 200'e ulaştığında hem test doğruluğunun hem de eğitim doğruluğunun yavaşlayarak arttığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar model gözlemlenmiş ve doğruluk fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Nthu-DDD veri kümesi üzerinde elde edilen sonuçlar Şekil 4.23 ve Şekil 4.24'te gösterilmiştir. Göz ve ağız durum bilgisine dayanan Model-1(b) için (Nthu-DDD-100 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) grafiği Şekil 4.23(a)'da ve göz ve ağız durum bilgisine dayanan Model-1(b) için eğitim ve test doğruluk grafiği Şekil 4.23(b)'de gösterilmiştir. Eğitim hatası ve test hatası değerleri Tablo 4.14'te verilmiştir.



Şekil 4.23. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği

Tablo 4.14. Eğitim ve test hata değerleri (döngü sayısı=100)

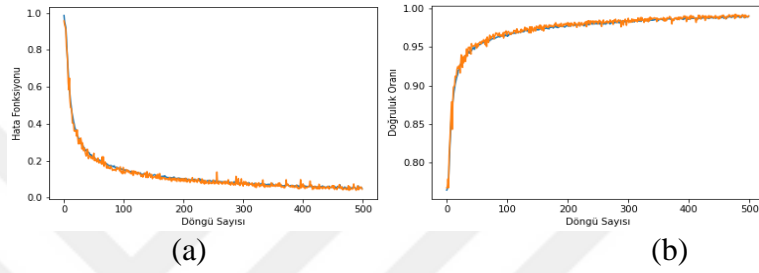
Döngü sayısı	Eğitim Hatası	Test Hatası
1	0.9565	1.0029
20	0.2010	0.2203
40	0.1542	0.1549
60	0.1194	0.1307
80	0.1172	0.1007
100	0.0993	0.0927

Tablo 4.14'te eğitim hatası ve test hatası değerleri Şekil 4.23'teki grafikte belirtilen 1, 20, 40, 80, 100. döngü sayılarındaki eğitim hatası ve test hatası değerleridir. Şekil 4.23(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 20'ye ulaştığında eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 40'a ulaştığında eğitim ve test hatasının azalmasının yavaşladığı gözlenmiştir. Yapılan ilk modellemede, model 100 döngü sayısına kadar gözlemlenmiş ancak hata fonksiyonuna bakılarak bir yörüngeye oturmadığı ve modelin öğrenmesinin bitmediği sonucu çıkarılmıştır.

Şekil 4.23(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 20'ye ulaştığında hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 40'a ulaştığında hem test doğruluğunun hem de eğitim doğruluğunun yavaşlayarak arttığı gözlenmiştir. Yapılan ilk

modellemede, model 100 döngü sayısına kadar gözlemlenmiş ancak hata fonksiyonuna bakılarak bir yörüngeye oturmadığı ve modelin öğrenmesinin bitmediği sonucu çıkarılmıştır.

Göz ve ağız durum bilgisine dayanan Model-1(b) için (Nthu-DDD-500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.24(a)'da ve göz ve ağız durum bilgisine dayanan Model-1(b) için eğitim ve test doğruluk grafiği Şekil 4.24(b)'de gösterilmiştir.



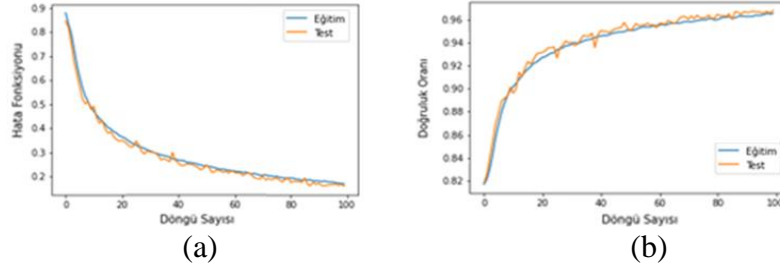
Şekil 4.24. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği

Şekil 4.24(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 30'a ulaştığında eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 60'a ulaştığında eğitim ve test hatasının azalmasının yavaşladığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Şekil 4.24(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 20'ye ulaştığında hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 40'a ulaştığında hem test doğruluğunun hem de eğitim doğruluğunun yavaşlayarak arttığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar model gözlemlenmiş ve doğruluk fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

KouBM-DFD veri kümesi üzerinde elde edilen sonuçlar Şekil 4.25 ve Şekil 4.26'da gösterilmiştir. Göz ve ağız durum bilgisine dayanan Model-1(b) için (KouBM-DFD-100 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function)

grafığı Şekil 4.25(a)'da ve göz ve ağız durum bilgisine dayanan Model-1(b) için eğitim ve test doğruluk grafığı Şekil 4.25(b)'de gösterilmiştir. Tablo 4.15'te eğitim hatası ve test hatası değerleri verilmiştir.



Şekil 4.25. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafığı, b) eğitim ve test doğruluk grafığı

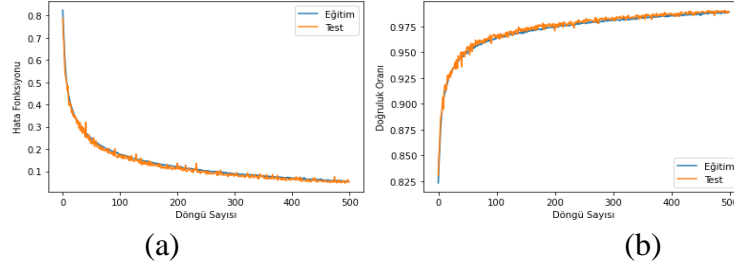
Tablo 4.15. Eğitim ve test hata değerleri (döngü sayısı=100)

Döngü sayısı	Eğitim Hatası	Test Hatası
1	0.8455	0.8799
20	0.3472	0.3694
40	0.2670	0.2714
60	0.2244	0.2240
80	0.1939	0.1949
100	0.1695	0.1602

Tablo 4.15'te eğitim hatası ve test hatası değerleri Şekil 4.25'teki grafikte belirtilen 1, 20, 40, 80, 100. döngü sayılarındaki eğitim hatası ve test hatası değerleridir. Şekil 4.25(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 20'ye ulaştığında eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 65'e ulaştığında eğitim ve test hatasının düşüşünün yavaşladığı gözlenmiştir. Yapılan ilk modellemede, model 100 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak salınımın hala devam ettiği ve modelin öğrenmesinin bitmediği sonucu çıkarılmıştır.

Şekil 4.25(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 20'ye ulaştığında hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 65'e ulaştığında hem test doğruluğunun hem de eğitim doğruluğunun yavaşlayarak arttığı gözlenmiştir. Yapılan ilk modellemede, model 100 döngü sayısına kadar gözlemlenmiş ve doğruluk fonksiyonuna bakılarak salınımın hala devam ettiği ve modelin öğrenmesinin bitmediği sonucu çıkarılmıştır.

Göz ve ağız durum bilgisine dayanan Model-1(b) için (KouBM-DFD-500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.26(a)'da ve göz ve ağız durum bilgisine dayanan Model-1(b) için eğitim ve test doğruluk grafiği Şekil 4.26(b)'de gösterilmiştir.

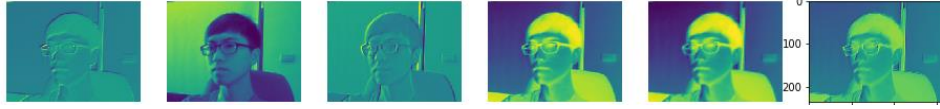


Şekil 4.26. Model-1(b) için, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği

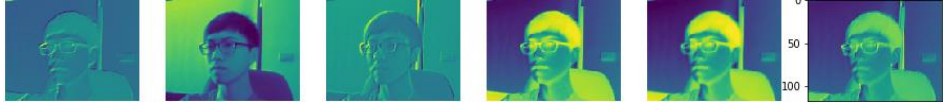
Şekil 4.26(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 20'ye ulaştığında eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 100'e ulaştığında eğitim ve test hatasının azalmasının yavaşladığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı ve değişmediği; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Şekil 4.26(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 20'ye ulaştığında hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 50'ye ulaştığında hem test doğruluğunun hem de eğitim doğruluğunun yavaşlayarak arttığı gözlenmiştir. Grafikte model 500 döngü sayısına kadar gözlemlenmiş ve doğruluk fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı ve değişmediği; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Model-1(b)'de elde edilen konvolüsyon 1. katman çıktı örneği Şekil 4.27'de, havuzlama 1. katman çıktı örneği Şekil 4.28'de, konvolüsyon 2. katman çıktı örneği Şekil 4.29'da, havuzlama 2. katman çıktı örneği Şekil 4.30'da, konvolüsyon 3. katman çıktı örneği Şekil 4.31'de havuzlama 3. katman çıktı örneği Şekil 4.32'de gösterilmiştir. Bu katmanlarda elde edilen kapsamlı hesaplamalar ayrıntılı olarak EK-D'de verilmiştir.



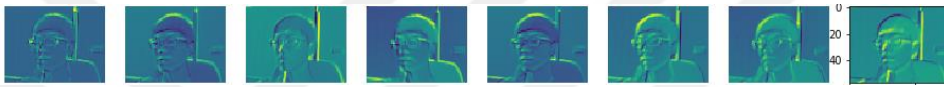
Şekil 4.27. Model-1(b) konvolüsyon 1. katman (6, 236, 316) çıktı örneği



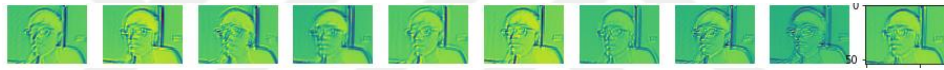
Şekil 4.28. Model-1(b) havuzlama 1. katman (6,118,158) çıktı örneği



Şekil 4.29. Model-1(b) konvolüsyon 2. katman (8,114,154) çıktı örneği



Şekil4.30. Model-1(b) havuzlama 2. katman (8,57,77) çıktı örneği



Şekil 4.31. Model-1(b) konvolüsyon 3. katman (10,54,74) çıktı örneği

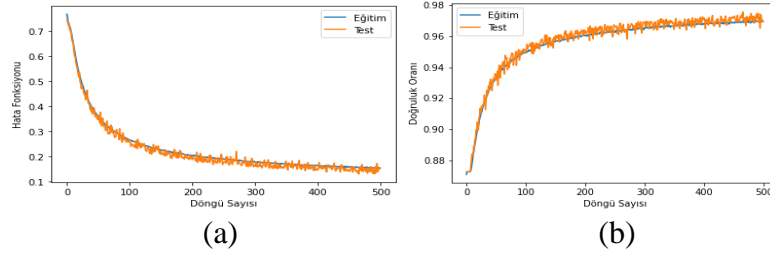


Şekil 4.32. Model-1(b) havuzlama 3. katman (10,27,37) çıktı örneği

Model-2 için eğitim ve test; YawDD, Nthu-DDD ve KouBM-DFD veri kümeleri üzerinde yapılmıştır. Model-2 ağının yapısı Tablo 4.11’de verilmiştir.

Model-2’de konvolüsyon katmanlarında sırasıyla 6, 8 ve 10 adet filtre kullanılmış ve her konvolüsyon katmanı için adım sayısı hiper parametresi 2 olarak değiştirilmiştir. Havuzlama katmanında 2×2 ’lik filtre kullanılmış ve adım sayısı hiper parametresi 2 olarak belirlenmiştir. Bu katmanda maksimum havuzlama yöntemi kullanılmıştır.

Göz ve ağız durum bilgisine dayanan Model-2 için (YawDD-500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.33(a)’da ve göz ve ağız durum bilgisine dayanan Model-2 için eğitim ve test doğruluk grafiği Şekil 4.33(b)’de gösterilmiştir.

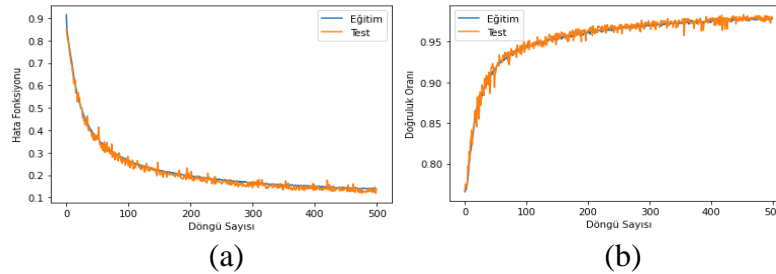


Şekil 4.33. Model-2, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği

Şekil 4.33(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 100'e ulaşıncaya kadar eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 200'e ulaştığında eğitim ve test hatasının düşüşünün yavaşladığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Şekil 4.33(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 100'e ulaşıncaya kadar hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 200'e ulaştığında hem test doğruluğunun hem de eğitim doğruluğunun yavaşlayarak arttığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve doğruluk fonksiyonuna bakılarak modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Model-2 için (Nthu-DDD-500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.34(a)'da ve göz ve ağız durum bilgisine dayanan Model-2 için eğitim ve test doğruluk grafiği Şekil 4.34(b)'de gösterilmiştir.



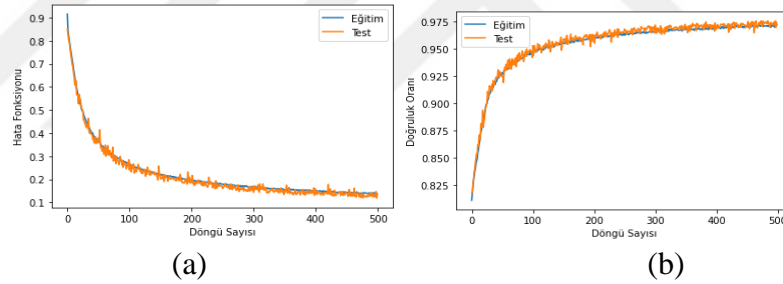
Şekil 4.34. Model-2, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği

Şekil 4.34(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 50'ye ulaştığında eğitim ve test hatasının aynı anda hızlı bir azalma eğilimi gösterdiği

gözlenmiştir. Döngü sayısı yaklaşık 60'a ulaştığında eğitim ve test hatasının düşüşünün yavaşladığı gözlenmiştir. Grafikte 500 döngü sayısına kadar model gözlemlenmiş ve hata fonksiyonuna bakılarak modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Şekil 4.34(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 50'ye ulaştığında hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 60'a ulaştığında hem test doğruluğunun hem de eğitim doğruluğunun yavaşlayarak arttığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve doğruluk fonksiyonuna bakılarak modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Model-2 için (KouBM-DFD-500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.35(a)'da ve göz ve ağız durum bilgisine dayanan Model-2 için eğitim ve test doğruluk grafiği Şekil 4.35(b)'de gösterilmiştir.



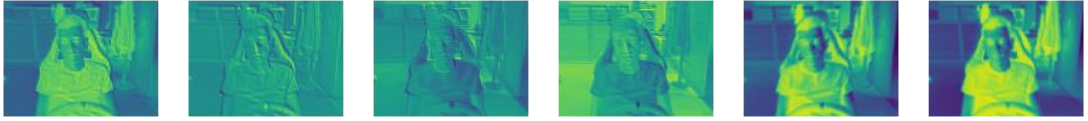
Şekil 4.35. Model-2, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği

Şekil 4.35(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 40'a ulaştığında eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 80'e ulaştığında eğitim ve test hatasının azalmasının yavaşladığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Şekil 4.35(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 40'a ulaştığında hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 80'e ulaştığında hem test doğruluğunun hem de eğitim doğruluğunun yavaşlayarak arttığı gözlenmiştir. Grafikte, model 500

döngü sayısına kadar gözlemlenmiş ve doğruluk fonksiyonuna bakılarak modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

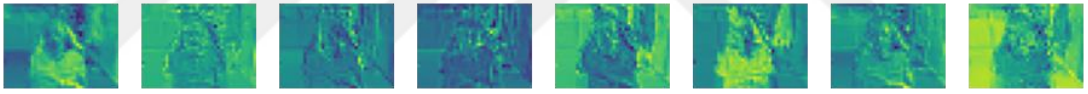
Model-2’de elde edilen konvolüsyon 1. katman çıktı örneği Şekil 4.36’da, havuzlama 1. katman çıktı örneği Şekil 4.37’de, konvolüsyon 2. katman çıktı örneği Şekil 4.38’de, havuzlama 2. katman çıktı örneği Şekil 4.39’da, konvolüsyon 3. katman çıktı örneği Şekil 4.40’ta havuzlama 3. katman çıktı örneği Şekil 4.41’de gösterilmiştir.



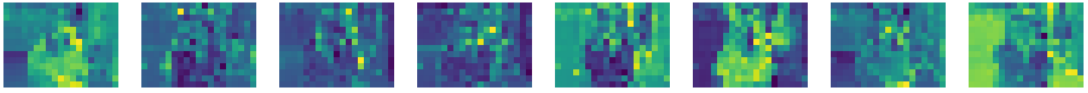
Şekil 4.36. Model-2 konvolüsyon 1. katman (6, 158, 118) çıktı örneği



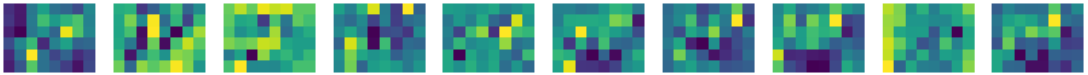
Şekil 4.37. Model-2 havuzlama 1. katman (6, 79, 59) çıktı örneği



Şekil 4.38. Model-2 konvolüsyon 2. katman (8, 38, 28) çıktı örneği



Şekil 4.39. Model-2 havuzlama 2. katman (8, 19, 14) çıktı örneği



Şekil 4.40. Model-2 konvolüsyon 3. katman (10, 8, 6) çıktı örneği



Şekil 4.41. Model-2 havuzlama 3. katman (10, 4, 3) çıktı örneği

Elde edilen sonuçlara bakıldığında konvolüsyon katmanındaki adım sayısının artırılması kaydırma adımının daha büyük olmasını sağlamıştır ve bu modelin avantajı, elde edilen sonuç matrisinin küçülmesi ve bunun da işlem yükünün gözle görülür

biçimde azaltmasıdır. Ancak bu da daha fazla bilginin görmezden gelinmesi anlamına gelmektedir. Bu sebeple ağıın dezavantajı olarak elde edilen doğruluk oranında azalma görülmektedir.

Model-2 için elde edilen doğruluk oranları Tablo 4.16’da gösterilmiştir. Tablo 4.16’daki tablo incelendiğinde 500 döngü sayısı sonunda YawDD test doğruluğu %96,94, Nthu-DDD test doğruluğu %97,97, KouBM-DFD test doğruluğu %97,24 elde edilmiştir.

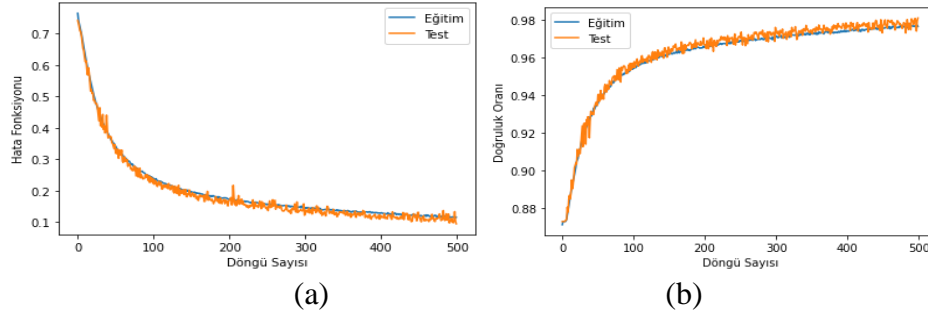
Tablo 4.16. Model-2 için elde edilen doğruluk oranları

Model-2	Doğruluk Oranı			
	Döngü sayısı	YawDD	Nthu-DDD	KouBM-DFD
Eğitim doğruluğu	100	%94,96	%96,55	%94,70
	200	%96,06	%96,20	%95,90
	300	%96,50	%96,88	%96,56
	400	%96,77	%97,30	%96,43
	500	%96,95	%97,80	%97,17
Test doğruluğu	100	%94,91	%94,10	%94,18
	200	%96,16	%96,34	%95,92
	300	%96,90	%97,18	%96,43
	400	%97,07	%97,82	%96,73
	500	%96,94	%97,97	%97,24

Göz ve ağız durum bilgisine dayanan iki çıkışlı Çok Görevli ConNN (Model-3)’in eğitim ve testi; YawDD, Nthu-DDD ve KouBM-DFD veri kümeleri üzerinde yapılmaktadır.

Model-3’te konvolüsyon katmanlarında sırasıyla 6,8 ve 10 adet filtre kullanılmış ve her konvolüsyon katmanı için adım sayısı hiper parametresi 1 olarak değiştirilmiştir. Havuzlama katmanında 4×4’lük filtre uygulanmış ve adım sayısı hiper parametresi 3 olarak değiştirilerek maksimum havuzlama yöntemi kullanılmıştır. Konvolüsyon, aktivasyon ve havuzlama katmanlarını içeren Model-3 ağıının yapısı Tablo 4.11’de verilmiştir.

Model-3 için (YawDD-500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.42(a)’da ve göz ve ağız durum bilgisine dayanan Model-3 için eğitim ve test doğruluk grafiği Şekil 4.42(b)’de gösterilmiştir.

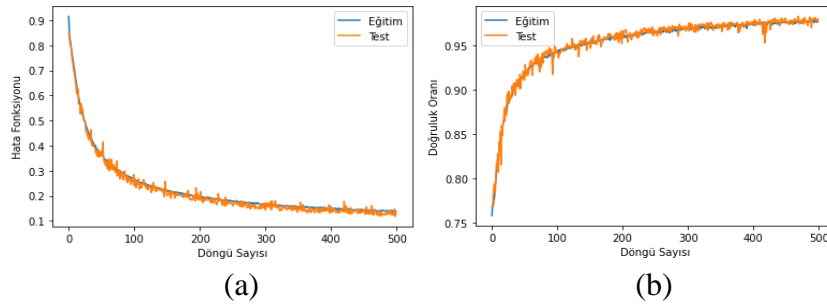


Şekil 4.42. Model-3, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği

Şekil 4.42(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 50'ye ulaşana kadar eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 60'a ulaştığında eğitim ve test hatasının düşüşünün yavaşladığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Şekil 4.42(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 50'ye ulaşana kadar hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 60'a ulaştığında hem test doğruluğunun hem de eğitim doğruluğunun yavaşlayarak arttığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve doğruluk fonksiyonuna bakılarak modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

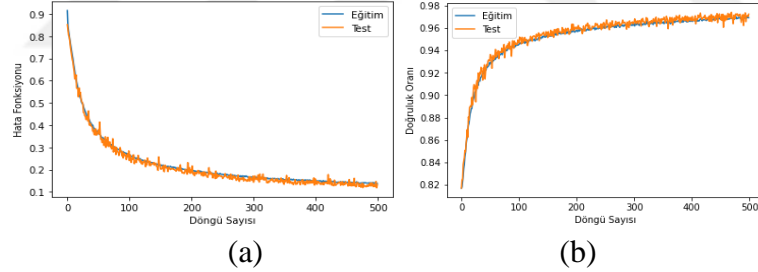
Model-3 için (Nthu-DDD-500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.43(a)'da ve göz ve ağız durum bilgisine dayanan Model-3 için eğitim ve test doğruluk grafiği Şekil 4.43(b)'de gösterilmiştir.



Şekil 4.43. Model-3, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği

Şekil 4.43(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 60'a ulaştığında eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 70'e ulaştığında eğitim ve test hatasının düşüşünün yavaşladığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Şekil 4.43(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 50'ye ulaştığında hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 60'a ulaştığında hem test doğruluğunun hem de eğitim doğruluğunun yavaşlayarak arttığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve doğruluk fonksiyonuna bakılarak modelin öğrenmesinin bittiği sonucu çıkarılmıştır. Model-3 için (KouBM-DFD 500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.44(a)'da ve göz ve ağız durum bilgisine dayanan Model-3 için eğitim ve test doğruluk grafiği Şekil 4.44(b)'de gösterilmiştir.



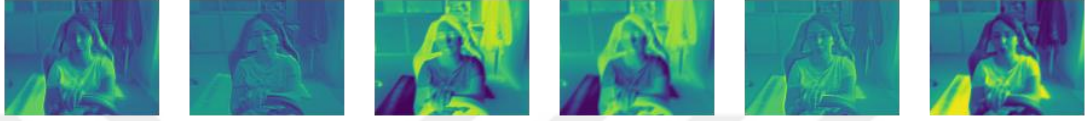
Şekil 4.44. Model-3, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluk grafiği

Şekil 4.44(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 20'ye ulaştığında eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 50'ye ulaştığında eğitim ve test hatasının düşüşünün yavaşladığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

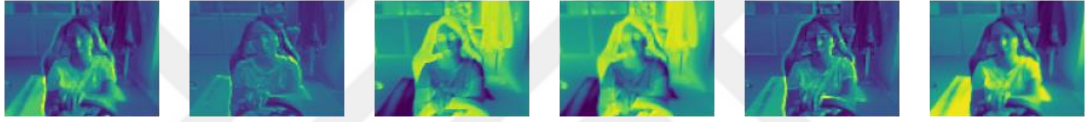
Şekil 4.44(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 20'ye ulaştığında hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 50'ye ulaştığında hem test

doğruluğunun hem de eğitim doğruluğunun yavaşlayarak arttığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve doğruluk fonksiyonuna bakılarak modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

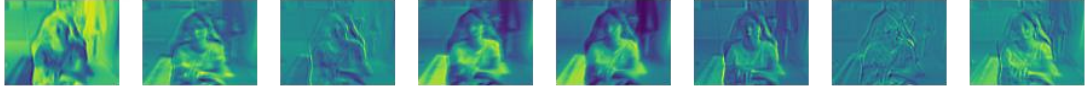
Model-3'te elde edilen konvolüsyon 1. katman çıktı örneği Şekil 4.45'te, havuzlama 1. katman çıktı örneği Şekil 4.46'da, konvolüsyon 2. katman çıktı örneği Şekil 4.47'de, havuzlama 2. katman çıktı örneği Şekil 4.48'de, konvolüsyon 3. katman çıktı örneği Şekil 4.49'da havuzlama 3. katman çıktı örneği Şekil 4.50'de gösterilmiştir.



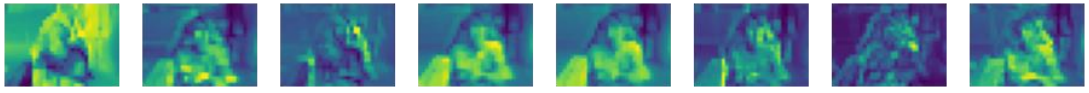
Şekil 4.45. Model-3 konvolüsyon 1. katman (6,316,236) çıktı örneği



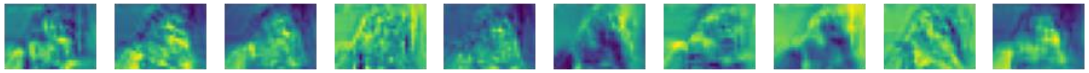
Şekil 4.46. Model-3 havuzlama 1. katman (6,105,78) çıktı örneği



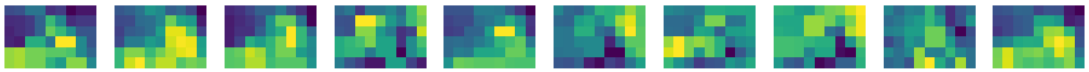
Şekil 4.47. Model-3 konvolüsyon 2. katman (8,102,75) çıktı örneği



Şekil 4.48. Model-3 havuzlama 2. katman (8,33, 24) çıktı örneği



Şekil 4.49. Model-3 konvolüsyon 3. katman (10,30,21) çıktı örneği



Şekil 4.50. Model-3 havuzlama 3. katman (10,9,6) çıktı örneği

Elde edilen sonuçlara bakıldığında havuzlama katmanındaki filtre ve adım sayılarının değiştirilmesi, kaydırma adımının daha büyük olmasını sağlamıştır ve bu modelin avantajı; elde edilen sonuç matrisinin küçülmesi ve bunun da işlem yükünün gözle

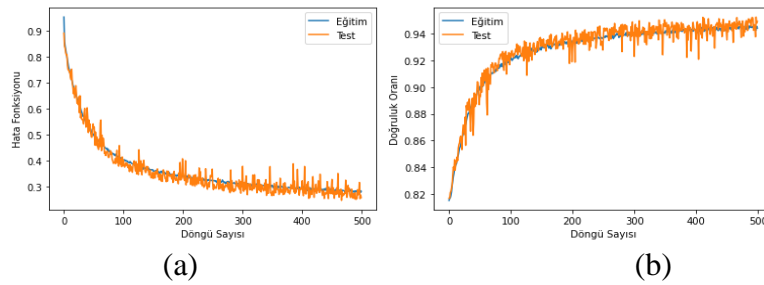
görülür biçimde azalmasına yol açmasıdır. Ancak bu da daha fazla bilginin görmezden gelinmesi anlamına gelmektedir. Bu sebeple ağın dezavantajı olarak elde edilen doğruluk oranında azalma görülmektedir.

Model-3 için elde edilen doğruluk oranları Tablo 4.17’de gösterilmiştir. Tablo 4.17 incelendiğinde 500 döngü sayısı sonunda YawDD test doğruluğu %98,11, Nthu-DDD test doğruluğu %98,01, KouBM-DFD test doğruluğu %97,26 elde edilmiştir.

Tablo 4.17. Model-3 için elde edilen doğruluk oranları

Model-3	Doğruluk Oranı			
	Döngü sayısı	YawDD	Nthu-DDD	KouBM-DFD
Eğitim doğruluğu	100	%95,48	%94,27	%94,50
	200	%96,59	%95,86	%95,75
	300	%97,05	%96,99	%96,44
	400	%97,44	%97,55	%96,70
	500	%97,66	%97,78	%96,94
Test doğruluğu	100	%95,71	%94,46	%94,68
	200	%96,69	%96,67	%96,04
	300	%97,61	%97,03	%96,40
	400	%97,57	%97,51	%96,66
	500	%98,11	%98,01	%97,26

Model-4 için eğitim ve test KouBM-DFD veri kümesi üzerinde yapılmıştır. Model-4 ağıının yapısı Tablo 4.11’de verilmiştir. Model-4’te konvolüsyon katmanlarında sırasıyla 6, 8 ve 10 adet filtre kullanılmış ve konvolüsyon adım sayısı hiper parametresi 2 olarak değiştirilmiştir. Havuzlama katmanında uygulanan filtre 4×4 olarak ve adım sayısı hiper parametresi 3 olarak değiştirilmiştir. Göz ve ağız durum bilgisine dayanan Model-4 için (KouBM-DFD 500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.51(a)’da ve göz ve ağız durum bilgisine dayanan Model-4 için eğitim ve test doğruluk grafiği Şekil 4.51(b)’de gösterilmiştir.



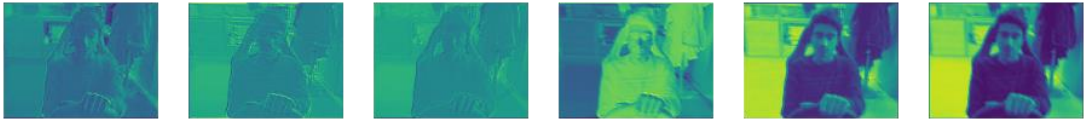
Şekil 4.51. Model-4, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği

Şekil 4.51(a)'da görüldüğü gibi eğitim aşamasındaki döngü sayısı yaklaşık 90'a ulaşana kadar eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 110'a ulaştığında eğitim ve test hatasının düşüşünün yavaşladığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak salınımlar devam etse bile artık değişmediği ve modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

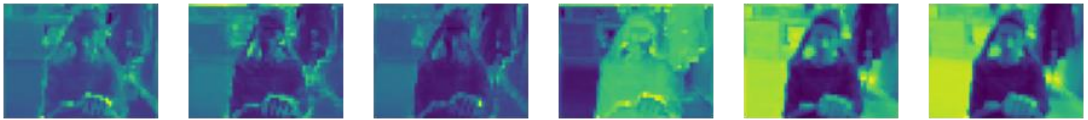
Şekil 4.51(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 90'a ulaşana kadar hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 110'a ulaştığında hem test doğruluğunun hem de eğitim doğruluğunun yavaşlayarak arttığı gözlenmiştir.

Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve doğruluk fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı ve değişmediği; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

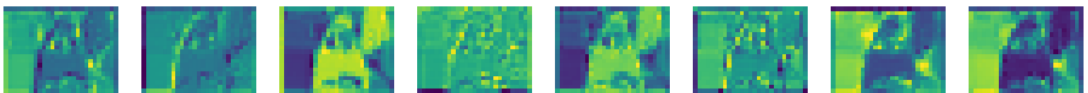
Model-4'te elde edilen konvolüsyon 1. katman çıktı örneği Şekil 4.52'de, havuzlama 1. katman çıktı örneği Şekil 4.53'te, konvolüsyon 2. katman çıktı örneği Şekil 4.54'te, havuzlama 2. katman çıktı örneği Şekil 4.55'te, konvolüsyon 3. katman çıktı örneği Şekil 4.56'da havuzlama 3. katman çıktı örneği Şekil 4.57'de gösterilmiştir.



Şekil 4.52. Model-4 konvolüsyon 1. katman (6, 160, 120) çıktı örneği



Şekil 4.53. Model-4 havuzlama 1. katman (6, 53, 39) çıktı örneği



Şekil 4.54. Model-4 konvolüsyon 2. katman (8, 27, 20) çıktı örneği



Şekil 4.55. Model-4 havuzlama 2. katman (8, 8, 6) çıktı örneği



Şekil 4.56. Model-4 konvolüsyon 3. katman (10, 5, 4) çıktı örneği



Şekil 4.57. Model-4 havuzlama 3. katman (10, 1, 1) çıktı örneği

Elde edilen sonuçlara bakıldığında hem konvolüsyon katmanındaki adım sayısının artırılması hem de havuzlama katmanındaki kaydırma adımının daha büyük olması sebebiyle bu modelin avantajı; elde edilen sonuç matrisinin küçülmesi ve işlem yükünün gözle görülür biçimde azalmasıdır. Ancak bu da daha fazla bilginin görmezden gelinmesi anlamına gelmektedir. Bu sebeple ağır dezavantajı olarak elde edilen doğruluk oranında azalma görülmüştür.

Model-4 için elde edilen doğruluk oranları Tablo 4.18'de gösterilmiştir.

Tablo 4.18. Model-4 eğitiminden elde edilen doğruluk oranları

Model-4	Doğruluk Oranı		
	Döngü Sayısı	Nthu-DDD	KouBM-DFD
Eğitim doğruluğu	100	%92,84	%92,15
	200	%94,72	%93,35
	300	%95,53	%93,93
	400	%95,79	%94,28
	500	%96,27	%94,42
Test doğruluğu	100	%92,25	%92,52
	200	%94,40	%91,89
	300	%95,80	%94,07
	400	%95,96	%94,12
	500	%96,65	%94,92

Tablo 4.18 incelendiğinde 500 döngü sayısı sonunda, Nthu-DDD test doğruluğu %96,65 ve KouBM-DFD test doğruluğu %94,92 elde edilmiştir.

İki Çıkışlı Çok Görevli ConNN ağ modelleri için doğruluk oranları karşılaştırması Tablo 4.19’da gösterilmiştir.

Tablo 4.19. İki Çıkışlı Çok Görevli ConNN ağ modelleri için doğruluk oranları

	Döngü Sayısı	KouBM-DFD (Model-1(b))	KouBM-DFD (Model-2)	KouBM-DFD (Model-3)	KouBM-DFD (Model4)
Eğitim doğruluğu	100	%96,82	%94,70	%94,50	%92,15
	200	%97,50	%95,90	%95,75	%93,35
	300	%98,36	%96,56	%96,44	%93,93
	400	%98,71	%96,43	%96,70	%94,28
	500	%98,94	%97,17	%96,94	%94,42
Test doğruluğu	100	%96,78	%94,18	%94,68	%92,52
	200	%97,48	%95,92	%96,04	%91,89
	300	%98,06	%96,43	%96,40	%94,07
	400	%98,55	%96,73	%96,66	%94,12
	500	%98,88	%97,24	%97,26	%94,92

Tablo 4.19 incelendiğinde 500 döngü sayısı sonunda Model-1(b) için KouBM-DFD üzerindeki test doğruluğu %98,88, Model-2 için KouBM-DFD’deki test doğruluğu %97,24, Model-3 için KouBM-DFD üzerindeki test doğruluğu %97,26 ve Model-4 için KouBM-DFD veri kümesindeki test doğruluğu %94,92’dir. Tablo 4.20’de İki Çıkışlı Çok Görevli ConNN ağ modelleri eğitimlerinin zaman ve doğruluk oranı karşılaştırması verilmiştir.

Tablo 4.20. İki Çıkışlı Çok Görevli ConNN ağ modelleri eğitimlerinin zaman ve doğruluk oranı karşılaştırması

İki Çıkışlı Çok Görevli ConNN (KouBM-DFD)	Döngü sayısı=1	Döngü sayısı=500	Döngü sayısı=500
	Zaman	Zaman	Doğruluk oranı
Model-1(b)	7 dk 11 sn	2 gün 11 sa 51 dk 40sn	%98,88
Model-2	7 dk 2 sn	2 gün 10 sa 16 dk 40sn	%97,24
Model-3	7 dk 5 sn	2 gün 10 sa 45 dk	%97,26
Model-4	7 dk 7 sn	2 gün 11 sa 01 dk 40 sn	%94,92

İki Çıkışlı Çok Görevli ConNN ağ modellerinin eğitimlerinin zaman ve doğruluk oranları karşılaştırıldığında; konvolüsyon katmanlarındaki adım sayılarının ya da havuzlama katmanındaki kaydırma adımlarının değişiminin, ağın çalışması için gereken süre ve elde edilen doğruluk oranına etkisinin ters orantılı olduğu söylenebilmektedir.

Katmanlarda bulunan filtreleri kaydırmak için gerekli adım sayısı artarken eğitim için gereken toplam sürenin de kısaldığı ancak doğruluk oranının da buna bağlı azaldığı tespit edilmiştir. Göz ve ağız durum bilgisine dayanan iki çıkışlı çok görevli ConNN

ağ modellerinden elde edilen doğruluk oranlarının literatürde bulunan diğer yaklaşımlarla karşılaştırılması Tablo 4.21’de gösterilmiştir.

Elde edilen sonuçların diğer yaklaşımlardan daha başarılı olduğu tespit edilmiş ve önerilen modelin diğer modeller kadar güçlü olduğu gözlemlenmiştir.

Tablo 4.21. Doğruluk oranlarının literatürde bulunan diğer yaklaşımlarla karşılaştırılması

Literatür	Yıl	Yöntem	Veri kümesi	Doğruluk oranı		
Zhang ve diğ., 2015	2015	CNN	Kendi oluşturdukları veri kümesi, YawDD	%92		
Gu ve diğ., 2018	2018	Hiyerarşik CNN	EMD, ZJU, CEW	%96,89		
Houssaini ve diğ., 2019	2019	SVM	IBUG 300-W	%83		
Geng ve diğ., 2019	2019	MTCNN	TJPU-FDD, Brain4Cars, ZJU	%92,47		
Önerilen yöntem	2020	İki Çıkışlı Çok Görevli ConNN (Model-1(b))	YawDD, Nthu-DDD, KouBM-DFD	%99,18	%98,93	%98,88

4.3.4.2. Üç çıkışlı çok görevli ConNN ile yorgunluk tahmin sistemi

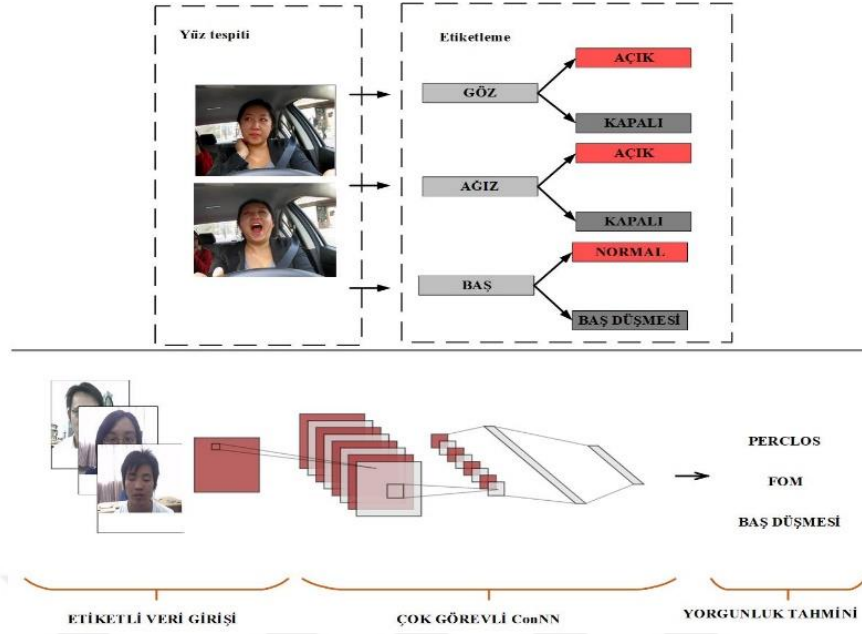
Üç Çıkışlı Çok Görevli ConNN modelinde kullanılan Nthu-DDD ve KouBm-DFD veri kümelerinin dağılımı Tablo 4.22’de gösterilmiştir.

Çalışmada eğitim aşamasında; göz, ağız ve kafa durum bilgileri aynı anda sınıflandırıldığı için Nthu-DDD ve KouBM-DFD veri kümeleri içerisinde bulunan güneş gözlüklü sürücü verileri kullanılamamıştır. Modellerdeki doğruluk hesabı Eşitlik (4.7)’de verilmiştir.

Tablo 4.22. Veri kümeleri dağılımı

Veri kümesi	Nthu-DDD	KouBm-DFD
Eğitim	28943	26694
Test	4964	5730
Toplam	33907	32424

Önerilen Göz, ağız ve kafa durum bilgilerine dayanan sistemin genel yapısı Şekil 4.58’de gösterilmiştir.



Şekil 4.58. Göz, ağız ve kafa durum bilgilerine dayanan sistemin genel yapısı

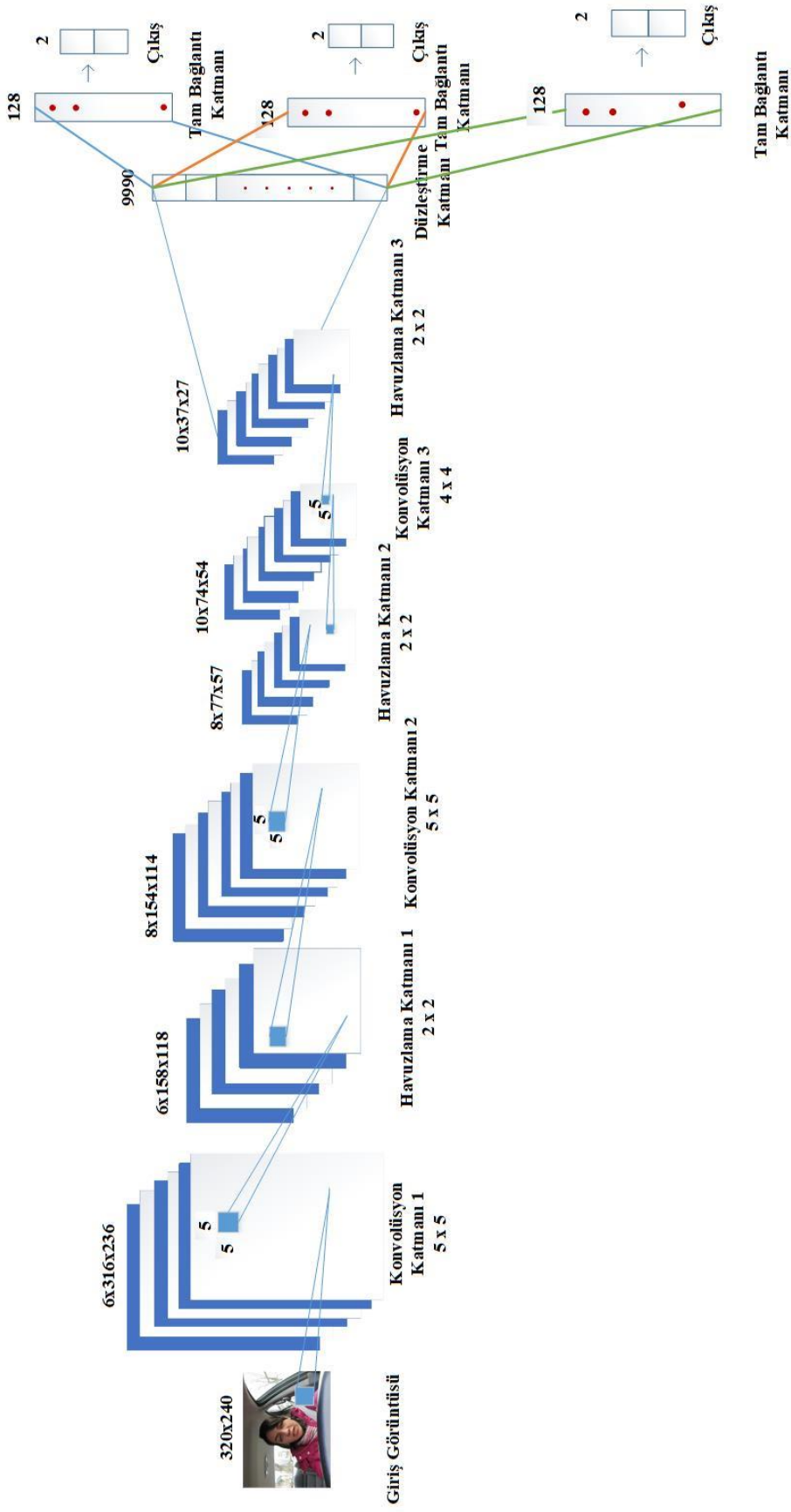
Bu sistem için tasarlanan modellerin ağ yapıları Tablo 4.23'te verilmiş olup değiştirilen hiper parametreler koyu renkli belirtilmiştir. Göz, ağız ve kafa durum bilgisine dayanan Üç Çıkışlı Çok Görevli ConNN modeli Şekil 4.59'da gösterilmiştir.

Tablo 4.23. Modellerin ağ yapıları

Model	Katman adı	Katman Tipi	Parametreler
Model-1(a)	Conv1	Konvolüsyon	5×5, 3 , adım sayısı=1
	Pool1	Havuzlama	2×2, maksimum havuzlama, adım sayısı=2
	A1	Aktivasyon	ReLU
	Conv2	Konvolüsyon	5×5, 4 , adım sayısı=1
	Pool2	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A2	Aktivasyon	ReLU
	Conv3	Konvolüsyon	4×4, 5 , adım sayısı =1
	Pool3	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A3	Aktivasyon	ReLU
FullyCon1	Tam bağlantı	(128, 2d), (128, 2d), (128, 2d)	
Model-1(b)	Conv1	Konvolüsyon	5×5, 6 , adım sayısı=1
	Pool1	Havuzlama	2×2, maksimum havuzlama, adım sayısı=2
	A1	Aktivasyon	ReLU
	Conv2	Konvolüsyon	5×5, 8 , adım sayısı=1
	Pool2	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A2	Aktivasyon	ReLU
	Conv3	Konvolüsyon	4×4, 10 , adım sayısı =1
	Pool3	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A3	Aktivasyon	ReLU
FullyCon1	Tam bağlantı	(128, 2d), (128, 2d), (128, 2d)	

Tablo 4.23. (Devam) Modellerin ağ yapıları

Model	Katman adı	Katman Tipi	Parametreler
Model-1(c)	Conv1	Konvolüsyon	5×5, 12 , adım sayısı=1
	Pool1	Havuzlama	2×2, maksimum havuzlama, adım sayısı=2
	A1	Aktivasyon	ReLU
	Conv2	Konvolüsyon	5×5, 16 , adım sayısı=1
	Pool2	Havuzlama	2×2, maksimum havuzlama, adım sayısı =2
	A2	Aktivasyon	ReLU
	Conv3	Konvolüsyon	4×4, 20 , adım sayısı =1
	Pool3	Havuzlama	2×2, maksimum havuzlama, stride=2
	A3	Aktivasyon	ReLU
FullyCon1	Tam bağlantı	(128, 2d), (128, 2d), (128, 2d)	
Model-2	Conv1	Konvolüsyon	5×5, 6 , adım sayısı =2
	Pool1	Havuzlama	2×2, maksimum havuzlama, adım sayısı=2
	A1	Aktivasyon	ReLU
	Conv2	Konvolüsyon	5×5, 8 , adım sayısı =2
	Pool2	Havuzlama	2×2, maksimum havuzlama, adım sayısı=2
	A2	Aktivasyon	ReLU
	Conv3	Konvolüsyon	4×4, 10 , adım sayısı =2
	Pool3	Havuzlama	2×2, maksimum havuzlama, adım sayısı=2
	A3	Aktivasyon	ReLU
FullyCon1	Tam bağlantı	(128, 2d), (128, 2d), (128, 2d)	
Model-3	Conv1	Konvolüsyon	5×5, 6 , adım sayısı =1
	Pool1	Havuzlama	4×4, maksimum havuzlama, adım sayısı=3
	A1	Aktivasyon	ReLU
	Conv2	Konvolüsyon	5×5, 8 , adım sayısı =1
	Pool2	Havuzlama	4×4, maksimum havuzlama, adım sayısı =3
	A2	Aktivasyon	ReLU
	Conv3	Konvolüsyon	4×4, 10 , adım sayısı =1
	Pool3	Havuzlama	4×4, maksimum havuzlama, adım sayısı =3
	A3	Aktivasyon	ReLU
FullyCon1	Tam bağlantı	(128, 2d), (128, 2d), (128, 2d)	
Model-4	Conv1	Konvolüsyon	5×5, 6 , adım sayısı=2
	Pool1	Havuzlama	4×4, maksimum havuzlama, adım sayısı=3
	A1	Aktivasyon	ReLU
	Conv2	Konvolüsyon	5×5, 8 , adım sayısı=2
	Pool2	Havuzlama	4×4, maksimum havuzlama, adım sayısı=3
	A2	Aktivasyon	ReLU
	Conv3	Konvolüsyon	4×4, 10 , adım sayısı=2
	Pool3	Havuzlama	4×4, maksimum havuzlama, adım sayısı=3
	A3	Aktivasyon	ReLU
FullyCon1	Tam bağlantı	(128, 2d), (128, 2d), (128, 2d)	



Şekil 4.59. Üç çıkışlı Çok Görevli ConNN modeli

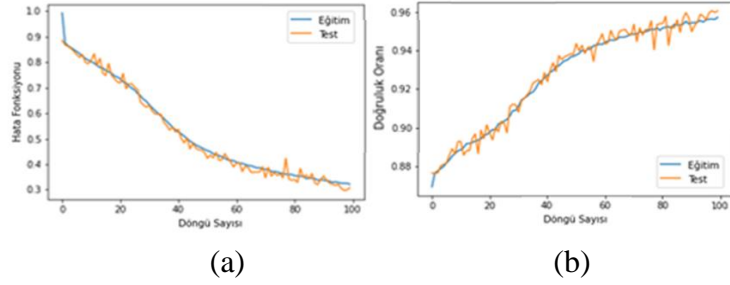
Model-1 için hiper parametreler değiştirilerek Model-1(a), Model-1(b) ve Model-1(c) alt modelleri oluşturulmuştur. Bu modellerden elde edilen doğruluk oranlarının karşılaştırılması Tablo 4.24'te gösterilmiştir.

Tablo 4.24 Model-1 için elde edilen doğruluk oranları

Model	Üç çıkışlı Çok Görevli ConNN	Doğruluk Oranı		
		Döngü Sayısı	Nthu-DDD	KouBM-DFD
Model-1(a)	Eğitim doğruluğu	100	%95,98	%96,28
		200	%96,25	%96,99
		300	%97,60	%97,33
		400	%97,90	%97,80
		500	%98,17	%98,06
	Test doğruluğu	100	%95,54	%95,37
		200	%96,25	%96,90
		300	%97,75	%97,60
		400	%97,87	%97,78
		500	%97,34	%98,12
Model-1(b)	Eğitim doğruluğu	100	%97,61	%97,38
		200	%98,59	%98,22
		300	%99,01	%98,78
		400	%99,26	%99,05
		500	%99,30	%99,35
	Test doğruluğu	100	%97,41	%97,22
		200	%98,41	%98,25
		300	%98,80	%98,70
		400	%99,02	%99,00
		500	%99,27	%99,23
Model-1(c)	Eğitim doğruluğu	100	%96,92	%96,22
		200	%97,19	%96,48
		300	%96,88	%96,73
		400	%97,26	%97,04
		500	%97,84	%97,56
	Test doğruluğu	100	%95,73	%95,28
		200	%95,98	%95,56
		300	%96,80	%96,36
		400	%97,23	%97,80
		500	%97,35	%97,04

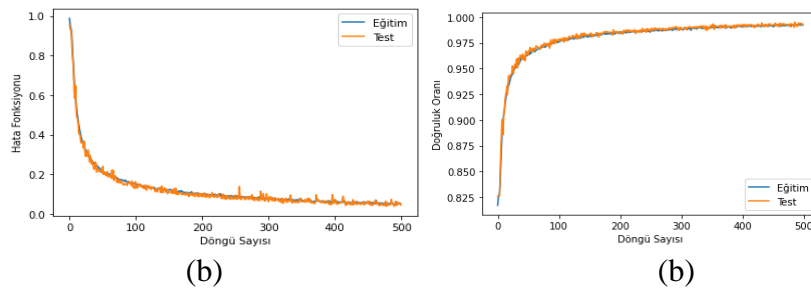
Test doğruluğu için en iyi doğruluk oranları Model-1(b)'de elde edilmiştir. Bu nedenle tezde Üç Çıkışlı Çok Görevli ConNN modeli için Model-1(b) ağı tercih edilmiştir. Tablo 4.24 incelendiğinde; 500 döngü sayısı sonunda test doğruluğu için en iyi doğruluk oranları Model-1(b)'de elde edilmiştir. Nthu-DDD test doğruluğu %99,27, KouBM-DFD test doğruluğu %99,23 olarak elde edilmiştir. Göz, ağız ve kafa durum bilgisine dayanan bağlı Üç Çıkışlı Çok Görevli ConNN (Model-1(b)) eğitimi ve testi Nthu-DDD ve KouBM-DFD veri kümeleri üzerinde yapılmaktadır. Model-1(b) için (KouBM-DFD 100 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu)

(loss function) grafiği Şekil 4.60(a)'da ve göz, ağız ve kafa durum bilgisine dayanan Model-1(b) eğitim ve test doğruluk grafiği Şekil 4.60(b)'de gösterilmiştir.



Şekil 4.60. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği

Şekil 4.60(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 60'a ulaştığında eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 100'e ulaştığında eğitim ve test hatasının düşüş eğiliminin azaldığı belirlenmiştir. Yapılan ilk modellemede, model 100 döngü sayısına kadar gözlemlenmiş ancak hata fonksiyonuna bakılarak bir yörüngeye oturmadığı ve modelin öğrenmesinin bitmediği devam ettiği sonucu çıkarılmıştır. Şekil 4.60(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 60'a ulaştığında hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 100'e ulaştığında hem eğitim hem de test doğruluğundaki artışın yavaşladığı gözlenmiştir. Yapılan ilk modellemede, model 100 döngü sayısına kadar gözlemlenmiş ancak doğruluk fonksiyonuna bakılarak bir yörüngeye oturmadığı ve modelin öğrenmesinin bitmediği devam ettiği sonucu çıkarılmıştır. Model-1(b) için (KouBM-DFD-500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) grafiği Şekil 4.61(a)'da ve göz, ağız ve kafa durum bilgisine dayanan Model-1(b) için eğitim ve test doğruluk grafiği Şekil 4.61(b)'de gösterilmiştir.

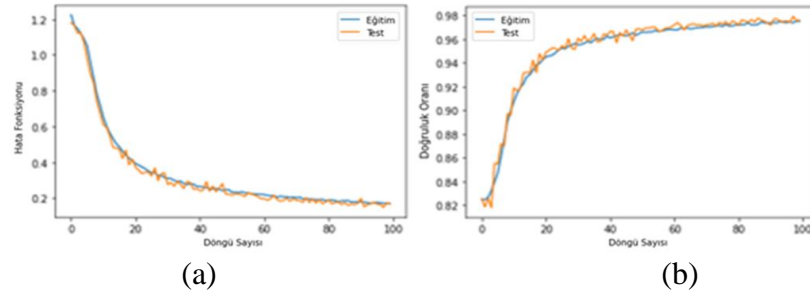


Şekil 4.61. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği

Şekil 4.61(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 80'e ulaşıncaya kadar eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlemlenmiştir. Döngü sayısı yaklaşık 100'e ulaştığında eğitim ve test hatasının düşüş eğiliminin yavaşladığı gözlemlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı ve değişmediği; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Şekil 4.61(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 80'e ulaşıncaya kadar hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlemlenmiştir. Döngü sayısı yaklaşık 100'e ulaştığında hem eğitim hem de test doğruluğundaki artışın yavaşladığı gözlemlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve doğruluk fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı ve değişmediği; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Nthu-DDD veri kümesi üzerinde elde edilen sonuçlar Şekil 4.62 ve Şekil 4.63'te gösterilmiştir. Model-1(b) için (Nthu-DDD-100 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.62(a)'da ve göz, ağız ve kafa durum bilgisine dayanan Model-1(b) için eğitim ve test doğruluk grafiği Şekil 4.62(b)'de gösterilmiştir.

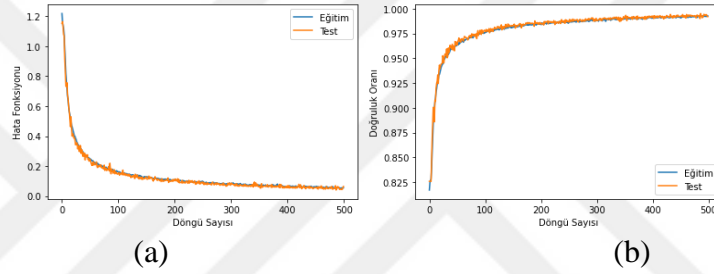


Şekil 4.62. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği

Şekil 4.62(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 20'ye ulaştığında eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlemlenmiştir. Döngü sayısı yaklaşık 50'ye ulaştığında eğitim ve test hatasının düşüş eğiliminin yavaşladığı belirlenmiştir. Grafikte, model 100 döngü sayısına kadar model gözlemlenmiş ve hata fonksiyonuna bakılarak salınımlar devam etse bile değişmediği bu sebeple modelin öğrenmesinin bitmiş olabileceği sonucu çıkarılmıştır.

Şekil 4.62(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 20'ye ulaştığında hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği fark edilmektedir. Döngü sayısı yaklaşık 50'ye ulaştığında hem eğitim hem de test doğruluğundaki artışın yavaşladığı gözlenmiştir. Grafikte, model 100 döngü sayısına kadar model gözlemlenmiş ve doğruluk fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı ve değişmediği; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Model-1(b) için (Nthu-DDD-500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.63(a)'da ve Model-1(b) için eğitim ve test doğruluk grafiği Şekil 4.63(b)'de gösterilmiştir.

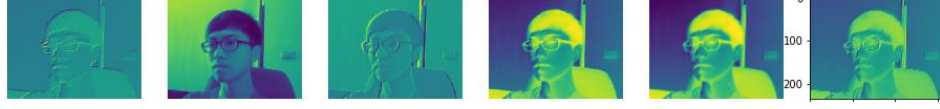


Şekil 4.63. Model-1(b), a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği

Şekil 4.63(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 35'e ulaştığında eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 50'ye ulaştığında eğitim ve test hatasının düşüşünün yavaşladığı fark edilmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak salınımlar devam etse bile artık değişmediği bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Şekil 4.63(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 35'e ulaştığında hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği belirlenmiştir. Döngü sayısı yaklaşık 50'ye ulaştığında test doğruluğunun bu hızlı yükselmesinin yavaşladığı ve eğitim doğruluğunun yaklaşık 40 döngü sayısına ulaştığında yavaşladığı fark edilmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak salınımların artık aynı şekilde kaldığı ve değişmediği; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Model-1(b)'de elde edilen konvolüsyon 1. katman çıktı örneği Şekil 4.64'te, havuzlama 1. katman çıktı örneği Şekil 4.65'te, konvolüsyon 2. katman çıktı örneği Şekil 4.66'da, havuzlama 2. katman çıktı örneği Şekil 4.67'de, konvolüsyon 3. katman çıktı örneği Şekil 4.68'te havuzlama 3. katman çıktı örneği Şekil 4.69'da gösterilmiştir.



Şekil 4.64. Model-1(b) konvolüsyon 1. katman (6, 316, 236) çıktı örneği



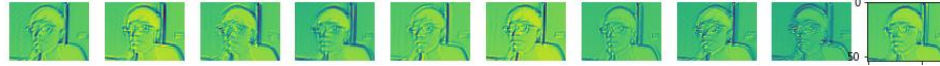
Şekil 4.65. Model-1(b) havuzlama 1. katman (6, 158, 118) çıktı örneği



Şekil 4.66. Model-1(b) konvolüsyon 2. katman (8, 154, 114) çıktı örneği



Şekil 4.67. Model-1(b) havuzlama 2. katman (6, 77, 57) çıktı örneği



Şekil 4.68. Model-1(b) konvolüsyon 3. katman (10, 74, 54) çıktı örneği

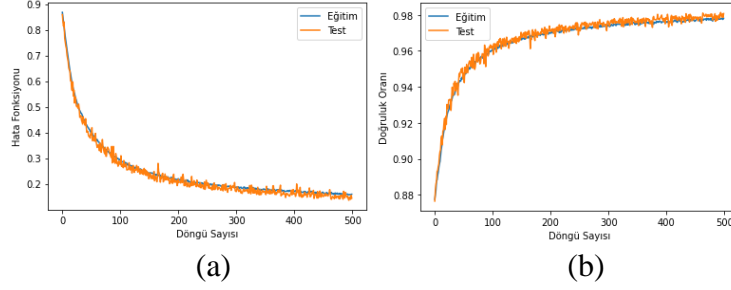


Şekil 4.69. Model-1(b) havuzlama 3. katman (10, 37, 27) çıktı örneği

Model-2 ağ yapısı Tablo 4.23'te verilmiştir. Model-2'de konvolüsyon katmanlarında sırasıyla 6, 8 ve 10 adet filtre kullanılmış ve her katmanda adım sayısı hiper parametresi 2 olarak değiştirilmiştir. Havuzlama katmanında 2×2 'lik filtre kullanılmış ve 2 adım kaydırmalı maksimum havuzlama yöntemi kullanılmıştır.

Model-2 eğitimi ve testi KouBM-DFD veri kümesi üzerinde test edilmiştir. Göz, ağız ve kafa durum bilgisine dayanan Model-2 için (KouBM-DFD-500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil

4.70(a)'da ve göz, ağız ve kafa durum bilgisine dayanan Model-2 için eğitim ve test doğruluk grafiği Şekil 4.70(b)'de gösterilmiştir.

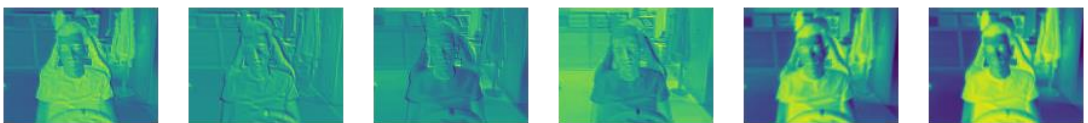


Şekil 4.70. Model-2, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği

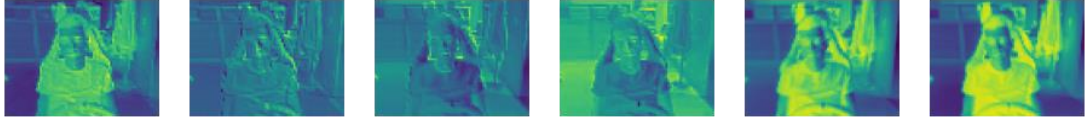
Şekil 4.70(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 70'e ulaşıncaya kadar eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 90'a ulaştığında eğitim ve test hatasının düşüşünün yavaşladığı gözlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı ve değişmediği; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Şekil 4.70(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 70'e ulaşıncaya kadar hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlenmiştir. Döngü sayısı yaklaşık 90'a ulaştığında hem test doğruluğundaki hem de eğitim doğruluğundaki artışın yavaşladığı gözlemlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve doğruluk fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı ve değişmediği; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

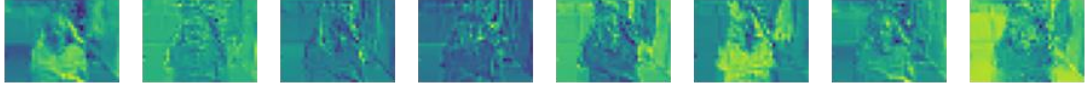
Model-2'de elde edilen konvolüsyon 1. katman çıktı örneği Şekil 4.71'de, havuzlama 1. katman çıktı örneği Şekil 4.72'de, konvolüsyon 2. katman çıktı örneği Şekil 4.73'te, havuzlama 2. katman çıktı örneği Şekil 4.74'te konvolüsyon 3. katman çıktı örneği Şekil 4.75'te havuzlama 3. katman çıktı örneği Şekil 4.76'da gösterilmiştir.



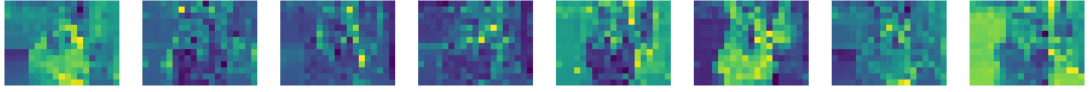
Şekil 4.71. Model-2 konvolüsyon 1. katman (6,158,118) çıktı örneği



Şekil 4.72. Model-2 havuzlama 1. katman (6,79,59) çıktı örneği



Şekil 4.73. Model-2 konvolüsyon 2. katman (8,38,28) çıktı örneği



Şekil 4.74. Model-2 havuzlama 2. katman (8,19, 14) çıktı örneği



Şekil 4.75. Model-2 konvolüsyon 3. katman (10,8,6) çıktı örneği

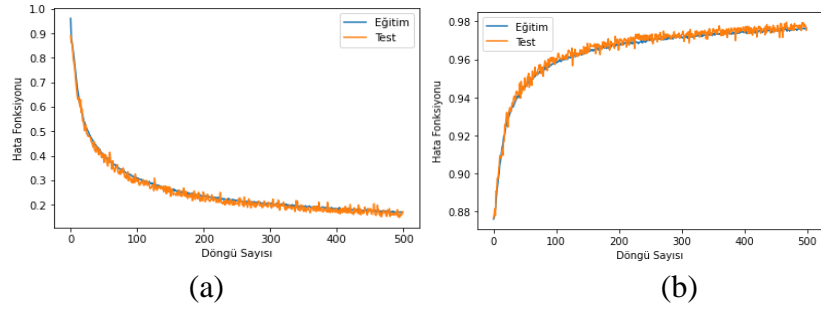


Şekil 4.76. Model-2 havuzlama 3. katman (10,4,3) çıktı örneği

Elde edilen sonuçlara bakıldığında konvolüsyon katmanındaki adım sayısının artırılması kaydırma adımının daha büyük olmasını sağlamıştır ve bu modelin avantajı; elde edilen sonuç matrisinin küçülmesi ve bunun da işlem yükünün gözle görülür biçimde azaltmasıdır. Ancak bu daha fazla bilginin görmezden gelinmesi anlamına gelmektedir. Bu sebeple ağıın dezavantajı olarak elde edilen doğruluk oranında azalma görülmüştür.

Model-3 ağının yapısı Tablo 4.23'te verilmiştir. Model-3'te konvolüsyon katmanlarında sırasıyla 6, 8 ve 10 adet filtre kullanılmış ve her katmanda adım sayısı hiper parametresi 1 olarak değiştirilmiştir. Havuzlama katmanında 4×4'lük filtre kullanılmış ve adım sayısı hiper parametresi 3 olarak değiştirilerek maksimum havuzlama yöntemi kullanılmıştır.

Model-3 için (KouBM-DFD-500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.77(a)'da ve Model-3 için eğitim ve test doğruluk grafiği Şekil 4.77(b)'de gösterilmiştir.



Şekil 4.77. Model-3, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği

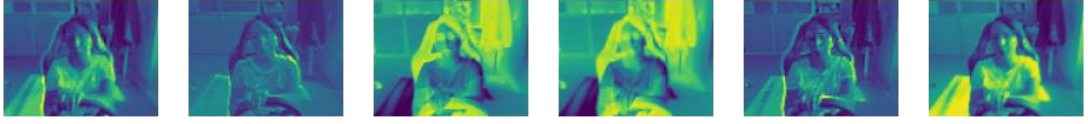
Şekil 4.77(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 50'ye ulaşıncaya kadar eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlemlenmiştir. Döngü sayısı yaklaşık 150'ye ulaştığında eğitim ve test hatasının düşüşünün yavaşladığı gözlemlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı ve değişmediği; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Şekil 4.77(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 50'ye ulaşıncaya kadar hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlemlenmiştir. Döngü sayısı yaklaşık 150'ye ulaştığında hem test hem de eğitim doğruluğundaki artışın yavaşladığı gözlemlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve doğruluk fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı ve değişmediği; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

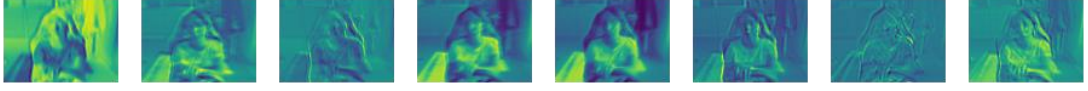
Model-3'te elde edilen konvolüsyon 1. katman çıktı örneği Şekil 4.78'de, havuzlama 1. katman çıktı örneği Şekil 4.79'da, konvolüsyon 2. katman çıktı örneği Şekil 4.80'de, havuzlama 2. katman çıktı örneği Şekil 4.81'de, konvolüsyon 3. katman çıktı örneği Şekil 4.82'de havuzlama 3. katman çıktı örneği Şekil 4.83'te gösterilmiştir.



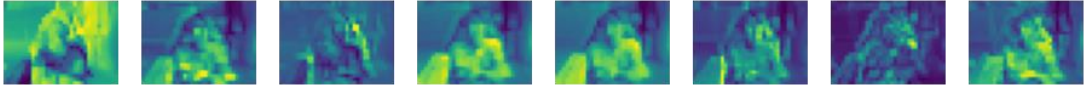
Şekil 4.78. Model-3 konvolüsyon 1. katman (6, 316, 236) çıktı örneği



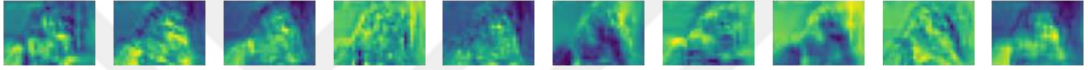
Şekil 4.79. Model-3 havuzlama 1. katman (6, 105, 78) çıktı örneği



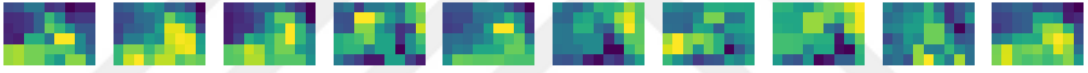
Şekil 4.80. Model-3 konvolüsyon 2. katman (8, 102, 75) çıktı örneği



Şekil 4.81. Model-3 havuzlama 2. katman (8, 33, 24) çıktı örneği



Şekil 4.82. Model-3 konvolüsyon 3. katman (10, 30, 21) çıktı örneği

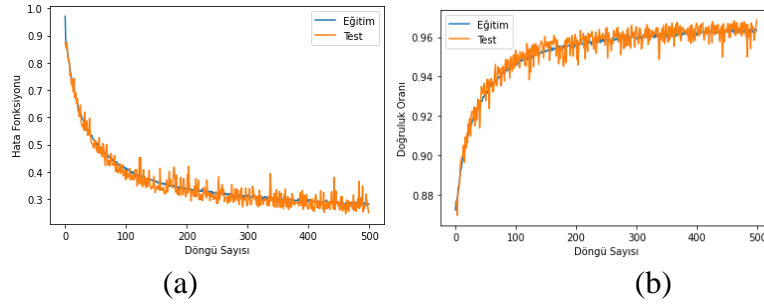


Şekil 4.83. Model-3 havuzlama 3. katman (10, 9, 6) çıktı örneği

Elde edilen sonuçlara bakıldığında havuzlama katmanındaki filtre ve adım sayılarının değiştirilmesi kaydırma adımının daha büyük olmasını sağlamıştır ve bu modelin avantajı, elde edilen sonuç matrisinin küçülmesi ve bunun da işlem yükünün gözle görülür biçimde azalmasına yol açmasıdır. Ancak bu da daha fazla bilginin görmezden gelinmesi anlamına gelmektedir. Bu sebeple ağır dezavantajı olarak elde edilen doğruluk oranında azalma görülmektedir.

Model-4 ağının yapısı Tablo 4.23'te verilmiştir. Model-4'te konvolüsyon katmanlarında sırasıyla 6,8 ve 10 adet filtre kullanılmış ve her katmanda adım sayısı hiper parametresi 2 olarak değiştirilmiştir. Havuzlama katmanında filtre 4×4 uygulanmış ve adım sayısı hiper parametresi 3 olarak değiştirilmiştir.

Model-4 için (KouBM-DFD-500 döngü sayısı) eğitim ve test ölçüt fonksiyonu (hata fonksiyonu) (loss function) grafiği Şekil 4.84(a)'da ve Model-4 için eğitim ve test doğruluk grafiği Şekil 4.84(b)'de gösterilmiştir.

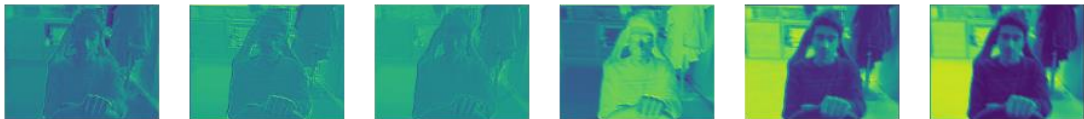


Şekil 4.84. Model-4, a) eğitim ve test ölçüt fonksiyonu grafiği, b) eğitim ve test doğruluğu grafiği

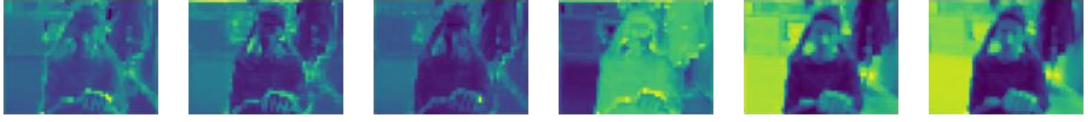
Şekil 4.84(a)'da görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 80'e ulaşıncaya kadar eğitim ve test hatasının aynı anda hızlı bir düşüş eğilimi gösterdiği gözlemlenmiştir. Döngü sayısı yaklaşık 100'e ulaştığında eğitim ve test hatasının düşüşünün yavaşladığı gözlemlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve hata fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı ve değişmediği; bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

Şekil 4.84(b)'de görüldüğü gibi eğitim aşamasında döngü sayısı yaklaşık 80'e ulaşıncaya kadar hem eğitim hem de test doğruluğunun aynı anda hızlı bir artış eğilimi gösterdiği gözlemlenmiştir. Döngü sayısı yaklaşık 100'e ulaştığında hem test hem de eğitim doğruluğundaki artışın yavaşladığı gözlemlenmiştir. Grafikte, model 500 döngü sayısına kadar gözlemlenmiş ve doğruluk fonksiyonuna bakılarak salınımlar devam etse bile artık aynı şekilde kaldığı ve değişmediği bu sebeple modelin öğrenmesinin bittiği sonucu çıkarılmıştır.

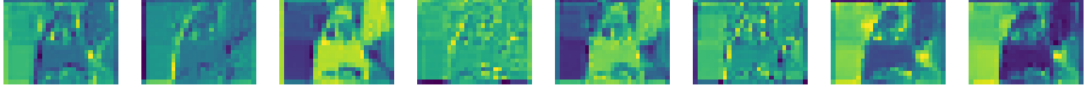
Model-4'te elde edilen konvolüsyon 1. katman çıktı örneği Şekil 4.85'te, havuzlama 1. katman çıktı örneği Şekil 4.86'da, konvolüsyon 2. katman çıktı örneği Şekil 4.87'de, havuzlama 2. katman çıktı örneği Şekil 4.88'de, konvolüsyon 3. katman çıktı örneği Şekil 4.89'da havuzlama 3. katman çıktı örneği Şekil 4.90'da gösterilmiştir.



Şekil 4.85. Model-4 konvolüsyon 1. katman (6,160,120) çıktı örneği



Şekil 4.86. Model-4 havuzlama 1. katman (6,53,39) çıktı örneği



Şekil 4.87. Model-4 konvolüsyon 2. katman (8,27,20) çıktı örneği



Şekil 4.88. Model-4 havuzlama 2. katman (8, 8, 6) çıktı örneği



Şekil 4.89. Model-4 konvolüsyon 3. katman (10, 5, 4) çıktı örneği



Şekil 4.90. Model-4 havuzlama 3. katman (10, 1, 1) çıktı örneği

Elde edilen sonuçlara bakıldığında hem konvolüsyon katmanındaki adım sayısının artırılması hem de havuzlama katmanındaki kaydırma adımının daha büyük olması sebebiyle bu modelin avantajı; elde edilen sonuç matrisinin küçülmesi ve işlem yükünün gözle görülür biçimde azalmasıdır. Ancak bu da daha fazla bilginin görmezden gelinmesi anlamına gelir. Bu sebeple ağıın dezavantajı olarak elde edilen doğruluk oranında azalma görülmüştür.

İki Çıkışlı Çok Görevli ConNN ağ modelleri için KouBM-DFD veri kümesi üzerinde doğruluk oranları karşılaştırması Tablo 4.25'te gösterilmiştir.

Tablo 4.25 değerlendirildiğinde 500 döngü sayısı sonunda Model-1(b) için KouBM-DFD'deki test doğruluğu %99,23, Model-2 için KouBM-DFD üzerindeki test doğruluğu %98,07, Model-3 için KouBM-DFD'deki test doğruluğu %97,53 ve Model-4 için KouBM-DFD'deki test doğruluğu %96,87 şeklinde elde edilmiştir.

Tablo 4.25. Üç Çıkışlı Çok Görevli ConNN ağ modelleri için doğruluk oranları

Üç çıkışlı Çok Görevli ConNN	Doğruluk Oranı				
	Döngü Sayısı	KouBM-DFD (Model-1(b))	KouBM-DFD (Model-2)	KouBM-DFD (Model-3)	KouBM-DFD (Model-4)
Eğitim doğruluğu	100	%97,38	%96,00	%95,87	%94,62
	200	%98,22	%96,99	%96,79	%95,64
	300	%98,78	%97,37	%97,19	%95,96
	400	%99,05	%97,59	%97,41	%96,16
	500	%99,35	%97,81	%97,63	%96,35
Test doğruluğu	100	%97,22	%96,26	%96,04	%95,21
	200	%98,25	%97,12	%96,49	%96,02
	300	%98,70	%97,66	%97,24	%96,53
	400	%99,00	%97,85	%97,67	%96,71
	500	%99,23	%98,07	%97,53	%96,87

Tablo 4.26’da Üç Çıkışlı Çok Görevli ConNN için zaman ve doğruluk oranı karşılaştırması verilmiştir.

Tablo 4.26. Üç Çıkışlı Çok Görevli ConNN ağ modelleri için zaman ve doğruluk oranı karşılaştırması

Üç Çıkışlı Çok Görevli ConNN (KouBM-DFD)	Döngü sayısı=1	Döngü sayısı=500	Döngü sayısı=500
	Zaman	Zaman	Doğruluk oranı
Model-1(b)	8 dk 42 sn	3 gün 30 dk	%99,23
Model-2	7 dk 56 sn	2 gün 18 sa 06 dk 40 sn	%98,07
Model-3	7 dk 20 sn	2 gün 12 sa 58 dk 20 sn	%97,53
Model-4	7 dk 7 sn	2 gün 11 sa 01 dk 40 sn	%96,87

Üç Çıkışlı Çok Görevli ConNN eğitimlerinin sonuçları karşılaştırıldığında konvolüsyon katmanlarındaki adım sayılarının ya da havuzlama katmanındaki kaydırma adımlarının değişiminin ağırlık çalışması için gereken süre ve elde edilen doğruluk oranına etkisinin ters orantılı olduğu görülmüştür.

Katmanlarda bulunan filtreleri kaydırmak için gerekli adım sayısı artarken eğitim için gereken toplam sürenin de kısaldığı ancak doğruluk oranının da azaldığı tespit edilmiştir.

4.3.5. Senaryo 5: DBN modeli ile yorgunluk tahmin sistemi

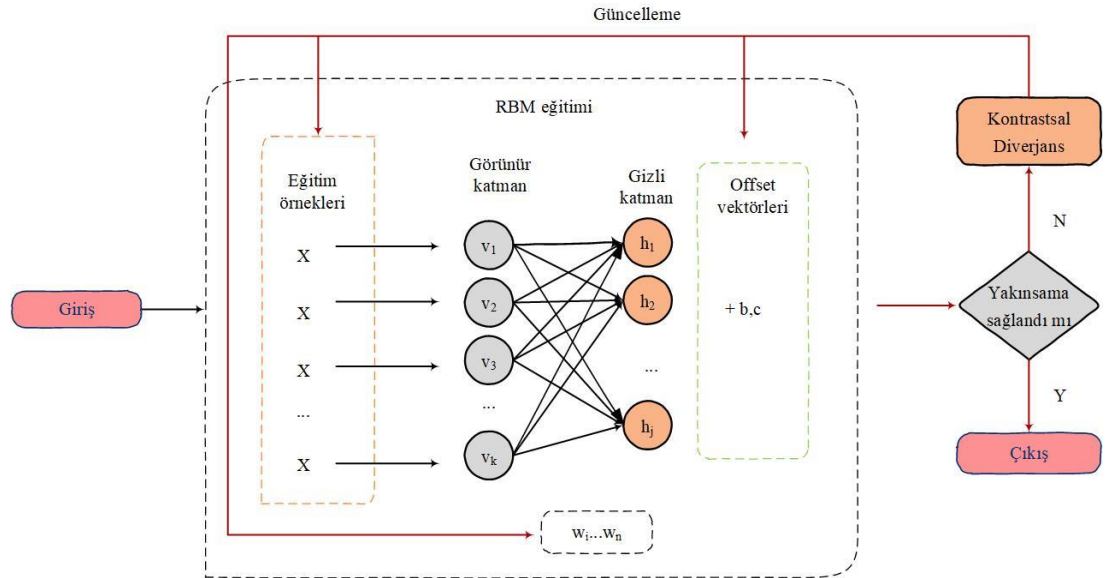
Tez çalışmasında eğitim sürecinde, veri kümeleri eğitim kümesine ve test kümesine ayrılmıştır. Önerilen DBN modelinde kullanılan YawDD, Nthu-DDD ve KouBM-DFD veri kümelerinin dağılımı Tablo 4.27’de gösterilmiştir.

Tablo 4.27. DBN modelinde kullanılan veri kümelerinin dağılımı

Veri kümesi	YawDD	Nthu-DDD	KouBM-DFD
Eğitim	20480	20480	20480
Test	4096	4096	4096
Toplam	24576	24576	24576

Veri kümelerindeki görüntüler göz eğitimi ve ağız eğitimi için ikiye ayrılmaktadır. DBN modelinde kullanılmak için YawDD, Nthu-DDD ve KouBM-DFD veri kümeleri, ağ içindeki hesaplama karmaşıklığını azaltmak için gri seviyeye çevrilmiş ve görüntüler 640×480 çözünürlük boyutundan 160×120 boyutuna yeniden boyutlandırılmıştır. Bu sebeple bu verileri işleyen sinir ağı görünür katman 19200 giriş düğümüne sahiptir. Kısaca görünür RBM katmanı, 120×160 piksel = 19200 piksel ve bir sapmaya sahip olan veri vektörüdür.

Önerilen DBN modelinde ilk olarak denetimsiz öğrenme modeli kullanılarak, her bir RBM'deki iki katman arasındaki koşullu olasılığa bağlı olarak düğümler arasındaki ağırlıkların değerleri için ön tanımlama yapılmıştır. İkinci olarak ise ön tanımlaması RBM tarafından yapılan sınıflandırma parametrelerinin denetimli öğrenme dâhil edilerek ağırlıkların ince ayar optimizasyonu yapılmaktadır. Önerilen sistemdeki RBM eğitim algoritması Şekil 4.91'de verilmiştir.



Şekil 4.91. RBM eğitim algoritması

RBM eğitimi için önce parametreler rastgele başlatılmıştır. W ağırlık matrisi ve b, c görünür ve gizli katmandaki birim sayılarıdır. Model verisi ve gerçek veri vektörü

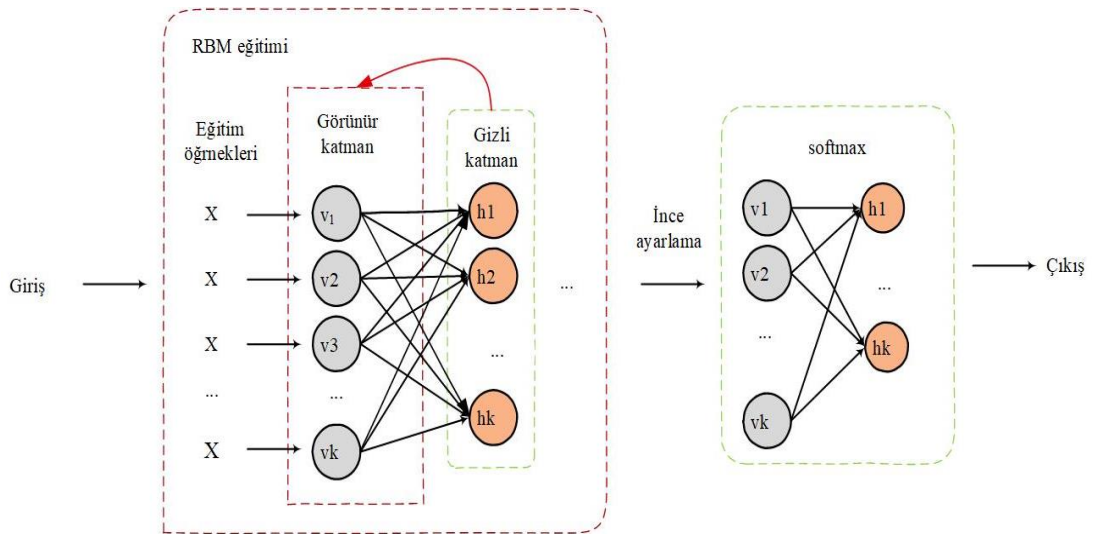
tarafından üretilen örnekler arasındaki hatanın azaltılması için, görünür katmanların olasılığının türevi alınarak ağırlıklar güncellenmektedir. Bu güncellenme, pozitif faz katkısı ile negatif faz katkısı arasındaki farkı verir.

Tablo 4.28’de RBM gizli katman sayısı 2 iken ilk gizli katman üzerinde farklı düğümlerin ters/negatif faz hatası (KouBM-DFD) örneği gösterilmiştir. Burada RBM sayısı 2 olduğunda ters/negatif faz hatasının farklı düğümler üzerindeki etkisi gösterilmiştir. Ters/negatif faz hatasının düşük olduğu durumlar dikkate alınarak önerilen modelde optimum birim sayısı belirlenmiştir.

Tablo 4.28. İlk gizli katman üzerinde farklı düğümlerin ters/negatif faz hatası (KouBm_DFD)

RBM Sayısı	Gizli katmandaki birim sayısı	İlk gizli katman üzerinde farklı düğümlerin ters/negatif faz hatası
2	32	0,1233
	64	0,0702
	128	0,4381
	256	0,4330
	500	0,3159

Daha sonra DBN'nin eğitimine geçilmiştir. Tez çalışmasında, göz/ağız durumları için farklı 2 durum ele alınmıştır. Bu nedenle son katman 2 birimden oluşmaktadır. Son katmanda her birimin en yüksek olasılığını hesaplamak için ince tonlama algoritması uygulanmıştır. Böylece ağırlıklar güncellenmiş ve sınıflandırma iyileştirilmiştir. Şekil 4.92’de önerilen DBN modeli eğitimi verilmiştir.



Şekil 4.92. DBN modeli eğitimi

Tez çalışmasında, modelin en iyi performans ile çalışması için modelde kaç gizli katmanın bulunması gerektiğine dair kesin bir cevap yoktur. Bu durum genelde veri kümelerinin türlerine ve yapılarına bağlıdır. Az sayıda gizli katmana sahip model kısa bir sürede eğitilebilir ancak bu model eğitim veri kümelerinin tüm özelliklerini tam olarak depolayamadığından düşük performansa neden olabilir. Fazla sayıda gizli katman kullanılması ise aşırı öğrenmeye neden olabilir. Bu nedenle, en iyi durumu incelemek için farklı katman sayılarında ve bu katmanlarda bulunan farklı düğüm sayıları için farklı 28 test yapılmıştır. Göz durum bilgisine dayanan farklı RBM katmanlarından elde edilen RBM eğitim doğruluk oranları Tablo 4.29’da gösterilmiştir.

Tablo 4.29. Göz durum bilgisine dayanan farklı RBM katmanlarından elde edilen RBM eğitim doğruluk oranı

RBM		Eğitim Doğruluk Oranı			Döngü sayısı
Katman sayısı	Düğüm sayısı	KouBM-DFD	YawDD	Nthu-DDD	R/I
1	32	%92,18	%86,10	%48,56	10/100
1	64	%92,28	%84,34	%48,76	10/100
1	128	%92,63	%86,52	%48,76	10/100
1	256	%92,44	%87,14	%49,22	10/100
1	500	%92,02	%85,45	%48,93	10/100
2	32×32	%92,90	%85,61	%49,37	10/100
2	64×64	%93,60	%87,19	%50,14	10/100
2	128×128	%92,90	%85,38	%48,47	10/100
2	256×128	%92,84	%85,71	%49,28	10/100
2	500×500	%92,94	%86,46	%49,48	10/100
3	32×32×32	%92,19	%86,30	%48,93	10/100
3	64×64×64	%92,58	%86,13	%48,96	10/100
3	128×128×128	%91,80	%85,16	%48,14	10/100
3	128×64×64	%92,09	%85,74	%49,54	10/100
3	128×64×32	%92,71	%86,23	%48,99	10/100
3	256×128×64	%92,01	%86,59	%49,61	10/100
3	256×128×32	%92,80	%86,65	%49,19	10/100
3	1000×500×250	%92,40	%85,61	%51,07	10/100
4	32×32×32×32	%92,29	%87,57	%49,58	10/100
4	64×64×64×64	%92,90	%86,13	%48,37	10/100
4	128×128×64×64	%92,58	%85,81	%49,38	10/100
4	256×128×64×64	%92,38	%85,03	%48,27	10/100
4	500×500×250×100	%92,68	%85,77	%48,86	10/100
5	32×32×32×32×32	%92,32	%85,97	%49,54	10/100
5	64×64×64×64×64	%92,40	%85,84	%49,38	10/100
5	128×128×64×64×64	%92,71	%85,77	%49,28	10/100
5	256×128×64×64×64	%92,42	%85,61	%49,28	10/100
5	500×500×250×100×100	%92,51	%85,84	%49,22	10/100

KouBM-DFD veri kümesi üzerinde RBM Eğitim Doğruluk Oranı yaklaşık %92 olarak elde edilmiştir. Bu durumda; KouBm-DFD veri kümesi üzerinde elde edilen sonuçlar diğer veri kümelerinden elde edilen sonuçlardan daha iyi sonuç vermektedir. Bu veri

kümesinde elde edilen en yüksek doğruluk oranı ise gizli katman sayısı 2 olduğunda ve bu katmandaki düğüm sayısı 64×64 olduğunda %93,60 olarak elde edilmiştir. YawDD veri kümesi üzerinde RBM eğitim doğruluk oranı yaklaşık %86 olarak elde edilmiştir. Bu veri kümesinde elde edilen en yüksek doğruluk oranları ise gizli katman sayısı 2 olduğunda ve düğüm sayısı 64×64 olduğunda %87,19 olarak ve gizli katman sayısı 4 olduğunda ve düğüm sayısı 32×32×32×32 olduğunda %87,57 olarak elde edilmiştir. Nthu-DDD veri kümesi üzerinde RBM eğitim doğruluk oranı yaklaşık %86 olarak elde edilmiştir. Bu veri kümesinde elde edilen en yüksek doğruluk oranları ise gizli katman sayısı 2 olduğunda ve düğüm sayısı 64×64 olduğunda %51,14 olarak ve gizli katman sayısı 3 olduğunda ve düğüm sayısı 1000×500×250 olduğunda %51,07 olarak elde edilmiştir. Göz durum bilgisine dayanan farklı RBM katmanlarından elde edilen DBN Test Doğruluk Oranı Tablo 4.30'da gösterilmiştir.

Tablo 4.30. Göz durum bilgisine dayanan farklı RBM katmanlarından elde edilen DBN test doğruluk oranı

RBM		Test Doğruluk Oranı			
Katman sayısı	Düğüm sayısı	KouBM-DFD	YawDD	Nthu-DDD	Döngü sayısı
1	32	%82,67	%85,22	%71,91	100
1	64	%83,94	%84,41	%70,38	100
1	128	%83,45	%85,90	%69,96	100
1	256	%84,44	%86,17	%69,80	100
1	500	%83,72	%86,26	%72,50	100
2	32×32	%82,94	%86,75	%71,03	100
2	64×64	%85,30	%87,63	%71,18	100
2	128×128	%83,53	%86,68	%72,80	100
2	256×128	%84,54	%85,56	%71,13	100
2	500×500	%85,02	%86,77	%71,22	100
3	32×32×32	%82,85	%85,61	%70,04	100
3	64×64×64	%83,53	%86,17	%69,69	100
3	128×128×128	%83,69	%85,51	%69,63	100
3	128×64×64	%83,17	%85,77	%70,96	100
3	128×64×32	%84,11	%85,51	%69,50	100
3	256×128×64	%82,19	%85,94	%69,11	100
3	256×128×32	%83,63	%86,04	%70,21	100
3	1000×500×250	%83,50	%84,60	%68,46	100
4	32×32×32×32	%84,54	%84,93	%68,95	100
4	64×64×64×64	%83,17	%85,35	%70,44	100
4	128×128×64×64	%83,59	%86,98	%71,71	100
4	256×128×64×64	%82,94	%84,51	%70,67	100
4	500×500×250×100	%83,59	%85,64	%71,45	100
5	32×32×32×32×32	%84,15	%85,94	%70,51	100
5	64×64×64×64×64	%83,69	%84,83	%71,06	100
5	128×128×64×64×64	%83,76	%86,17	%70,02	100
5	256×128×64×64×64	%84,86	%86,43	%70,51	100
5	500×500×250×100×100	%84,21	%85,68	%70,64	100

Göz durum bilgileri dikkate alınarak KouBM-DFD veri kümesi üzerinde farklı RBM katmanları için DBN test doğruluk oranı yaklaşık %84 olarak elde edilmiştir. Bu veri kümesinde en yüksek doğruluk oranı RBM sayısı 2 olduğunda elde edilmiştir. YawDD veri kümesi üzerinde farklı RBM katmanları için DBN test doğruluk oranı yaklaşık %86 olarak elde edilmiştir. Bu veri kümesinde en yüksek doğruluk oranları ise RBM sayısı 2 olduğunda elde edilmiştir. Nthu-DDD veri kümesi üzerinde farklı RBM katmanları için DBN test doğruluk oranı yaklaşık %71 olarak elde edilmiştir. Bu veri kümesinde en yüksek doğruluk oranı ise RBM sayısı 2 olduğunda elde edilmiştir. Sadece ağız durum bilgileri kullanılarak farklı RBM katmanlarından elde edilen RBM eğitim doğruluk oranı sonuçları Tablo 4.31’de gösterilmiştir.

Tablo 4.31. Ağız durum bilgisine dayanan farklı RBM katmanlarından elde edilen RBM eğitim doğruluk oranı

RBM		Eğitim Doğruluk Oranı			Döngü sayısı
Katman sayısı	Düğüm sayısı	KouBM-DFD	YawDD	Nthu-DDD	R/I
1	32	%65,62	%91,24	%76,60	10/100
1	64	%66,46	%92,25	%76,79	10/100
1	128	%65,67	%91,18	%76,43	10/100
1	256	%66,02	%91,67	%76,30	10/100
1	500	%66,30	%91,28	%76,63	10/100
2	32×32	%65,28	%91,67	%77,76	10/100
2	64×64	%67,43	%91,73	%77,12	10/100
2	128×128	%67,77	%92,76	%86,79	10/100
2	256×128	%66,46	%91,83	%77,63	10/100
2	500×500	%66,37	%91,96	%77,57	10/100
3	32×32×32	%65,97	%92,29	%76,75	10/100
3	64×64×64	%65,19	%91,37	%76,69	10/100
3	128×128×128	%66,94	%92,12	%77,57	10/100
3	128×64×64	%65,77	%91,47	%76,95	10/100
3	128×64×32	%67,58	%91,08	%77,44	10/100
3	256×128×64	%66,21	%90,98	%76,26	10/100
3	256×128×32	%66,11	%91,34	%76,86	10/100
3	1000×500×250	%67,38	%91,11	%77,96	10/100
4	32×32×32×32	%65,95	%91,83	%77,08	10/100
4	64×64×64×64	%66,31	%91,31	%76,27	10/100
4	128×128×64×64	%66,60	%91,53	%76,95	10/100
4	256×128×64×64	%65,72	%91,80	%76,79	10/100
4	500×500×250×100	%66,28	%91,57	%76,66	10/100
5	32×32×32×32×32	%66,99	%91,15	%76,89	10/100
5	64×64×64×64×64	%66,60	%90,79	%86,00	10/100
5	128×128×64×64×64	%66,54	%91,50	%78,03	10/100
5	256×128×64×64×64	%63,93	%91,80	%76,69	10/100
5	500×500×250×100×100	%66,15	%90,92	%77,51	10/100

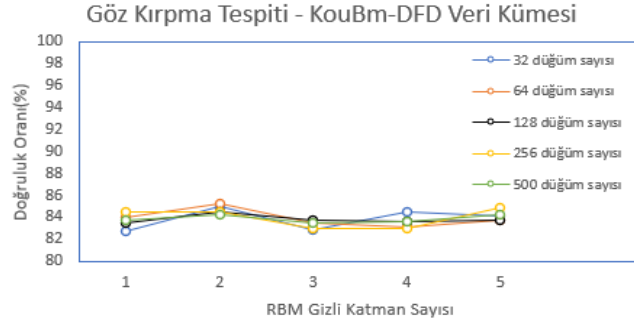
KouBM-DFD veri kümesi üzerinde RBM eğitim doğruluk oranı yaklaşık %66 olarak elde edilmiştir. Bu veri kümesinde gizli katman sayısı 2 olduğunda ve düğüm sayısı

128×128 olduğunda en yüksek doğruluk oranı %67,77 olarak elde edilmiştir. YawDD veri kümesi üzerinde RBM eğitim doğruluk oranı yaklaşık %91 olarak elde edilmiştir. Bu durumda; YawDD veri kümesi üzerinde elde edilen sonuçlar diğer veri kümelerinden elde edilen sonuçlardan daha iyi sonuç vermektedir. Bu veri kümesinde gizli katman sayısı 2 olduğunda ve düğüm sayısı 128×128 olduğunda en yüksek doğruluk oranı %92,76 olarak elde edilmiştir. Nthu-DDD veri kümesi üzerinde RBM eğitim doğruluk oranı yaklaşık %77 olarak elde edilmiştir. Bu veri kümesinde gizli katman sayısı 2 olduğunda ve düğüm sayısı 128×128 olduğunda en yüksek doğruluk oranı %86,79 olarak ve gizli katman sayısı 3 olduğunda ve düğüm sayısı 1000×500×250 olduğunda %86,79 olarak elde edilmiştir. Ağız durum bilgisine dayanan farklı RBM katmanlarından elde edilen DBN test doğruluk oranı Tablo 4.32’de gösterilmiştir.

Tablo 4.32. Ağız durum bilgisine dayanan farklı RBM katmanlarından elde edilen DBN test doğruluk oranı

RBM		Test Doğruluk Oranı			Döngü sayısı
Katman sayısı	Düğüm sayısı	KouBM-DFD	YawDD	Nthu-DDD	
1	32	%79,10	%86,10	%87,14	100
1	64	%81,05	%86,72	%87,01	100
1	128	%78,76	%86,07	%86,39	100
1	256	%80,96	%86,65	%86,59	100
1	500	%80,73	%86,69	%86,13	100
2	32×32	%79,98	%87,37	%87,12	100
2	64×64	%82,37	%87,98	%87,79	100
2	128×128	%79,59	%87,58	%88,41	100
2	256×128	%79,39	%86,87	%87,17	100
2	500×500	%80,60	%87,88	%88,18	100
3	32×32×32	%77,54	%87,34	%86,88	100
3	64×64×64	%79,93	%86,56	%86,95	100
3	128×128×128	%78,22	%87,40	%86,72	100
3	128×64×64	%80,03	%86,26	%86,10	100
3	128×64×32	%79,93	%87,11	%86,69	100
3	256×128×64	%80,76	%85,90	%87,08	100
3	256×128×32	%80,81	%86,91	%87,17	100
3	1000×500×250	%78,97	%86,59	%86,39	100
4	32×32×32×32	%80,57	%86,46	%87,66	100
4	64×64×64×64	%79,69	%86,10	%86,95	100
4	128×128×64×64	%80,14	%87,04	%85,58	100
4	256×128×64×64	%79,85	%86,52	%86,43	100
4	500×500×250×100	%80,14	%87,34	%87,57	100
5	32×32×32×32×32	%79,52	%86,82	%87,01	100
5	64×64×64×64×64	%80,79	%86,04	%76,79	100
5	128×128×64×64×64	%79,04	%87,27	%86,75	100
5	256×128×64×64×64	%80,05	%85,84	%86,56	100
5	500×500×250×100×100	%79,65	%85,55	%86,46	100

Ağız durum bilgisine dayanan KouBM-DFD veri kümesi üzerinde farklı RBM katmanları için DBN test doğruluk oranı yaklaşık %80 olarak elde edilmiştir. Bu veri kümesinde en yüksek doğruluk oranları RBM sayısı 2 olduğunda elde edilmiştir. YawDD veri kümesi üzerinde farklı RBM katmanları için DBN test doğruluk oranı yaklaşık %87 olarak elde edilmiştir. Bu veri kümesinde de en yüksek doğruluk oranları RBM sayısı 2 olduğunda elde edilmiştir. Nthu-DDD veri kümesi üzerinde farklı RBM katmanları için DBN test doğruluk oranı yaklaşık %87 olarak elde edilmiştir. Bu veri kümesinde de en yüksek doğruluk oranları RBM sayısı 2 olduğunda elde edilmiştir. Tablo 4.30’da verilen KouBM-DFD veri kümesi üzerinde göz durum bilgisine dayanan farklı sayıda RBM gizli katmanlarından elde edilen test doğruluk oranları Şekil 4.93’de gösterilmiştir.

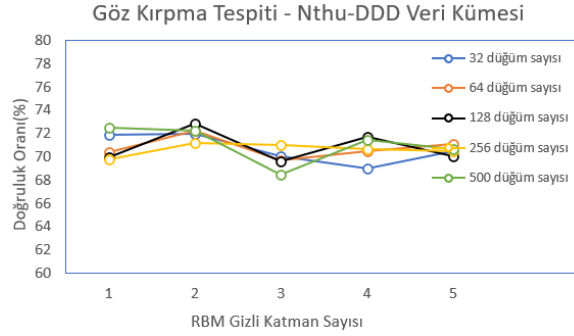


Şekil 4.93. Test doğruluk oranları (KouBM-DFD)

RBM gizli katman sayısı 2 olduğunda eğitim ve test doğruluğunun optimum sonuç verdiği Şekil 4.93’te görülmektedir. RBM gizli katman sayısı 2’den az ya da 2’den fazla olduğunda, eğitim ve test doğruluğunda düşüş gözlenmektedir. KouBM_DFD veri kümesi üzerinde farklı RBM katmanlarından elde edilen DBN test doğruluk oranı yaklaşık %84 olarak elde edilmiştir. En optimum düğüm sayısı 64 olarak tespit edilmiştir.

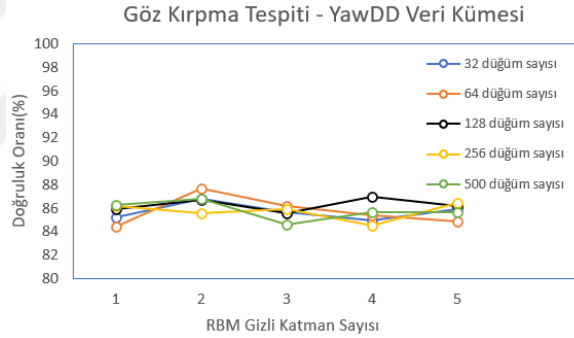
Tablo 4.30’da verilen Nthu-DDD veri kümesi üzerinde göz durum bilgisine dayanan farklı sayıda RBM gizli katmanlarından elde edilen test doğruluk oranları Şekil 4.94’te gösterilmiştir. RBM gizli katman sayısı 2 olduğunda eğitim ve test doğruluğunun daha iyi olduğu Şekil 4.94’te görülmektedir. RBM gizli katman sayısı 2’den az ya da 2’den fazla olduğunda, eğitim ve test doğruluğunda düşüş gözlenmektedir. Nthu-DDD veri kümesi üzerinde farklı RBM katmanlarından elde edilen DBN test doğruluk oranı

yaklaşık %71 olarak elde edilmiştir. En optimum düğüm sayısı 128 olarak tespit edilmiştir.



Şekil 4.94. Test doğruluk oranları (Nthu-DDD)

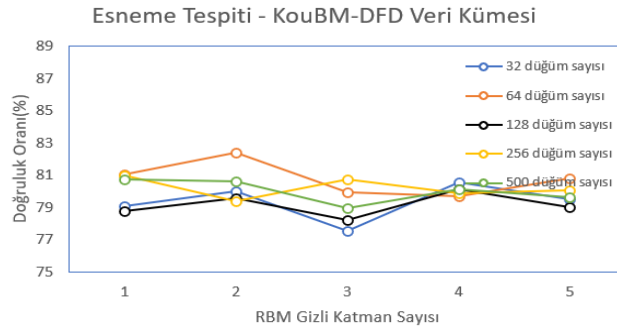
Tablo 4.30’da verilen YawDD veri kümesi üzerinde göz durum bilgisine dayanan farklı sayıda RBM gizli katmanlarından elde edilen test doğruluk oranları Şekil 4.95’te gösterilmiştir.



Şekil 4.95. Test doğruluk oranları (YawDD)

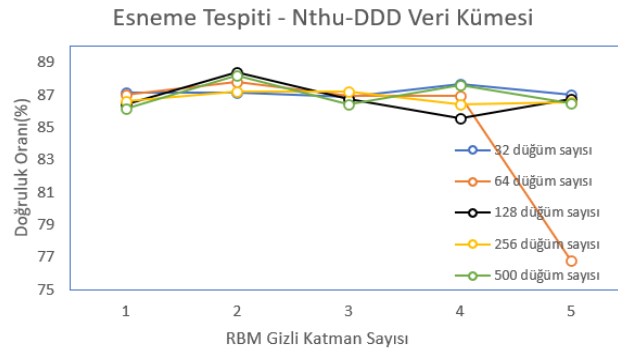
RBM gizli katman sayısı 2 olduğunda eğitim ve test doğruluğunun daha iyi olduğu Şekil 4.95’te görülmektedir. RBM gizli katman sayısı 2’den az ya da 2’den fazla olduğunda, eğitim ve test doğruluğunda düşüş gözlenmektedir. YawDD veri kümesi üzerinde farklı RBM katmanlarından elde edilen DBN test doğruluk oranı yaklaşık %86 olarak elde edilmiştir. En optimum düğüm sayısı 64 olarak tespit edilmiştir.

Tablo 4.32’de verilen KouBM-DFD veri kümesi üzerinde ağız durum bilgisine dayanan farklı sayıda RBM gizli katmanlarından elde edilen test doğruluk oranları Şekil 4.96’da gösterilmiştir.



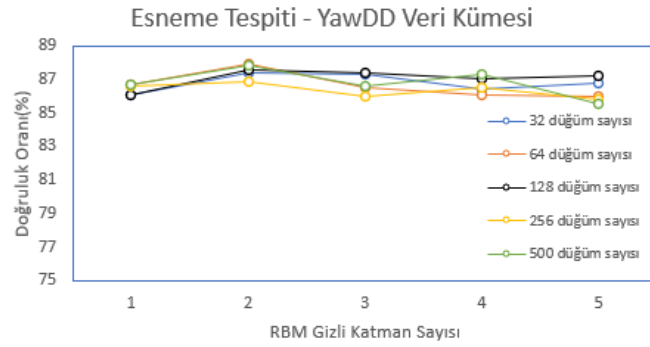
Şekil 4.96. Test doğruluk oranları (KouBM-DFD)

RBM gizli katman sayısı 2 olduğunda eğitim ve test doğruluğunun daha iyi olduğu Şekil 4.96’da görülmektedir. RBM gizli katman sayısı 2’den az ya da 2’den fazla olduğunda, eğitim ve test doğruluğunda düşüş gözlenmektedir. KouBM-DFD veri kümesi üzerinde farklı RBM katmanlarından elde edilen DBN Test Doğruluk Oranı yaklaşık %80 olarak elde edilmiştir. En optimum düğüm sayısı 64 olarak tespit edilmiştir. Tablo 4.32’de verilen Nthu-DDD veri kümesi üzerinde ağız durum bilgisine dayanan farklı sayıda RBM gizli katmanlarından elde edilen test doğruluk oranları Şekil 4.97’de gösterilmiştir.



Şekil 4.97. Test doğruluk oranları (Nthu-DDD)

RBM gizli katman sayısı 2 olduğunda eğitim ve test doğruluğunun daha iyi olduğu Şekil 4.97’de görülmektedir. RBM gizli katman sayısı 2’den az ya da 2’den fazla olduğunda, eğitim ve test doğruluğunda düşüş gözlenmektedir. Nthu-DDD veri kümesi üzerinde farklı RBM katmanlarından elde edilen DBN test doğruluk oranı yaklaşık %87 olarak elde edilmiştir. Optimum düğüm sayısı 128 olarak tespit edilmiştir. Tablo 4.32’de verilen YawDD veri kümesi üzerinde ağız durum bilgisine dayanan farklı sayıda RBM gizli katmanlarından elde edilen test doğruluk oranları Şekil 4.98’de gösterilmiştir.



Şekil 4.98. Test doğruluk oranları (YawDD)

RBM gizli katman sayısı 2 olduğunda eğitim ve test doğruluğunun daha iyi olduğu Şekil 4.112’de görülmektedir. RBM gizli katman sayısı 2’den az ya da 2’den fazla olduğunda eğitim ve test doğruluğunda düşüş gözlenmektedir. YawDD veri kümesi üzerinde farklı RBM katmanlarından elde edilen DBN test doğruluk oranı yaklaşık %87 olarak elde edilmiştir. En optimum düğüm sayısı 64 olarak tespit edilmiştir. Tüm deneysel sonuçlar değerlendirildiğinde, eğitim ve test doğruluğundaki en iyi sonuçların RBM gizli katman sayısı 2 ve toplam katman sayısı 4 olduğunda elde edildiği açıktır. Gizli katman sayısı 2’den az ya da 2’den fazla olduğunda, eğitim ve test doğruluğunda düşüş gözlenmektedir. Bu nedenle, DBN sürücü yorgunluk algılama modelinin optimum gizli katman sayısı 2 olarak belirlenmiştir. Önerilen yöntemin doğruluk oranlarının (yaklaşık olarak oranları) literatürde bulunan diğer yaklaşımlarla karşılaştırılması Tablo 4.39’da verilmektedir.

Tablo 4.33. Önerilen yöntemin doğruluk oranlarının literatürde bulunan diğer yaklaşımlarla karşılaştırılması

Literatür	Yıl	Yöntem	Veri Kümeleri	Doğruluk Oranı (%)		
Zhang ve diğ., 2015	2015	CNN	Kendi oluşturdukları veri kümesi, YawDD	%88		
Weng ve diğ., 2016	2016	HTDBN	Nthu-DDD , YawDD	%84		
Zhao ve diğ., 2017	2017	DBN	Kendi oluşturdukları veri kümesi	%96		
Jabbar ve diğ., 2018	2018	DBN	Nthu-DDD	%81		
Gu ve diğ., 2018	2018	Hierarchical CNN	EMD, ZJU, CEW	%96		
Ma ve diğ., 2018	2018	DBN	Kendi oluşturdukları veri kümesi	%88		
Zheng ve diğ., 2019	2019	DBN-BPNN	Beijing Jiaotong üni. Lokomotif driver	%83		
Önerilen yöntem	2020	DBN(göz)	KouBM-DFD Nthu-DDD, YawDD	%84	%71	%86
	2020	DBN(ağız)	KouBM-DFD, Nthu-DDD, YawDD	%80	%87	%87

Yapılan deneysel çalışmalarda, önerilen DBN yapısının bilgiyi iyi öğrendiğini ve veri kümesi için yüksek tanıma doğruluğunu sağladığını göstermektedir. Tablo 4.39’da verilen sınıflandırma sonuçları incelendiğinde tüm bilgiyi kullanan DBN modelinin de diğer modeller kadar sağlam olduğu görülmektedir. Ayrıca yapılan çalışma tüm veriyi kullandığı için ten rengi ve cinsiyet bilgilerinden etkilenmemektedir. Tanıma sonuçları, Hierarchical CNN ve DBN (Zhao ve diğ., 2017) dışındaki diğer sınıflandırıcılardan daha yüksek tanıma oranına ulaştığını göstermektedir.

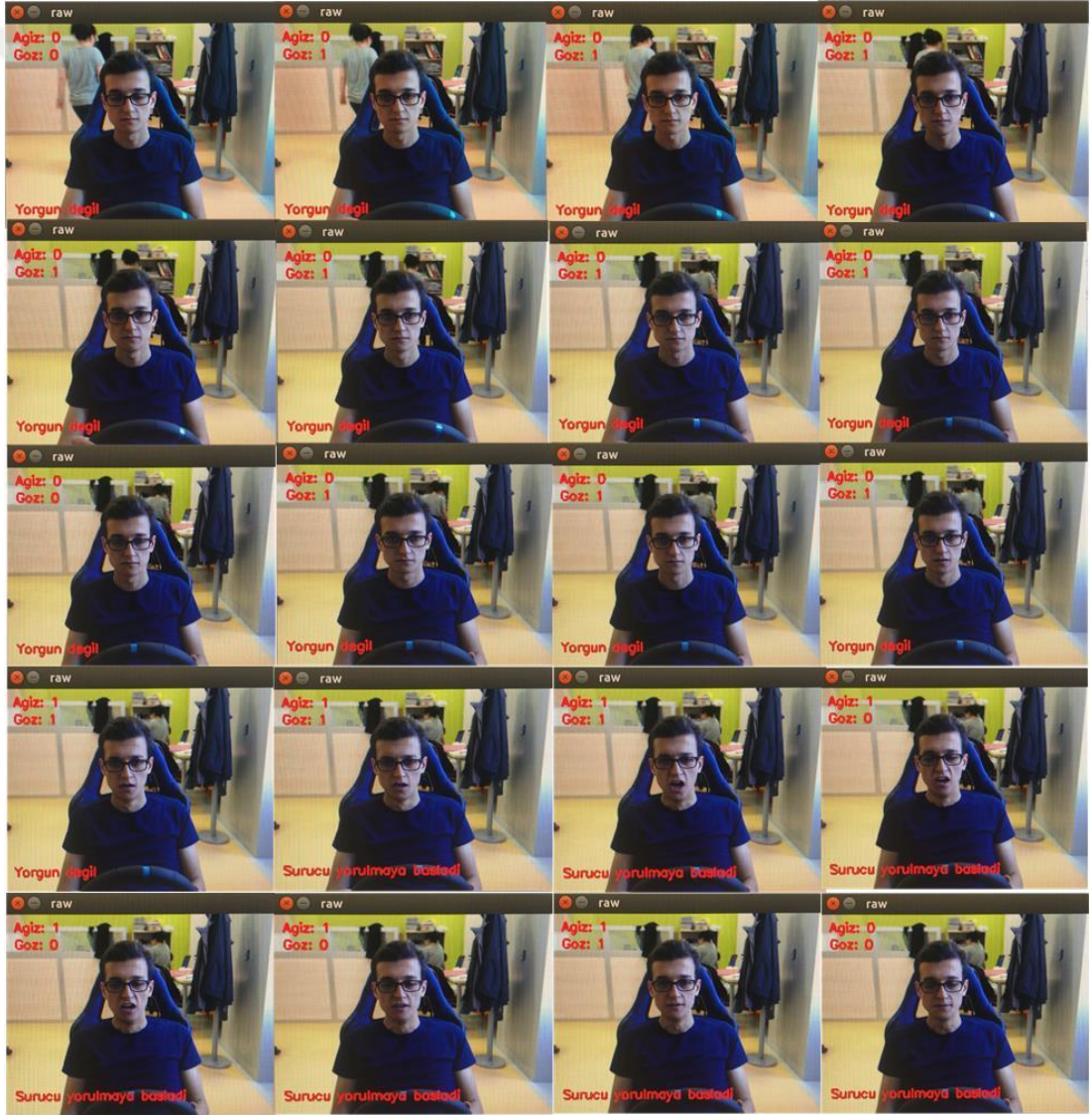
4.3.6. Senaryo 6: Jetson TX2 üzerinde ConNN ve DBN modellerinin test edilmesi

Tez çalışmasında tasarlanan sistemlerin taşınabilir cihazlarda iş verimliliğini test etmek için, gömülü platform olan Jetson TX2’den yararlanılmaktadır. Gömülü sistem aynı anda 1 sn de 30 çerçeve işlemektedir. Bu da tasarladığımız sistemin temelde gerçek zamanlı gereksinimi karşıladığını göstermektedir. Buna ek olarak, paralel algılama ve yorgunluk algılama için harcanan zaman çok az olduğu için sistemin yüksek performansla çalıştığı söylenebilmektedir. Sürücüler yorulduğunda gözlerini kapatma, hızlı bir şekilde kırpma veya esneme gibi bir dizi davranışsal tepkiler vermeye başlamaktadırlar. Bu nedenle tez kapsamında elde edilen İki Çıkışlı Çok Görevli ConNN modellerini test etmek için, PERCLOS ve FOM metrikleri hesaplanmakta ve yorgunluk tahmin edilmeye çalışılmaktadır.

Literatüre göre (Zhang ve diğ., 2015; Zhang ve diğ., 2017), sürücü için yorgunluk durumu PERCLOS eşiği $f_{PERCLOS} > 0,24$ durumu kabul edilmiştir. Ayrıca $f_{FOM} > 0.16$ olduğunda esneme semptomlarının yorgunluğu gösterdiği belirtilmiştir. Tez çalışmasının bu aşamasında sürücünün yorgunluk durumu üç seviyeye ayrılmaktadır.

Birinci tip çok yorgun bir seviyedir ve bu değerlendirme iki aşamadan oluşmaktadır. Birincisi, yalnızca göz kapama süresi 2 saniyeyi ($n_{close} > 60$ kare) aştığında gerçekleşmektedir. Bu durumda sistem esnmeden bağımsız olarak sürücünün aşırı yorulduğuna dair uyarı vermektedir. Diğer durum, göz durumunu ve esneme durumunu birlikte kontrol etmektedir. Sürücünün gözlerinin kapanma eşiği ve esneme eşiği $f_{PERCLOS} > 0.24$ ise || $f_{FOM} > 0.16$ ise sürücü aşırı derecede yorgun olarak kabul edilmektedir. İkinci tip yorgunlukta sürücünün gözlerini kapama eşiği ve esneme eşiği

$f_{PERCLOS} = 0.15 \sim 0.24$ || $f_{PERCLOS} > 0.16$ aralığında ise sürücünün yorulmaya başladığı kabul edilmektedir. Üçüncü tip normaldir. Burada sürücü yorgunluk belirtisi göstermez ve normal şekilde sürebilir. Bu çalışmada PERCLOS ve FOM eşik değerleri eğitim için kullanılan veri kümelerinden farklı test videoları üzerinde test edilmiştir. Ayrıca yapılan tez çalışmada sürücünün konuşma durumu ile esneme durumunun birbirine karışmadığı gözlemlenmektedir. Şekil 4.99'da İki Çıkışlı Çok Görevli ConNN (KouBM-DFD) modelinin testi için yorgunluğa geçiş evresindeki bir sürücüye ait rastgele seçilen 5 dk'lık videonun peşpeşe gelen 20 çerçevesi gösterilmiştir.



Şekil 4.99. İki Çıkışlı Çok Görevli ConNN (KouBM-DFD) modelinin testi için, yorgunluğa geçiş evresindeki bir sürücüye ait rastgele seçilen 5 dk'lık videoda peşpeşe gelen 20 çerçevenin gösterimi

Şekil 4.99’da rastgele seçilen bir video içerisinde peşpeşe gelen çerçevelerde sürücünün göz ve ağız bilgilerinin değişimi verilmiştir. Sol üst köşede bulunan durumlarda sürücünün ağız ve göz bilgileri belirtilmiştir. En altta kırmızı ile yazılı olan durum etiketi ise sürücünün yorgun olup olmadığı bilgisini tutmaktadır.

Şekil 4.100’de İki Çıkışlı Çok Görevli ConNN (Nthu-DDD) modelinin testi için yorgun olmayan bir sürücüye ait rastgele seçilen 5 dk’lık videoda peşpeşe gelen 20 çerçevesi gösterilmiştir.



Şekil 4.100. İki Çıkışlı Çok Görevli ConNN (Nthu-DDD) modelinin testi için yorgun olmayan bir sürücüye ait rastgele seçilen 5 dk’lık videoda peşpeşe gelen 20 çerçevenin gösterimi

Şekil 4.100’de rastgele seçilen bir video içerisinde peşpeşe gelen çerçevelerde sürücünün göz ve ağız bilgilerinin değişimi verilmiştir. Sol üst köşede bulunan

durumlarda sürücünün ağız ve göz bilgileri belirtilmiştir. En altta kırmızı ile yazılı olan durum etiketi ise sürücünün yorgun olup olmadığı bilgisini tutmaktadır.

Test aşamasında Üç Çıkışlı ConNN bilgilerine ek olarak burada başın durumu da sürücü yorgunluğu bilgilerine dahil edilmektedir. Bu aşamada sürücünün yorgunluk durumu üç seviyeye ayrılmaktadır.

Birinci tip çok yorgun bir seviyedir ve bu değerlendirme üç aşamadan oluşmaktadır. Birincisi, yalnızca kafa 3 sn'yi ($n_{kafa} > 90$ kare) aştığında sistem diğer koşullardan bağımsız sürücünün aşırı yorulduğuna dair uyarı vermektedir.

İkincisi yalnızca göz kapama süresi 5 saniyeyi ($n_{close} > 150$ kare) aştığında gerçekleşmektedir. Bu durumda sistem yine diğer durumlarda bağımsız olarak sürücünün aşırı yorulduğuna dair uyarı vermektedir. Son durum ise, göz durumunu ve esneme durumunu birlikte kontrol etmektedir. Sürücünün gözlerinin kapanma eşiği ve esneme eşiği $f_{PERCLOS} > 0.24$ ise $\parallel f_{FOM} > 0.16$ ise sürücü aşırı derecede yorgun olarak kabul edilmektedir. İkinci tip yorgunlukta sürücü gözlerinin kapanma eşiği ya da ve esneme eşiği $f_{PERCLOS} = 0.15 \sim 0.24 \parallel f_{PERCLOS} > 0.16$ aralığında ise sürücünün yorulmaya başladığı kabul edilmektedir.

Üçüncü tip normaldir. Burada sürücü yorgunluk belirtisi göstermez ve normal şekilde sürebilir. Bu çalışmada PERCLOS, FOM eşik değerleri ve kafa düşmesi durumları eğitim için kullanılan veri kümelerinden ayrı test videoları üzerinde test edilmiştir.

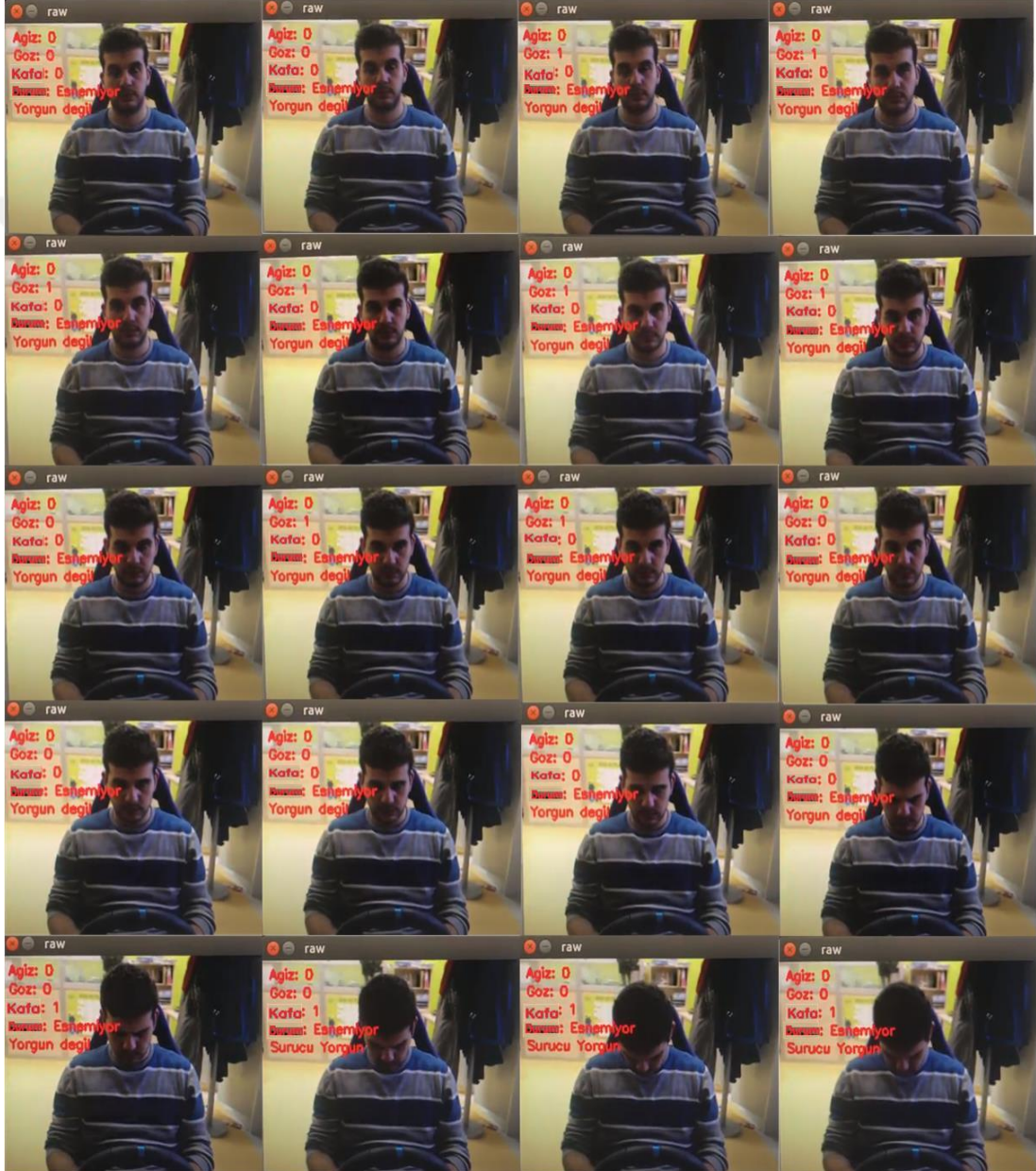
Şekil 4.101'de Üç Çıkışlı Çok Görevli ConNN (Nthu-DDD) modelinin testi için yorgun bir sürücüye ait rastgele seçilen 2 dk'lık videoda peşpeşe gelen 60 çerçeveyi göstermektedir.

Şekil 4.101'de rastgele seçilen bir video içerisinde peşpeşe gelen çerçevelerde sürücünün göz ve ağız bilgilerinin değişimi verilmiştir. Sol üst köşede bulunan durumlarda sürücünün ağız, göz ve baş bilgileri belirtilmiştir. Durum etiketinde ise sürücünün yorgunluğa geçiş evresi ve hangi tip yorgun olduğu bilgisi tutulmaktadır.



Şekil 4.101. Üç Çıkışlı Çok Görevli ConNN (Nthu-DDD) modelinin testi için yorgun bir sürücüye ait 2 dk'lık videoda peşpeşe gelen 60 çerçevenin gösterimi

Şekil 4.102’de Üç Çıkışlı Çok Görevli ConNN (KouBM-DFD) modelinin testi için yorgun bir sürücünün 2 dk’lık videoda peşpeşe gelen 20 çerçeve gösterilmiştir. Şekil 4.102’de rastgele seçilen bir video içerisinde peşpeşe gelen çerçevelerde sürücünün göz ve ağız bilgilerinin değişimi verilmiştir. Sol üst köşede bulunan durumlarda sürücünün ağız, göz ve baş bilgileri belirtilmiştir. Durum etiketinde ise sürücünün yorgun olup olmadığı bilgisi tutulmaktadır.



Şekil 4.102. Üç Çıkışlı Çok Görevli ConNN (KouBM-DFD) modelinin testi için yorgun bir sürücüye ait rastgele seçilen 2 dk’lık videoda peşpeşe gelen 20 çerçevenin gösterimi

Göz bilgileri için oluşturulan DBN modelinin Jetson TX2 üzerinde test etmek için, PERCLOS metriği hesaplanmakta ve yorgunluk tahmin edilmeye çalışılmaktadır. Tez çalışmasının bu aşamasında sürücünün yorgunluk durumu iki seviyeye ayrılmaktadır. Birinci tip yorgun bir seviyedir ve bu değerlendirme iki aşamadan oluşmaktadır. Birincisi, yalnızca göz kapama süresi 5 saniyeyi ($n_{close} > 150$ kare) aştığında gerçekleşmektedir. Bu durumda sistem sürücünün aşırı yorulduğuna dair uyarı vermektedir. Diğer durum sürücünün gözlerinin kapanma süresi $f_{PERCLOS} > 0.24$ ise ise sürücü yorgun olarak kabul edilmektedir. İkinci tip yorgunlukta sürücünün normal yani yorgun olmadığı durumdur. Burada sürücü yorgunluk belirtisi göstermez ve normal şekilde sürebilir.

Şekil 4.103'te göz bilgileri için oluşturulan DBN (ZJU) modelinin testi için yorgun olmayan bir sürücüye ait rastgele seçilen 4 sn'lik videoda peşpeşe gelen 10 çerçeve gösterilmiştir.



Şekil 4.103. Göz bilgileri için oluşturulan DBN (ZJU) modelinin testi için yorgun olmayan bir sürücüye ait rastgele seçilen 4 sn'lik videoda peşpeşe gelen 10 çerçevenin gösterimi

Şekil 4.104'de Gerçek sürüş sahnesinde performans gösterimi verilmiştir.



Şekil 4.104. Gerçek sürüş sahnesinde performans gösterimi

5. SONUÇLAR VE ÖNERİLER

Tez çalışmasında, geçici/sürekli fiziksel özelliklerden yorgunluk/uykululuk halinin sürücü davranışları üzerine etkisi araştırılmış ve araştırma sonucunda yorgun ve/veya uykusuz sürücüler tespit edilmiştir. Bu amaçla, tezde sürücü yorgunluğunun değerlendirilmesi sırasında, sürücü yorgunluk derecelerinin belirlenmesi için, birden fazla özgün ve yeni ağ modeli (Senaryo 4 içerisinde yer alan Model-1, Model-2, Model-3 ve Model-4) önerilmiştir. Oluşturulan sınıflandırma modelleri için; Adaboost, SVM, DBN, Tek Görevli ve Çok Görevli çalışan farklı iki ConNN'den (İki Çıkışlı ConNN ve Üç Çıkışlı ConNN) yararlanılmıştır.

Sürücü yorgunluğu/uykululuğu tespiti literatürde yer alan önemli ve çözülmesi zor problemlerden biridir. Çünkü bu problem insana bağlıdır ve her insanın yorgunluk karşısında tepkisi değişkendir. Günümüzde sürücü yorgunluğu tespit sistemleri hala güvenilirliği, doğruluğu ve duyarlılığı konusunda belirsizliğini korumaktadır. Bu nedenle, son yıllarda insanların hareketlerinin dikkate alındığı davranış temeli ölçüme (bilgisayar görsü ve görüntü işleme) olan ilgi artmıştır. Bilgisayarlı görme ve görüntü işleme yöntemleri, sürücü yorgunluklarını tespit etmek için tercih edilmektedir. Bu tür yaklaşımlar, sürücü yorgunluğunu anlamak için sürücünün yüz ifadelerine odaklanır. Davranışsal tabanlı yorgunluk tespit yöntemlerinin en büyük avantajlarından birisi de sürücünün dikkatini dağıtmadan uygulanabilmesidir.

Bu tez çalışmasının temel amacı, sürücüye hemen müdahale etme durumu gerektiren veya gerektirmeyen sistemler için, gerçek zamanlı sistemlerle uyumlu, test aşaması eğitim aşaması kadar yüksek başarımla gerçekleştirilebilecek, özgün bir sürücü yorgunluk tespit ve tahmin sistemi tasarlamaktır. Tez çalışması kapsamında, literatürde yer alan diğer yaklaşımlar denenmiş ve tespit edilen açık problemlere çözüm amaçlı Çok Görevli ConNN modeli (İki çıkışlı Çok Görevli ConNN ve Üç çıkışlı Çok Görevli ConNN) önerilmiştir.

Bunun yanı sıra veri kümesi yetersizliği problemine çözüm odaklı, Kocaeli Üniversitesi Bilgisayar Mühendisliği Görüntü İşleme ve Bilgisayar Görüsü Araştırma

Laboratuvarında simülasyon ortamında, farklı etnik gruplara ait gönüllü sürücülerden yorgun ve yorgun olmadığı durumları içeren KouBM-DFD olarak adlandırılan özgün bir veri kümesi oluşturulmuştur. KouBM-DFD veri kümesinin yanı sıra, ZJU, YawDD ve Nthu-DDD veri kümeleri üzerinde tez çalışmasında önerilen özgün modeller test edilmiştir. Tez çalışması kapsamında oluşturulan senaryolardan elde edilen doğruluk oranları Tablo 5.1’de verilmiştir.

Tablo 5.1. Önerilen özgün farklı 5 senaryodan elde edilen doğruluk oranları

	Yöntemler	Model	ZJU	YawDD	Nthu- DDD	KouBM- DFD
Senaryo 1	Adaboost	-	-	-	-	%70
Senaryo 2	SVM	-	-	-	-	%97
Senaryo 3	Tek Görevli ConNN(göz)	Model-1(a)*	%98,75	%97,73	%98,55	%98,81
	Tek Görevli ConNN(ağız)	Model-1(a)	-	%99,35	%99,25	%99,01
Senaryo 4	İki Çıkışlı Çok Görevli ConNN	Model-1(b)**	-	%99,18	%98,93	%98,88
		Model-2	-	%96,94	%97,97	%97,24
		Model-3	-	%98,11	%98,01	%97,26
		Model-4	-	%96,55	%96,65	%94,92
	Üç Çıkışlı Çok Görevli ConNN	Model-1(b)	-	-	%99,27	%99,23
		Model-2	-	-	%98,42	%98,07
		Model-3	-	-	%98,78	%97,53
		Model-4	-	-	%97,14	%96,87
Senaryo 5	DBN	-göz	-	%86	%71	%84
		-ağız	-	%87	%87	%80

* Modelde konvolüsyon katmanında seçilen filtreler sırasıyla conv1(3), conv2(4), conv3(5).

**Modelde konvolüsyon katmanında seçilen filtreler sırasıyla conv1(6), conv2(8), conv3(10).

Tablo 5.1’de verilen sonuçlar incelendiğinde, Senaryo 1’de Adaboost algoritması için %70, Senaryo 2’de SVM algoritması için %97 doğruluk oranında sürücü yorgunluğu tespit edilmiştir. Senaryo 3’te, ConNN modeli ile göz bilgilerine ve ağız durum bilgilerine dayanarak farklı 2 sınıflandırma yapılmıştır. Bu senaryoda konvolüsyon katmanlarında bulunan filtre sayısı ve adım sayısı hiper parametreleri değiştirilerek optimum parametre değerlerinin bulunması için farklı modeller (Model-1(a), Model-1(b) ve Model-1(c)) (Bkz. Tablo 4.6) uygulanmıştır. Sadece göz durum bilgilerine dayanan Model-1(a) için; ZJU, YawDD, Nthu-DDD ve KouBM-DFD veri kümeleri için sırasıyla %98,75, %97,73, %98,55 ve %98,81 en başarılı doğruluk oranları elde edilmiştir. Sadece ağız durum bilgilerine dayanan Model-1(a) da ise; YawDD, Nthu-DDD ve KouBM-DFD veri kümeleri için sırasıyla %97,56, %97,83 ve %99,01 en başarılı doğruluk oranlarıyla sürücü yorgunluğu tespit edilmiştir.

Senaryo 3 dikkate alındığında, tek görevli ConNN modellerinden biri kullanılırsa sistem sadece göz ya da sadece ağız tespiti yapabilir. Bu durumda göz ve ağız aynı anda tespit edilemez. Bu nedenle sürücü yorgunluğu tespiti için çoklu durumlarda (ağız ve göz durumlarının ikisinin aynı anda tespit edilmesi, ağız, göz ve baş durumlarının üçünün aynı anda tespit edilmesi) çalışacak Senaryo 4'te verilen İki Çıkışlı Çok Görevli ConNN ve Üç Çıkışlı Çok Görevli ConNN ağ modelleri tasarlanmıştır. Her bir ağ modelinde optimum sonucun bulunması için, ağ içinde yer alan hiper parametrelerin değişiminden Model-1 (Model-1(a), Model-1(b) ve Model-1(c)), Model-2, Model-3 ve Model-4 ağ modelleri oluşturulmuştur. İki Çıkışlı ConNN ağ modelinde göz ve ağız durum bilgileri giriş bilgisi olarak kullanılmış ve sürücünün davranışlarının sürücünün yorgunluğuna etkisi gözlemlenmiştir. En iyi doğruluk oranı Model-1(b)'de elde edilirken en hızlı çalışan ise Model-3 ağ modeli olmuştur (Bkz. Tablo 4.20). İki çıkışlı Çok Görevli ConNN Model-1(b)'de konvolüsyon katmanında sırasıyla 6,8 ve 10 adet filtre kullanılmış ve adım sayısı 1 olarak belirlenmiştir. Havuzlama katmanında ise giriş görüntüsüne 2×2'lik filtre ile 2 adım kaydırmalı maksimum havuzlama yöntemi kullanılmıştır. YawDD, Nthu-DDD, KouBM-DFD veri kümeleri için sırasıyla, %99,18, %98,93, %98,88 doğruluk oranları elde edilmiştir. Senaryo-4'te verilen Üç Çıkışlı ConNN ağ modelinde göz, ağız ve kafa düşme bilgileri giriş bilgisi olarak kullanılmış ve sürücünün davranışlarının sürücünün yorgunluğuna etkisi gözlemlenmiştir. En iyi sonuç Model-1(b)'de elde edilmiş, en hızlı çalışan ise Model-3 ağ modeli olmuştur (Bkz. Tablo 4.26). Üç çıkışlı Çok Görevli ConNN Senaryo-4 Model-1(b)'de konvolüsyon katmanında sırasıyla 6,8 ve 10 adet filtre kullanılmış ve adım sayısı 1 olarak belirlenmiştir. Havuzlama katmanında ise giriş görüntüsüne 2×2'lik filtre ile 2 adım kaydırmalı maksimum havuzlama yöntemi kullanılmıştır. Nthu-DDD, KouBM-DFD veri kümeleri üzerinde sırasıyla %99,27, ve %99,23 doğruluk oranları elde edilmiştir. Bu da tez çalışmasında önerilen yeni ve özgün Senaryo 4 modelinin başarılı ve istikrarlı olduğunu kanıtlar.

Senaryo 5'te, sadece göz ve sadece ağız durum bilgileri dikkate alınarak farklı 2 sürücü yorgunluğu tespiti çalışması yapılmış ve sınıflandırma için DBN modeli kullanılmıştır. Göz durum bilgileri kullanılarak, KouBM-DFD, Nthu-DDD ve YawDD veri kümeleri için sırasıyla %84, %71 ve %86 doğruluk oranı elde edilmiştir. Sadece ağız durum

bilgileri kullanılarak, KouBM-DFD, Nthu-DDD ve YawDD veri kümeleri için sırasıyla %80, %87 ve %87 doğruluk oranlarıyla sürücü yorgunluğu tespit edilmiştir.

Yukarıda verilen tüm senaryolara ek olarak tez çalışmasında modellerin gerçek zamanlı sistemlere uygulanabilirliğinin gösterilmesi amacıyla, Çok Görevli ConNN mimarileri (İki Çıkışlı ConNN ve Üç Çıkışlı ConNN) ve DBN mimarileri Jetson TX2 modülü üzerinde uygulanmıştır. Elde edilen sonuçlar değerlendirildiğinde modelin gömülü uygulaması yüksek başarı oranıyla gerçekleşmiştir (Bkz. bölüm 4.3.6.).

Bu tez çalışmasında, yapılan tüm çalışmalar ve sonuçları ayrıntılı olarak verilmiştir. Elde edilen sonuçlar değerlendirildiğinde, tez çalışması sonunda elde edilen tüm çıktılar özgündür. Tez kapsamında sürücü yorgunluk/uykululuk tespiti için göz, ağız ve kafa durumlarını aynı anda tespit edebilen Çok Görevli ConNN modeli ve KouBM-DFD veri kümesi literatüre önemli katkılar sağlayacaktır.

İleriki çalışmalar değerlendirildiğinde, bu tezde önerilen Çok Görevli ConNN modeli kullanılarak özel tipteki problemlere has yeni özgün model yapıları oluşturulacaktır. Ayrıca gece koşulları da dikkate alınarak, yeni sürücü yorgunluğu tespit modelleri üzerine çalışmalar yapılması planlanmaktadır. Tez kapsamında oluşturulan KouBM-DFD veri kümesi içinde gece koşullarında çekilmiş görüntüler de yer almaktadır, ancak gece koşullarında sürücü yorgunluğu/uykululuğu tespiti problemi ileriki çalışma olarak bırakılmıştır. Bunun yanı sıra, modellerin gerçek zamanlı uygulamaları için farklı gömülü sistemlerin (Jetson TX2/Jetson Nano ve NVIDIA Tam Yapay Zeka Kiti vb.) kullanılması hedeflenmektedir.

KAYNAKLAR

Abathi S., Omidyeganeh M., Shirmohammadi S., Hariri B., YawDD: a yawning detection dataset, *ACM MultimeDBN Systems Conference*, Singapore, 19-21 Mart 2014.

Abtahi S., Hariri B., Shirmohammadi S., Driver Drowsiness Monitoring Based on Yawning Detection, *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Hangzhou, Çin, 10-12 Mayıs 2011.

Akrout B., & Mahdi W., Yawning detection by the analysis of variational descriptor for monitoring driver drowsiness. In *2016 International Image Processing, Applications and Systems (IPAS)*, 5-7 Kasım 2016, Hammamet, Tunisia.

Ali M. J., Sarkar S., Kumar G. P., Cabibihan J. J., A non intrusive human fatigue monitoring system, *International Journal of Future Computer and Communication*, 2012, 1(3), 284.

Ali M., Abdullah S., Raizal C. S., Rohith K. F., Menon V. G., A Novel and Efficient Real Time Driver Fatigue and Yawn Detection-Alert System, In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, Hindistan. 23-25, Nisan 2019.

Alioua N., Amine A., Rogozan A., Bensrhair A., Rziza M., Driver head pose estimation using efficient descriptor fusion, *Eurasip J. Image Video Process*, 2016, 2(2016), 1-14.

Artaud P., Planque S., Lavergne C., Cara H., Lepine P., Tarriere C., Gueguen B. An on-board system for detecting lapses of alertness in car driving, *The Fourteenth International Technical Conference on the Enhanced Safety of Vehicles*, Munich, Almanya, 23-26 Mayıs 2014.

Baş F. İ., Geçici Fiziksel Özelliklerden Yorgunluk ve Uykusuzluğun Sürücü Davranışları Üzerine Etkisi, Yüksek Lisans Tezi, Atatürk Üniversitesi, Fen Bilimleri Enstitüsü, Erzurum, 2015, 380502.

Borghini G., Gasparini R., Vezzani R., Cucchiara R., Embedded recurrent network for head pose estimation in car, In *2017 IEEE Intelligent Vehicles Symposium (IV)*, CA, ABD, 11-14 Haziran 2017.

Bosch, "CES® 2017: Bosch is showing these smart solutions in Las Vegas- Bosch MeDBN Service, <http://www.bosch-presse.de/pressportal/de/en/ces-2017-bosch-is-showing-these-smart-solutions-in-las-vegas-81792.html> (Ziyaret tarihi: 16 Kasım 2017).

Braun A. D., Investigation Of Deep Neural Network Image Processing For Cubesat Size Satellites, Yüksek Lisans Tezi, Morehead State University, Kentucky, ABD, 2018.

Brown, I., 1994. Driver fatigue, *Human Factors*, 1994, **36**(2), 298-314.

Buduma, N., Locascio, N., Fundamentals of deep learning: Designing next-generation machine intelligence algorithms. *O'Reilly Media, Inc.*, First Edition, 2017.

Cech, J., Soukupova, T., Real-time eye blink detection using facial landmarks. *Cent. Mach. Perception, Dep. Cybern. Fac. Electr. Eng. Czech Tech. Univ. Prague, Slovenia*, 2016, 1-8.

Chau M., Betke M., Real time eye tracking and blink detection with usb cameras, *ETRA '12: Proceedings of the Symposium on Eye Tracking Research and Applications*, 2005-12, 173–176, 2012.

Ciresan D. C., Meier U., Gambardella L. M., Schmidhuber J., Convolutional Neural Network Committees for Handwritten Character Classification, *2011 International Conference on Document Analysis and Recognition*, 18-21 Eylül 2011.

Colic A., Marques O., B. Furht, *Driver Drowsiness Detection Systems and Solutions*, 2014th Edition, Springer, 2014.

Danisman T., Bilasco I. M., Djeraba C., Ihaddadene N., Drowsy driver detection system using eye blink patterns, *2010 International Conference on Machine and Web Intelligence*, Algiers, Cezayir, 3-5 Ekim 2010.

Darshana S., Fernando D., Jayawardena S., Wickramanayake S., De Silva C., Efficient PERCLOS and gaze measurement methodologies to estimate driver attention in real time, *2014 5th International Conference on Intelligent Systems, Modelling and Simulation*, Langkawi, Malezya, 27-29 Ocak 2014.

Daza I. G., Drowsiness monitoring based on driver and driving data fusion, 2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC), Washington, DC, ABD, 5-7 Ekim 2011.

Dlib, C, C++ Library, <http://dlib.net/> (Ziyaret tarihi: 25 Eylül 2019).

Dwivedi K., Biswaranjan K., Sethi A., Drowsy driver detection using representation learning, *2014 IEEE International Advance Computing Conference (IACC)*, Gurgaon, Hindistan, 21-22 Şubat 2014.

Eriksson M., Papanikotopoulos N. P., Eye-tracking for detection of driver fatigue, *Conference on Intelligent Transportation Systems*, Boston, Amerika, 9-12 Kasım 1997.

Facts T.S., National Highway Traffic Safety Administration, <http://www.americaninDBNn.net/sleepstats.html> (Ziyaret tarihi: 25 Eylül 2019).

- Fan X., Yin B., Sun Y., Nonintrusive driver fatigue detection, *2008 IEEE International Conference on Networking, Sensing and Control*, Sanya, Çin, 6-8 Nisan 2008.
- Feng R., Zhang G., Cheng B., An on-board system for detecting driver drowsiness based on multi-sensor data fusion using dempster-shafer theory, *2009 International Conference on Networking, Sensing and Control*, Okayama, 26-29 Mart 2009.
- Freund, Y., Schapire, R. E., Experiments with a New Boosting Algorithm, *Machine learning: proceedings of the Thirteenth International Conference (ICML '96)*, Bari, İtalya, 3-6 Temmuz 1996.
- Friedman J., Hastie T., Tibshirani R., Additive logistic regression: a statistical view of boosting, *Annals of Statistics*, 2000, **28**(2), 337-407.
- Frossard, D., Kee, E., & Urtasun, R., DeepSignals: Predicting Intent of Drivers Through Visual Signals. In *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, Kanada, 20-24 Mayıs 2019.
- Geng L., Hu Z., Xiao Z., Real-time fatigue driving recognition system based on deep learning and embedded platform, *American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)*, 2019, **53**(1), 164-175.
- Glorot, X., Bordes, A., Bengio, Y., Deep sparse rectifier neural networks, *Conference: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, Lauderdale, ABD, 11 - 13 Nisan 2011.
- Gökmen M., Kurt B., Kahraman F., Çapar A., Çok Amaçlı Gürbüz Yüz Tanıma, *Tübitak*, 104e121, 12-143, 2007.
- Grandjean E., Fatigue in industry, *Br. J. Internal Med*, 1979, **36**(3), 175–186.
- Grandjean E., *Fitting the Task to the Man*, 4th ed., Taylor and Francis, London, 1969.
- Gu W. H., Zhu Y., Chen X. D., He L. F., Zheng B. B., Hierarchical CNN-based real-time fatigue detection system by visual-based technologies using MSP model, *IET Image Processing*, 2018, **12**(12), 2319-2329.
- Gupta R., Aman K., Shiva N., & Singh Y., An improved fatigue detection system based on behavioral characteristics of driver. In *2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE)*, Singapore, 1-3 Eylül 2017.
- Hammoud R., Zhang H., Alertometer: Detecting and mitigating driver drowsiness and fatigue using an integrated human factors and computer vision approach. In *Passive Eye Monitoring*, Signals and Communication Technology, pages 301–321. Springer Berlin Heidelberg, 2008.
- Hamouda G., Saccomanno F. F., Neural network model for truck driver fatigue accident detection, *CanadBNn Conference on Electrical and Computer Engineering*, Montréal, Kanada, 5-8 Eylül 1995.

Han W., Yang Y., Bin Huang G., Sourina O., Klanner F., Denk C., Driver Drowsiness Detection Based on Novel Eye Openness Recognition Method and Unsupervised Feature Learning, *2015 IEEE International Conference on Systems, Man, and Cybernetics*, Hong Kong, 9-12 Ekim 2015.

Hartley L., Horberry T., Mabbott N., Krueger G P., Review of fatigue detection and prediction Technologies, *Institute for Research in Safety and Transport*, 1-28, 2000.

Hinton G. E., Salakhutdinov R. R., Reducing the Dimensionality of Data with Neural Networks, *Science*, 2006, **313**(5786), 504–507.

Hinton, G. E., A practical guide to training restricted Boltzmann machines. In *Neural networks: Tricks of the trade*, Springer, Berlin, Heidelberg, 2012, 599-619

Hinton, G. E., Training products of experts by minimizing contrastive divergence. *Neural computation*, 2002, **14**(8), 1771-1800.

Hong T., Qin H., Drivers drowsiness detection in embedded system. In *2007 IEEE International Conference on Vehicular Electronics and Safety*, Beijing, Çin, 13-15 Aralık 2007.

Horng W. B., Chen C. Y., A real-time driver fatigue detection system based on eye tracking and dynamic template matching, *Journal of Applied Science and Engineering*, 2008, **11**(1), 65-72.

Houssaini A. S., Sabri M. A., Qjidaa H., & Aarab A., Real-Time Driver's Hypovigilance Detection using Facial Landmarks. In *2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS)*, 3-4 Nisan 2019, Fez, Morocco.

Hsu, C. W., Chang, C. C., Lin, C. J., A practical guide to support vector classification. Technical Report, Department of Computer Science and Information Engineering, National Taiwan University, 2003, 1396-1400.

Hsu, H. P. (1995). *Schaum's outlines of theory and problems of signals and systems*. McGraw-Hill.

Hua Y., Guo J., Zhao H., Deep belief networks and deep learning, In *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things*, Harbin, Çin, 17-18 Ocak 2015.

Huang C., Ai H., Li Y., Lao S., High-performance rotation invariant multiview face detection, *IEEE Transactions on pattern analysis and machine intelligence*, 2007, **29**(4), 671-686.

Huang R., Wang Y., Guo L., P-FDCN based eye state analysis for fatigue detection, *IEEE 18th International Conference on Communication Technology (ICCT)*, Chongqing, Çin, 8-11 Ekim 2018.

Iizuka H. Yanagishima-T. Kataoka Y. Seno T. Yabuta, K., “The development of drowsiness warning devices”, In Proceedings 10th International Technical Conference on Experimental Safety Vehicles, (1985).

Ingre M., Åkerstedt T., Peters B., Anund A., Kecklund G., Subjective sleepiness, simulated driving performance and blink duration: examining individual differences, *Journal of Sleep Research*, 2006, **15**(1), 1365-2869.

International Federation of Societies for Electroencephalography and Clinical Neurophysiology, Recommendations for the practice of clinical neurophysiology, 1st ed., Elsevier Amsterdam, New York, 1983.

Jabbar R., Al-Khalifa K., Kharbeche M., Alhajyaseen W., Jafari M., Jiang S., Real-time driver drowsiness detection for android application using deep neural networks techniques, *ProceDBN computer science*, 2018, **130**(2018), 400-407.

Ji Q., Yang X., Real-time eye, gaze, and face pose tracking for monitoring driver vigilance, *Real-time imaging*, 2002, **8**(5), 357-377.

Ji Q., Zhu Z., Lan P., Real-time Nonintrusive monitoring and prediction of driver fatigue, *IEEE Transactions on Vehicular Technology*, 2004, **53**(4), 1052–1068.

Jo J., Lee S. J., Park K. R., Kim I. J., Kim J., Detecting driver drowsiness using feature-level fusion and user-specific classification, *Expert Systems with Applications*, 2014, **41**(4), 1139–1152.

Junaedi S., Akbar H., Driver Drowsiness Detection Based on Face Feature and PERCLOS, *International Conference on Computation in Science and Engineering*, Bandung, Endonezya, 10-12 Temmuz 2017.

Kang H. B., Various approaches for driver and driving behavior monitoring: A review, *2013 IEEE International Conference on Computer Vision Workshops*, Sidney, Avustralya, 1-8 Aralık 2013.

Karakuş B., Derin öğrenme ve büyük veri yaklaşımları ile metin analizi, Doktora Tezi, Fırat Üniversitesi, Türkiye, 2018.

Kır Savaş B., and Becerikli Y., Real Time Driver Fatigue Detection Based on SVM Algorithm, *6th International Conference on Control Engineering & Information Technology, CEIT*, 25-27 Ekim 2018, İstanbul, 2018.

Kır Savaş B., Becerikli Y., Development of Driver Fatigue Detection System By Using Video Images, *Akıllı Sistemlerde Yenilikler Ve Uygulamaları (ASYU 2017)*, Alanya-Antalya, Türkiye, 5-7 Ekim 2017.

Kır Savaş, B., and Becerikli, Y., Real Time Driver Fatigue Detection System Based on Multi-Task ConNN, *IEEE Access*, 2020, **8**(2020), 12491–12498.

Kim J., Shin H., *Algorithm & SoC Design for Automotive Vision Systems*, 1st ed., Springer Dordrecht, 2014.

Kim, J. W., Classification with deep belief networks. Available on https://www.ki-tu-berlin.de/fileadmin/fg135/publikationen/Hebbo_2013_CDB.pdf, 2013.

Kostinger M., Wohlhart P., Roth P. M., Bischof H., Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization, in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCV)*, Barselona, İspanya, 6-13 Kasım 2011.

Kotsiantis S. B., Supervised Machine Learning: A Review of Classification Techniques, *Informatica*, 2007, **31**(3), 249–268.

Krause J., Gebru T., Deng J., Learning features and parts for fine-grained recognition, *2014 22nd International Conference on Pattern Recognition*, Stockholm, İsveç, 24-28 Ağustos 2018.

Krizhevsky A., Sutskever I., Hinton G.E., Imagenet classification with deep convolutional neural networks, *Advances in Neural Information Processing Systems*, 2017, **60**(6), 84–90.

Lai H., Xiao S., Cui Z., Pan Y., Xu C., Yan S., Deep cascaded regression for face alignment, arXiv preprint arXiv:1510.09083, 1.

Lalonde M., Byrns D., Gagnon L., Teasdale N., Laurendeau D., Real-time eye blink detection with GPU-based SIFT tracking, In *Fourth CanaDBNn Conference on Computer and Robot Vision (CRV'07)*, Quebec, Kanada, 28-30 Mayıs 2007.

Lavergne A., Computer vision system for head movement detection and tracking, Doctoral dissertation, University of British Columbia, ABD, 1999.

Lavergne A., Computer vision system for head movement detection and tracking, Doctoral dissertation, University of British Columbia, ABD, 1999.

Lavergne C., P. De Lepine, P. Artaud, S. Planque, A. Domont, C. Tarriere, C. Arsonneau, X. Yu, A. Nauwink, C. Laugeau, J.M. Alloua, R.Y. Bourdet, J.M. Noyer, S. Ribouchon, and C. Confer, Results of the feasibility study of a system for warning of drowsiness at the steeringwheel based on analysis of driver eyelid movement, *Proceedings Of The Fifteenth International Technical Conference On The Enhanced Safety Of Vehicles*, Melbourne, Avustralya, 13-16 Mayıs 1996.

Le R. N., Bengio Y. Representational power of restricted Boltzmann machines and deep belief networks[J], *Neural Computation*, 2008, **20**(6): 1631-1649.

Le V., Brandt J., Lin Z., Bourdev L., Huang T. S., Interactive facial feature localization, *European Conference on Computer Vision 2012*, Floransa, İtalya, 7-13 Ekim 2012.

Learning, D. (2016). Ian Goodfellow and Yoshua Bengio and Aaron Courville.

LeCun Y., Bengio Y., Hilton G., Deep learning, *Nature*, **521**(2015), 436-444.

Lecun Y., Bottou L., Bengio Y., Gradient-based learning applied to document recognition, *IEEE*, 1998, **86**(11), 2278–2324.

Lee D., Oh S., Heo S., & Hahn, M., Drowsy driving detection based on the driver's head movement using infrared sensors. *In 2008 Second International Symposium on Universal Communication*, Osaka, Japan, 15-16 Aralık 2008.

Li H., Xiaoyu Z., Qisong Z., Design of human eye information detection system, *In 2018 Chinese Control And Decision Conference (CCDC)*, Shenyang, Çin, 9 - 11 Haziran 2018.

Li P., Jiang W., Su F., Single-channel EEG-based mental fatigue detection based on deep belief network, *In 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Florida, ABD, 16-20 Ağustos 2016.

Li Z., Li S. E., Li R., Cheng B., and Shi J., 2017 Online detection of driver fatigue using steering wheel angles for real driving conditions, *Sensors*, 2017, **17** (3), pp. 495.

Lin S. D., Lin J.J., Chung C.-Y., Sleepy Eye's Recognition for Drowsiness Detection, *2013 International Symposium on Biometrics and Security Technologies*, Chengdu, Sichuan Çin, 2-5 Temmuz 2013.

Liu S., Yu L., Hou M., An efficient method for driver fatigue state detection based on deep learning, *In 2019 2nd International Conference on Safety Produce Informatization (IICSPI)*, Chongqing, Çin, 28-30 Kasım 2019.

Liu Y., Zhang H., Liu H J., Driver Fatigue Monitoring Method Based on Eyes State Classification, *Chinese Control and Decision Conference*, Yantai, Shandong, Çin, 2-4 Temmuz 2008.

Luo X., Hu R., Fan T., The Driver Fatigue Monitoring System Based on Face Recognition Technology, *Fourth International Conference on Intelligent Control and Information Processing (ICICIP)*, Beijing, Çin, 9-11 Haziran 2013.

Lyu J., Zhang H., Yuan Z., Joint Shape and Local Appearance Features for Real-Time Driver Drowsiness Detection, *Computer Vision ACCV 2016 Workshops*, Taipei, Tayvan, 20-24 Kasım 2016.

M. Sacco and R. Farrugia, "Driver fatigue monitoring system using support vector machines," *5th Int. Symp. Commun. Control Signal Process.*, pp. 1–5, 2012.

Ma J., Zhang J., Gong Z., Du Y., Study on fatigue driving detection model based on steering operation features and eye movement features, *In 2018 IEEE 4th International Conference on Control Science and Systems Engineering (ICCSSE)*, Wuhan, Çin, 24-26 Ağustos 2018.

Ma M., Xu X., Wu J., Guo M., Design and analyze the structure based on deep belief network for gesture recognition, *In 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI)*, Xiamen, Çin, 29-31 Mart 2018.

Ma X., Chau L., K. Yap, Depth video-based two-stream convolutional neural networks for driver fatigue detection, *2017 International Conference on Orange Technologies (ICOT)*, Singapur, 8-10 Aralık 2017.

Mandal B., Li L., Wang G. S., Lin J., Towards Detection of Bus Driver Fatigue Based on Robust Visual Analysis of Eye State, *IEEE Trans. Intell. Transp. Syst.*, 2017, **18**(3), 545–557.

Manu B. N., Facial features monitoring for real time drowsiness detection, *2016 12th International Conference on Innovations in Information Technology (IIT)*, Al-Ain, Birleşik Arap Emirlikleri 28-30 Kasım 2016.

Mashko A., Review of approaches to the problem of driver fatigue and drowsiness, *2015 Smart Cities Symposium Prague (SCSP)*, Prague, 24-25 Haziran 2015.

Mbouna R. O., Kong S. G., Chun M. G., Visual analysis of eye state and head pose for driver alertness monitoring, *IEEE Trans. Intell. Transp. Syst.*, 2013, **14**(3), 1462–1469.

Mbusa, Attention Assist-Vehicle Safety Video|Mercedes-Benz, <https://www.mbusa.com/mercedes/technology/videos/detail/title-safety/videoId710835ab8d127410VgnVCM100000ccec1e35RCRD> (Ziyaret tarihi: 16 Kasım 2017).

Miah A. A., Ahmad M., Mim K. Z., Drowsiness detection using eyeblink pattern and mean eye landmarks distance, *International Joint Conference on Computational Intelligence (IJCCI 2018)*, Bangladeş, 4-15 Aralık 2018.

Mittal S., A Survey on optimized implementation of deep learning models on the NVIDIA Jetson platform, *Journal of Systems Architecture*, 2019, **97**(2019), 428-442.

Mohamed A. R., Dahl G. E., Hinton G., Acoustic modeling using deep belief networks, *IEEE transactions on audio, speech, and language processing*, 2012, **20**(1), 14-22.

Mohamed, A. R., Dahl, G., Hinton, G., Deep belief networks for phone recognition. In *Nips workshop on deep learning for speech recognition and related applications*, 2009 December, **1**(9), 39.

Nilsson N. J., *Principles of artificial intelligence*, 1st ed., Springer-Verlag Berlin Heidelberg, Berlin, 1982.

Nvidia, Jetson, <https://www.nviDBN.com/tr-tr/autonomous-machines/embedded-systems/jetson-tx2/> (Ziyaret tarihi: 12 Haziran 2020).

Nvidia, DRIVE IX Self-Driving Vehicles-NVIDIA Automotive, <https://www.nviDBN.com/en-us/self-driving-cars/drive-ix/> (Ziyaret tarihi: 16 Kasım 2017).

Oliveira L., Cardoso J. S., Lourenço A., Ahlström C., Driver drowsiness detection using non-intrusive signal acquisition, *2018 7th European Workshop on Visual Information Processing (EUVIP)*, Tampere, Finlandiya, 26-28 Kasım 2018.

Pan G., Sun L., Wu Z., Lao S., Eyeblink-based Anti-spoofing in Face Recognition from a Generic Webcam, *The 11th IEEE International Conference on Computer Vision (ICCV'07)*, Rio de Janeiro, Brazilya, 14-20 Ekim 2007.

Park S., Pan F., Kang S., Yoo C. D., Driver drowsiness detection system based on feature representation learning using various deep networks, *Computer Vision – ACCV 2016 Workshops*, Taipei, Tayvan, 20-24 Kasım 2016.

Pratama B. G., Ardiyanto I., Adji T. B., A review on driver drowsiness based on image, bio-signal, and driver behavior, *2017 3rd International Conference on Science and Technology- Computer (ICST)*, Yogyakarta, Endonezya, 11-12 Temmuz 2017.

Punitha A., Geetha M. K., Sivaprakash A, Driver fatigue monitoring system based on eye state analysis, 2014 International Conference on Circuits, Power and Computing Technologies [ICCPCT-2014], Nagercoil, Hindistan, 20-21 Mart 2014.

Raman K. J., Azman A., Arumugam V., Ibrahim S. Z., Yogarayan S., Abdullah M. F. A. & Sonaimuthu K. (2018). Fatigue monitoring based on yawning and head movement. In 2018 6th International Conference on Information and Communication Technology (ICoICT) (pp. 343-347). IEEE.

Reddy B., Kim Y. H., Yun S., Seo C., Jang, J., Real-time driver drowsiness detection for embedded system using model compression of deep neural networks, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Hawaii, ABD, 21-26 Temmuz 2017.

Reddy V. K., Swathi K. S., Investigation of effectiveness of simple thresholding for accurate yawn detection, *Advances in Computer and Computational Sciences*, Singapore, 2017.

Rende F. Ş., Bütün G., Karahan Ş., Enstitüsü B. T., TÜBİTAK BİLGEM G., Derin Öğrenme Algoritmalarında Model Testleri: Derin Testler, *Conference: 10. Ulusal Yazılım Mühendisliği Sempozyumu*, Çanakkale, 24-26 Ekim 2016.

Rivera M., Monitoring of Micro-sleep and Sleepiness for the Drivers Using EEG Signal, Lisans Tezi, Mälardalen University, Västerås, İsveç, 2013.

Royal D., National survey of distracted and drowsy driving attitudes and behavior: 2002, The Gallup Organization, DOT HS 809 566, 11-56, 2003.

Sahayadhas A., Sundaraj K., Murugappan M., Detecting driver drowsiness based on sensors: A review, *Sensors (Basel)*, 2012, **12**(12), 16937-53.

Salakhutdinov R, Mnih A., Hinton G., Restricted Boltzmann machines for collaborative filtering, *Proceedings of the 24th international conference on Machine learning*, Oregon, ABD, 20-24 Haziran 2007.

Salehian S., Far B., Embedded Real Time Blink Detection System for Driver Fatigue Monitoring, *Conference: The 27th International Conference on Software Engineering and Knowledge Engineering*, Pittsburgh, ABD, 5-7 Temmuz 2017.

Scherer D., Müller A., Behnke S., Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition, *20th International Conference on Artificial Neural Networks (ICANN)*, Thessaloniki, Yunanistan, Eylül 2010.

Schiilkop, P. B., Burgest, C., Vapnik, V., Extracting support data for a given task. *In Proceedings, First International Conference on Knowledge Discovery & Data Mining*. AAAI Press, Menlo Park, CA, 1995 August, 252-257.

Sharan S. S., Viji R., Pradeep R., Sajith V., Driver Fatigue Detection Based On Eye State Recognition Using Convolutional Neural Network, In 2019 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, Hindistan, 17-19 Temmuz 2019.

Sheng W., Ou Y., Tran D., Tadesse E., Liu M., An integrated manual and autonomous driving framework based on driver drowsiness detection, *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Tokyo, Japonya, 3-7 Kasım 2013.

Simonyan, K., Zisserman A., Two-stream convolutional networks for action recognition in videos, *Advances in neural information processing systems 27*, Montreal, Kanada, 8-13 Aralık 2014.

Sönmez A., *Ağır Vasıta Sürücülerinin Çalışma Koşulları ve Trafik Kazaları, Uzun Mesafe Yük ve Yolcu Taşımacılığı Yapan Sürücüler Üzerine Bir Çalışma*, 1. baskı, T.C. Emniyet Genel Müdürlüğü Trafik Hizmetleri Başkanlığı Trafik Araştırma Merkezi Müdürlüğü, Ankara, 1999.

Sparsh Mittala, A Survey on Evaluating and Optimizing Neural Network Applications on NVIDIA's Jetson Platform, *J. Syst. Archit.*, 2019, **97**(2019), 428-442.

Süzen A. A. , Duman B., Şen B., Benchmark Analysis of Jetson TX2, Jetson Nano and Raspberry PI using Deep-CNN, *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, Ankara, 26-27 Haziran 2020.

Talukder A.B.M. N. A., Face characterisation based on Gabor filters, Bayes Rule and AdaBoost, Yüksek Lisans Tezi, Rovira i Virgili University, Tarragona, İspanya, 2011.

Tang, X., Zhou, P., & Wang, P., Real-time image-based driver fatigue detection and monitoring system for monitoring driver vigilance. *In 2016 35th Chinese Control Conference (CCC)*, 27-29 Temmuz 2016, Chengdu, China.

Tayibnapis I. R., Choi M. K., Kwon S., Driver's gaze zone estimation by transfer learning, *2018 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, ABD 12-15 Ocak 2018.

Tefft B.C., Acute Sleep Deprivation and Risk of Motor Vehicle Crash Involvement, <https://aaafoundation.org/wp-content/uploads/2017/12/AcuteSleepDeprivationCrashRisk.pdf> (Ziyaret tarihi: 12 Haziran 2020)

TelematicsNews, "Bosch supplies Drowsiness Detection System for VW Passat, http://telematicsnews.info/2012/03/30/bosch-suppliesdrowsiness-detection-system-for-vw-passat-alltrack_m3301/ (Ziyaret tarihi: 16 Kasım 2017).

Toğaçar, M., Ergen, B., Cömert, Z., Application of breast cancer diagnosis based on a combination of convolutional neural networks, ridge regression and linear discriminant analysis using invasive breast cancer images processed with autoencoders. *Medical hypotheses*, 2020, 135, 109503.

Triyanti, V., Iridiastadi, H., Challenges in detecting drowsiness based on driver's behavior. In *IOP Conference Series: Materials Science and Engineering*, 2017, 277(1), pp. 012042.

URL-1: Insurance Information Institute, Facts and Statistics: Drowsy driving, <https://www.iii.org/fact-statistic/facts-statistics-drowsy-driving> (Ziyaret tarihi: 03 Şubat 2021)

URL-2: Traffic Safety Culture Index, Mart 2018 <https://aaafoundation.org/2017-traffic-safety-culture-index/> (Ziyaret tarihi: 10 Mayıs 2020)

URL-3: Regulatory impact analysis for 2010-2011 Hours of Service Rule. Federal Motor Carrier Safety Administration. Retrieved on Aralık 2011 (Ziyaret tarihi: 13 Haziran 2020).

Didi V. K. and S. B. Jamge, "A Vision Based System for Monitoring the Loss of Attention in Automotive Drivers," vol. 4, no. 7, pp. 2013–2016, 2015.

Vanlaar W., Simpson H., Mayhew D., Robertson R., Fatigued and drowsy driving: a survey of attitudes, opinions and behaviors, *J Safety Res.*, 2008, **39**(3), 303-9.

Viola P., Jones M. J., Robust real-time object detection, ICCV Workshop on Statistical and Computation Theories of Vision, Vancouver, Kanada, 13 Temmuz 2001.

Walch M., Mühl K., Kraus J., Stoll T., Baumann M., & Weber M., From Car-Driver-Handovers to cooperative Interfaces, Editors: Meixner G., & Müller C., *Automotive user interfaces*, 1st ed., Springer International Publishing, 273-294, 2017.

Wathiq O., & Ambudkar B. D., Optimized driver safety through driver fatigue detection methods. In *2017 International Conference on Trends in Electronics and Informatics (ICEI)*, 11-12 Mayıs 2017, Tirunelveli, InDBN.

Weiwei Zhang, Murphey Y. L., Wang T., Xu Q., Driver Yawning Detection based on Deep Convolutional Neural Learning and Robust Nose Tracking, *2015 International Joint Conference on Neural Networks (IJCNN)*, Killarney, İrlanda, 11-16 Temmuz 2015.

Weng C. H., Lai Y., Lai S., Driver Drowsiness Detection via a Hierarchical Temporal Deep Belief Network, In *Asian Conference on Computer Vision Workshop on Driver Drowsiness Detection from Video*, Taipei, Tayvan, 20-24 Kasım 2016.

Wu Y., Lee T., Wu Q., Liu H., An eye state recognition method for drowsiness detection, In *2010 IEEE 71st Vehicular Technology Conference*, Taipei, Tayvan, 16-19 Mayıs 2010.

Wu, Y., Wei, D., & Feng, J., Network Attacks Detection Methods Based on Deep Learning Techniques, *Security and Communication Networks*, **1**(2020), 1-17.

Xie J. F., Xie M., Zhu W., Driver fatigue detection based on head gesture and PERCLOS, *2012 Int. Conf. Wavelet Act. MeDBN Technol. Inf. Process. ICWAMTIP*, Chengdu, Çin, 17-19 Aralık 2012.

Xun Y., Real-time Nonintrusive Detection of Driver Drowsines, *Intelligent Transportation Systems Institute Center for Transportation Studies*, CTS 09-15, 1-19, 2009.

Yan C., Coenen F., Zhang B., Driving posture recognition by convolutional neural networks, *IET Computer Vision*, 2016, **10**(2), 103-114.

Yan C., Jiang H., Zhang B., Coenen F., Recognizing driver inattention by convolutional neural networks, *In 2015 8th International Congress on Image and Signal Processing (CISP)*, Shenyang, Çin, 14-16 Ekim 2015.

Yazdi M. Z. J., & Soryani M., Driver Drowsiness Detection by Yawn Identification Based on Depth Information and Active Contour Model. *In 2019 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICT)*, Kerala, InDBN, 5-6 Temmuz 2019.

Yen J. H., PCI/ISA bus single board computer card/CPU card and backplane using eisa bus connectors and eisa bus slots, 1998, U.S. Patent No. 5,852,725, *U.S. Patent and Trademark Office*.

You Z., Gao Y., Zhang J., Zhang H., Zhou M., Wu C, A study on driver fatigue recognition based on SVM method, *4th International Conference on Transportation Information and Safety (ICTIS)*, Alberta, Kanada, 8-10 Ağustos 2017.

Yuen, K., Martin, S., & Trivedi, M. M. (2016, November). Looking at faces in a vehicle: A deep CNN based approach and evaluation. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)* (pp. 649-654). IEEE.

Zhang C., Lu, X., Huang Z., Xia S., Fu C., A Driver Fatigue Recognition Algorithm Based on Spatio-Temporal Feature Sequence, *In 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Suzhou, Çin, 19-21 Ekim 2019.

Zhang F., Su J., Geng L., Z. Xiao, Driver Fatigue Detection Based on Eye State Recognition, *2017 International Conference on Machine Vision and Information Technology (CMVIT)*, Singapore, 17-19 Şubat 2017.

Zhang N., Paluri M., Ranzato M., Panda: pose aligned networks for deep attribute modeling, *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Ohio, Amerika, 24-27 Haziran 2014.

Zhang N., Zhang, H., & Huang, J. (2019). Driver Fatigue State Detection Based on Facial Key Points. In *2019 6th International Conference on Systems and Informatics (ICSAI)* (pp. 144-149). IEEE.

Zhang W., & Su J., Driver yawning detection based on long short term memory networks. In *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, Honolulu, HI, USA, 08 Şubat 2018.

Zhao L., Wang Z., Wang X., Liu Q., Driver drowsiness detection using facial dynamic fusion information and a DBN, *IET Intelligent Transport Systems*, 2018, **12**(2), 127-133.

Zhao Z., Guo J., Ding E., Zhu Z., Zhao D., Terminal replacement prediction based on deep belief networks, In *2015 International Conference on Network and Information Systems for Computers*, Wuhan, Çin, 23-25 Ocak 2015.

Zhao, Z., Zhou, N., Zhang, L., Yan, H., Xu, Y., Zhang, Z., Driver Fatigue Detection Based on Convolutional Neural Networks Using EM-CNN. *Computational Intelligence and Neuroscience*, 2020.

Zheng Z., Dai S., Liang Y., Xie X., Driver fatigue analysis based on upper body posture and DBN-BPNN model, In *2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chengdu, Çin, 20-22 Aralık 2019.



EKLER

EK-A

KouBM-DFD veri kümesi için farklı durum ve semptomlar için oluşturulan sürücü bilgileri Şekil A.1’de verilmiştir.

SÜRÜCÜ ve VIDEO BİLGİLERİ	
SÜRÜCÜ BİLGİLERİ	
Sürücü id (Video id)	
Tarih	
Sürücü Ad-Soyad	
Yaş	
Yaş	18-25 <input type="checkbox"/> 25-50 <input type="checkbox"/> 50-65 <input type="checkbox"/>
Cinsiyet	Bayan <input type="checkbox"/> Bay <input type="checkbox"/>
Doğum Yeri (Ülke)	
Yorgunluk Özellikleri	
Yorgun	Evet <input type="checkbox"/> Hayır <input type="checkbox"/>
Kaç saattir uyanık	
SÜRÜŞ ŞEKLİ	
Ağız kapalı konuşmadan	Evet <input type="checkbox"/>
Konuşurken	Evet <input type="checkbox"/>
Esnerken	Evet <input type="checkbox"/>
Konuşup-Esnerken	Evet <input type="checkbox"/>
CİNSİYET ÖZELLİKLERİ (BAYAN)	
Gözlüksüz	Evet <input type="checkbox"/>
Gözlüklü	Evet <input type="checkbox"/>
Güneş Gözlüklü	Evet <input type="checkbox"/>
Baş Örtülü	Evet <input type="checkbox"/>
Baş Örtüsüz	Evet <input type="checkbox"/>
CİNSİYET ÖZELLİKLERİ (BAY)	
Gözlüksüz	Evet <input type="checkbox"/>
Gözlüklü	Evet <input type="checkbox"/>
Güneş Gözlüklü	Evet <input type="checkbox"/>
Bıyıklı	Evet <input type="checkbox"/>
Bıyıksız	Evet <input type="checkbox"/>
Sakallı	Evet <input type="checkbox"/>
Sakalsız	Evet <input type="checkbox"/>
Video Özellikleri	
Boyut	
FPS	
Geçen süre	1 DK <input type="checkbox"/> 5 DK <input type="checkbox"/> 30 DK <input type="checkbox"/>
Başlama ve Bitiş saati	
Işıkdandırma Özellikleri	
Gündüz	Evet <input type="checkbox"/>
Akşam üzeri	Evet <input type="checkbox"/>

Şekil A.1. KouBM-DFD veri kümesi için farklı durum ve semptomlar

EK-B

Bölüm 1.5’te kullanılan kütüphaneler kısmında anlatılan Dlib, OpenCV, Cline, Cmake, Tensorflow, Pytorch yazılım teknolojileri ile ilgili bilgiler ayrıntılı bir şekilde verilmiştir.

Dlib

Dlib kütüphanesi (Dlib, 2019) C++ programlama dili ile geliştirilmiş ve 2002 yılından buyana geliştirilmeye devam eden içerisinde makine öğrenimi, derin öğrenme ve bilgisayarlı görü algoritmalarını barındıran açık kaynak kodlu bir kütüphanedir. C++ ve Python uygulama programı arayüzü sayesinde de Python programlama dili ile uygulama geliştirilebilmektedir. Farklı programlama dilleri ve platformlar içinde wrapper’ları geliştirilmiştir. Dlib; OS X, MS Windows, Linux, Solaris, BSD ve HP-UX işletim sistemlerini desteklemektedir, ayrıca Raspberry, Tinkerboard gibi gömülü donanımlar veya Android, IOS mobil platformlar üzerinde çalışabilmektedir.

Dlib kütüphanesinde önceden eğitilmiş yüz işareti detektörü, yüzdeki yüz yapılarıyla eşleşen $68(x,y)$ koordinat çiftlerini yüzün konumunu tahmin etmek için kullanılmaktadır. iBUG 300-W veri kümesi üzerinde yapılan eğitim ve testler sonucunda yüz işareti detektörleri belirlenmiştir. Dlib kütüphanesinin yararlandığı başlıca algortmalar arasında SVM, k-Means (K-ortalama), CNN, YSA, v.b bulunmaktadır.

OpenCV

Intel tarafından geliştirilen OpenCV, örüntü tanıma ve görüntü işleme konularında yaygın olarak kullanılmaktadır (Hong ve diğ., 2007; Raman ve diğ., 2018). OpenCV açık kaynak kodlu bir kütüphanedir ve C / C ++, Java ve Python dillerini desteklemektedir. Aynı zamanda birçok platformda (Windows, Linux... gibi) çalışabilmektedir. Bu kütüphane kullanılarak hareket algılama, kamera kalibrasyonu, yüz tanıma ve takibi gibi pek çok görüntü işleme uygulaması algoritmaları gerçekleştirilebilmektedir. OpenCV kütüphanesi içinde bu kütüphaneyi oluşturan bileşenler bulunmaktadır. Bu bileşenler 5 ana başlıkta incelebilmektedir.

Core bileşeni: OpenCV kütüphanesinin temel fonksiyonları ve matris işlemleri bu bileşende bulunmaktadır. Bunun dışında görüntüler üzerinde çizimler ve işlemler yapabilmek için kullanılabilir yöntemleri bu bileşende bulunmaktadır.

HighGui bileşeni: Resimleri ve görüntüleri ekrana getirme, çerçeveler üzerinde işlemler yapma ve ayrıca grafiksel kullanıcı arabirimleri arasındaki iletişimi içeren yöntemler bu bileşende bulunmaktadır.

Imgproc bileşeni: Görüntüler üzerinde kullanılan filtreleme matrisleri, kenarların tespiti, nesne tespiti, renklerin ve bileşenlerinin yönetimi ve eşik değerlerinin belirlenmesi gibi birçok fonksiyonlar bu bileşende bulunmaktadır.

Imgcodecs bileşeni: Bütün dosyalama işlemleri ve bu dosyalamada görüntü ve video işlemleri gibi birçok işlem bu bileşende bulunmaktadır.

Videoio bileşeni: Harici olarak kullanılan kameralar ve video cihazları gibi harici cihazlara erişme işlemleri bu bileşende bulunmaktadır. Ayrıca görüntülerin yakalanması ve görüntü işlemleri için gerekli metotlar bu bileşende bulunmaktadır.

CLion

CLion, C ve C ++ uygulamaları geliştirmek için oluşturulan bir platformdur. Ayrıca CLion için Windows, macOS ve Linux üzerinde çalışan bir platformlar arası IDE'dir denilebilir. Ek olarak, Objective-C / C ++, HTML (HTML5 dahil), CSS, JavaScript ve XML dillerini tam olarak desteklemektedir.

Cmake

Cmake, ücretsiz bir çapraz platform (birçok platformda çalışma özelliği) ve açık kaynaklı bir yazılım uygulamasıdır. Yazılım oluşturma sürecini yönetmek için derleyiciden bağımsız bir yöntem kullanılır. Birden çok kütüphaneye bağlı olan dizin hiyerarşilerini ve uygulamaları destekler. Apple, Xcode ve Microsoft Visual Studio gibi yerel yapı ortamlarıyla birlikte kullanılır. Kendi sisteminde sadece bir C ++ derleyicisine gereksinim duyan asgari bağımlılıklara sahiptir.

Tensorflow

Vektörler yüksek boyutlu olan verilerin gösterilmesi için üretilen çok boyutlu dizi anlamına gelmektedir. Çoğunlukla derin öğrenme mimarilerinde matris boyutlarının veri kümesinde bulunan farklı özelliklerle değişim göstermesi durumlarında kullanılmaktadır. Yani Tensorflow ismi sinir ağlarının vektörler üzerinde gerçekleştirdiği işlemlerden üretilmiştir. TensorFlow, farklı derin öğrenme modellerini uygulamak için çok sayıda işlemler ve hesaplamalar bulunduran Python dilini kullanan bir kütüphanedir.

Google tarafından uzun yıllardır geliştirilen TensorFlow kodları 2015 yılında herkesin erişimine açılmıştır. TensorFlow açık kaynak kodlu bir derin öğrenme kütüphanesidir. TensorFlow platform farketmeksizin hesaplamaları, bir veya birden fazla CPU ve GPU kullanarak kodlamamıza olanak sağlar. Temelinde Python kullanılarak geliştirilen bu kütüphane, günümüzde Python'un yanısıra Javascript, R, Swift gibi birçok dili desteklemektedir. Bunların dışında web uygulamaları için ya da mobil uygulamalar için geliştirilen Tensorflow paketleri de yazılmıştır.

PyTorch

PyTorch makine öğrenme algoritmaları için destek sunan bilimsel hesaplama yapısını kullanan bir kütüphanedir. LuaJIT ve C/CUDA uygulaması temelini kullanan bir yapıya sahiptir. Derin öğrenme tabanlı araştırmalarda en yaygın kullanılan platformlardan biridir. Güçlü bir GPU desteğiyle vektör hesaplamaları hızlı yapılabilir. PyTorch çekirdek özellikleri aşağıda verilmiştir.

- N-boyutlu dizilerin bellekte tutulma boyutlarının yüksekliği
- Veriler üzerinde temel işlemlerin yapılması (sıralama, kataloglama, kesme, yer değiştirme vb. gibi işlemler)
- Veriler üzerinde doğrusal cebir işlemlerinin hızlı ve etkili bir şekilde yapılması
- Sinir ağı ve enerji tabanlı modellerin tasarlanması
- Kodlama ve veri kullanımı için GPU desteği
- Ios, Android, FPGA ve Jetson gömülü sistemleri üzerinde kod yazılabilmesi

Torch, Lua topluluğunun hazırladıklarının yanısıra; makine öğrenmesi, bilgisayarlı görü, sinyal işleme, paralel işlem, resim, video, ses ve ağ iletişimi gibi çeşitli topluluklar tarafından hazırlanmış paketler ile kullanılabilir.

PyTorch kütüphanesinin seçiminin temel nedenlerinden biri sinir ağ modellerini oluşturabilme yeteneğidir. PyTorch'un içinde kompleks sinir ağı topolojisini uygulamak için çeşitli kütüphaneler bulunur. İsteğe bağlı olarak sinir ağının grafiklerini çizdirmek ve CPU/GPU'lar arasında etkili paralel işlem yapabilmeyi sağlamaktadır.



EK-C

Bölüm 2.1’de ve Bölüm 2.2’de verilen Adaboost ve SVM sınıflandırma yöntemleri kapsamlı olarak aşağıda verilmiştir.

Adaboost Yöntemi

AdaBoost (Uyarlamalı Boosting) yöntemi, Viola ve Jones (Viola ve diğ., 2001) tarafından elde edilen peş peşe zayıf sınıflandırıcıları eğitmek için kullanılmaktadır.

Freund ve Schapire (Freund ve diğ., 1996) tarafından 1995 yılında önerilen bu yöntem, temsil edilen her bir öznitelik üzerinde bir zayıf sınıflandırıcıyı eğittikten sonra seçtiği bazı zayıf sınıflandırıcıların doğrusal birleşimi ile kuvvetli sınıflandırıcıyı oluşturmayı amaçlar. Her turda bir öznitelik üzerinde, yanlış sınıflandırılmış her örneğin ağırlıkları artırılmakta (veya alternatif olarak, doğru sınıflandırılmış her örneğin ağırlıkları azaltılır), böylece yeni sınıflandırıcı bu örneklere daha fazla odaklanmaktadır. Bunun amacı her bir zayıf sınıflandırıcı eğitildikten sonra bu ağırlıklar kullanılarak genel bir hata hesaplaması yapılmasıdır. Böylece sistemin eğitilmesi sağlanır (Friedman ve diğ., 2000).

Destek Vektör Makinaları (SVM)

Destek vektör makineleri (SVM), bir makine öğrenmesi algoritmasıdır. SVM sürücü yorgunluğu/uykululuğu tespiti için en çok kullanılan makine öğrenme algoritmalarından biridir (Diddi ve diğ., 2015; Mbouna ve diğ., 2013; Sacco ve diğ., 2012; Rivera ve diğ., 2013; Alioua ve diğ., 2016).

SVM modelinin amacı, hedef değişkene ait sınıfları birbirinden en uygun şekilde ayıracak hiper düzlemi tespit etmektir. Bir dizi eğitim örneği verildiğinde $(x_1, y_1)(x_2, y_2)...(x_n, y_n)$ eğer pozitif ve negatif örnekleri ayıran bir hiper düzlem varsa, hiper düzlemde bulunan x noktaları, $w \cdot x_i + b = 0$ bakılır. Burada w hiper düzlem normaldir ve b orijinden olan mesafedir. Bir “margin”, yani ayırma hiper düzlemi (karar sınırı) ve örnekler (destek vektörleri) arasındaki mesafe etrafında döner (Kotsiantis ve diğ., 2007; You ve diğ., 2017). Yani ayırıcı bir hiper düzlemin marjı, en yakın pozitif veya negatif örneğe en kısa mesafe olarak tanımlanır. SVM eğitiminde,

destek vektörleri kullanılarak, margin en büyük olacak şekilde ayarlanmaya çalışılır. Böylece ayırma hiper düzleminin her iki tarafında bulunan örnekler arasında mümkün olan en büyük mesafe elde edilmiş olur (Kotsiantis ve diğ., 2007; Jo ve diğ., 2014). Burada belirlenen çekirdek işlevi, farklı veri kümeleri üzerinde yüksek performans elde etmek için kullanılmaktadır. SVM algoritmasının en çok kullanılan çekirdek işlevleri doğrusal, polinom (Cubic) ve RBF'dir. Tablo 2.1'de kullanılan Doğrusal SVM, Polinom (Cubic) SVM ve RBF SVM çekirdek modellerini gösterilmiştir.

Tablo C.1 Doğrusal, polinom ve RBF çekirdek modelleri (Hsu ve diğ, 2003)

Çekirdek tipi	İç çarpım	Açıklama
Doğrusal	$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \cdot \mathbf{x}_j$	\mathbf{x}_i ve \mathbf{x}_j destek vektörleridir. “ T ” transpoz işlemini, \cdot iççarpım işlemini ifade etmektedir.
Polinom	$K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i^T \mathbf{x}_j + r)^p, \gamma > 0$	\mathbf{x}_i ve \mathbf{x}_j destek vektörleridir. γ, r, p çekirdek parametreleridir.
RBF	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \ \mathbf{x}^T - \mathbf{x}^i\ ^2), \gamma > 0$	i destek vektörlerinin sayısıdır. γ , çekirdek parametresi, “ T ” transpoz işlemini ifade etmektedir.

EK-D

Şekil D’de verilen görüntü üzerinde Bölüm 4.3.4.1’te verilen İki Çıkışlı Çok Görevli ConNN Model-1 için konvolüsyon, aktivasyon, havuzlama katman filtre değişimleri ve bu filtrelere bağlı olarak özellik matrislerinin değişimi verilmiştir.



Şekil D.1. Örnek görüntü

Giriş görüntü matris bilgileri (3, 240,320)	
tensor([[[[25., 25., 24., ..., 154., 154., 155.], [31., 31., 31., ..., 154., 154., 155.], [28., 29., 32., ..., 154., 154., 155.], ..., [7., 7., 7., ..., 128., 128., 128.], [7., 7., 7., ..., 128., 128., 128.], [7., 7., 7., ..., 128., 128., 128.]], [[25., 25., 24., ..., 154., 154., 155.], [31., 31., 31., ..., 154., 154., 155.], [28., 29., 32., ..., 154., 154., 155.], ..., [7., 7., 7., ..., 128., 128., 128.], [7., 7., 7., ..., 128., 128., 128.], [7., 7., 7., ..., 128., 128., 128.]], [[25., 25., 24., ..., 154., 154., 155.], [31., 31., 31., ..., 154., 154., 155.], [28., 29., 32., ..., 154., 154., 155.], ..., [7., 7., 7., ..., 128., 128., 128.], [7., 7., 7., ..., 128., 128., 128.], [7., 7., 7., ..., 128., 128., 128.]])])])])	

Konvolüsyon 1. katman maskelerinin matris gösterimi		
Maske	Conv1_kernel : torch.Size([6, 3, 5, 5])	Renk Modeli
Conv(1,1)	tensor([[[[-2.9153e-02, 2.8968e-02, 3.7971e-02, -2.9955e-02, 1.1824e-02], [-3.4370e-02, -3.7345e-02, -3.8477e-02, 5.5241e-03, -1.9571e-02], [7.9875e-03, -3.7816e-02, -2.6429e-03, 1.3940e-02, 3.0520e-02], [2.3575e-02, -5.2586e-03, 2.0210e-02, -1.5240e-02, 4.1551e-02], [1.9832e-02, -1.5046e-02, 4.1453e-02, -2.3533e-03, 3.8019e-02]], [[[-2.9153e-02, 2.8968e-02, 3.7971e-02, -2.9955e-02, 1.1824e-02], [-3.4370e-02, -3.7345e-02, -3.8477e-02, 5.5241e-03, -1.9571e-02], [7.9875e-03, -3.7816e-02, -2.6429e-03, 1.3940e-02, 3.0520e-02], [2.3575e-02, -5.2586e-03, 2.0210e-02, -1.5240e-02, 4.1551e-02], [1.9832e-02, -1.5046e-02, 4.1453e-02, -2.3533e-03, 3.8019e-02]], [[[-2.9153e-02, 2.8968e-02, 3.7971e-02, -2.9955e-02, 1.1824e-02], [-3.4370e-02, -3.7345e-02, -3.8477e-02, 5.5241e-03, -1.9571e-02], [7.9875e-03, -3.7816e-02, -2.6429e-03, 1.3940e-02, 3.0520e-02], [2.3575e-02, -5.2586e-03, 2.0210e-02, -1.5240e-02, 4.1551e-02], [1.9832e-02, -1.5046e-02, 4.1453e-02, -2.3533e-03, 3.8019e-02]]])])])	R

	[[[-1.6473e-02, 4.5238e-03, -3.2498e-02, -3.0851e-02, 4.2595e-03], [-3.7861e-02, -1.1169e-02, -4.0175e-02, 3.4822e-02, -2.8040e-02], [-2.2453e-02, -2.5817e-02, 1.4361e-02, -1.7718e-02, 2.7322e-02], [2.5556e-02, 1.4051e-03, -2.2430e-02, 3.9491e-02, -2.2464e-02], [2.7719e-03, 3.9064e-02, -3.0062e-02, -1.2917e-02, 3.1602e-02]],	G
	[[1.9218e-02, -4.1580e-03, 7.2489e-03, -1.6464e-02, -1.4602e-02], [2.8440e-02, -2.6105e-02, -1.7572e-02, 2.4878e-03, -5.3821e-04], [5.0526e-03, -2.8758e-03, 2.9153e-02, -3.0721e-02, 2.1811e-02], [-2.3324e-02, -3.4958e-02, 1.4119e-02, 3.3497e-02, 1.4160e-04], [-2.4920e-02, 2.2221e-02, 2.0784e-02, -2.7165e-03, -2.0728e-02]]],	B
Conv(1,2)	[[[-1.1633e-01, 1.1982e-01, 4.9633e-01, 3.4875e-01, 1.2001e-01], [1.6421e-01, 4.7390e-01, 5.2399e-01, 3.8298e-01, 1.5880e-01], [-5.5281e-02, 1.4212e-01, 2.5049e-01, 5.1914e-02, 8.0453e-02], [1.3846e-01, 3.7066e-01, 7.4125e-01, 5.2642e-01, 2.3841e-01], [-9.9450e-02, -2.2295e-01, 4.1384e-02, -2.1325e-01, -3.9349e-01]],	R
	[[[-1.0803e-01, 8.3968e-02, 5.1885e-01, 3.6179e-01, 1.1305e-01], [1.6131e-01, 4.3266e-01, 5.6114e-01, 3.4895e-01, 1.5059e-01], [-3.1965e-02, 1.5282e-01, 2.4071e-01, 5.6428e-02, 7.3730e-02], [1.5030e-01, 4.1705e-01, 7.2744e-01, 5.2250e-01, 3.0447e-01], [-9.7756e-02, -1.7222e-01, 2.5437e-03, -2.4708e-01, -3.8172e-01]],	G
	[[[-6.7762e-02, 9.2885e-02, 5.2182e-01, 3.8224e-01, 5.0220e-02], [1.2202e-01, 4.7828e-01, 5.4580e-01, 4.0940e-01, 1.3995e-01], [6.8984e-03, 1.6003e-01, 1.8388e-01, 3.4706e-02, 4.3903e-02], [1.8085e-01, 4.1364e-01, 7.3455e-01, 5.0381e-01, 3.0095e-01], [-1.0389e-01, -1.6650e-01, -3.7670e-03, -2.4376e-01, -4.1740e-01]]],	B
Conv(1,3)	[[[1.9990e-01, 7.6986e-02, -2.9195e-02, -1.8279e-02, -1.2548e-01], [1.9609e-01, 5.8542e-02, -3.0029e-02, -2.5787e-02, -1.0994e-01], [8.6623e-02, 1.8275e-02, 3.7302e-02, 1.6994e-02, -3.8109e-02], [2.5069e-01, 1.4587e-01, 7.1159e-02, -2.6058e-02, -1.3887e-01], [1.0398e-01, -5.4911e-02, -1.5339e-01, -8.5409e-02, -2.0585e-01]],	R
	[[1.8229e-01, 1.1932e-01, 2.2964e-02, -5.8886e-02, -1.5307e-01], [1.6479e-01, 5.8908e-02, -1.2278e-02, -1.9185e-02, -6.9305e-02], [8.8240e-02, -1.9372e-02, 3.3282e-03, -3.8861e-02, -5.7180e-02], [2.5649e-01, 1.1352e-01, 4.0949e-02, 1.3186e-02, -1.3411e-01], [8.7623e-02, -5.8059e-02, -1.5492e-01, -1.2071e-01, -2.0826e-01]],	G
	[[1.4573e-01, 1.2702e-01, 1.1204e-02, -8.9999e-03, -1.4730e-01], [2.1884e-01, 1.1663e-01, -3.0397e-02, -3.8337e-02, -5.8662e-02], [1.1148e-01, 5.2120e-02, 2.9375e-02, -5.0550e-03, -1.6799e-02], [2.5559e-01, 8.8333e-02, 7.8354e-02, -1.2349e-02, -1.4475e-01], [1.2214e-01, 1.2817e-02, -1.4584e-01, -1.2194e-01, -2.0837e-01]]],	B
Conv(1,4)	[[[5.1272e-03, 7.7845e-03, -1.9001e-02, -4.0027e-02, 3.6667e-02], [7.5412e-03, -1.6222e-02, 1.1211e-02, -1.4130e-02, -7.7633e-03], [-2.2396e-03, -1.4387e-02, -1.1857e-02, -1.2368e-02, 3.8869e-02], [-3.3922e-02, 2.1822e-02, -9.5527e-03, -2.5671e-02, -3.7384e-02], [9.4338e-03, -2.2718e-02, -1.3764e-02, -4.1782e-03, 1.4679e-02]],	R
	[[3.1962e-02, 1.0376e-02, -2.0604e-02, 9.2206e-03, -2.4462e-03], [-2.9628e-03, 1.3032e-02, 2.2493e-02, -2.2454e-02, -3.1197e-02], [-3.4605e-02, 4.9074e-03, 5.1742e-03, -2.3869e-02, -3.0994e-02], [-1.8821e-02, -3.9444e-02, -1.3117e-02, -3.5639e-02, 6.6947e-03], [-6.8851e-03, -3.6554e-02, -3.3723e-02, -1.7804e-02, 5.1910e-03]],	G
	[[9.9619e-03, 3.4835e-02, 2.8681e-03, 3.8402e-02, -1.0893e-02], [-3.1886e-02, -1.9598e-02, -3.5492e-02, 1.4114e-02, 1.1325e-03], [-1.9885e-02, -1.4868e-02, 2.0764e-02, -2.3764e-02, 1.2154e-02], [8.1970e-03, -1.9856e-02, -3.6212e-02, -3.0718e-02, 3.7154e-02], [-5.1164e-03, -1.1956e-02, 1.1948e-02, 1.0268e-02, -2.7582e-02]]],	B
Conv(1,5)	[[[2.0507e-03, 1.1936e-02, -3.0707e-02, -3.5579e-02, 2.5269e-02], [-2.9086e-02, -3.5223e-02, -1.3532e-02, -3.1934e-02, -1.4756e-04], [-3.7890e-02, -7.7861e-03, -2.0282e-02, -6.6329e-03, -2.3737e-02], [-6.1798e-03, -3.3332e-02, 3.3759e-02, 2.3013e-04, -2.6626e-03],	R

	[-1.8802e-02, 3.7475e-03, 8.5041e-03, 1.7729e-02, -1.3007e-02]],	
	[[3.1766e-02, 1.6331e-02, -2.3003e-02, 2.8808e-02, -3.0057e-02], [1.4100e-02, 9.9300e-03, -2.1746e-02, 3.5413e-03, -1.0030e-02], [-1.5589e-02, 3.0236e-03, -3.3313e-02, 1.1062e-02, -8.2221e-03], [-1.0323e-02, -2.7932e-02, -2.9867e-02, -2.0007e-03, 9.6119e-03], [-2.9890e-02, -3.6804e-02, -8.0669e-04, -1.8423e-02, -3.6343e-02]],	G
	[[2.4226e-02, -3.8372e-02, 3.4295e-02, 1.5286e-02, -3.6000e-02], [-5.7448e-03, 1.0065e-02, 1.5729e-02, 1.4004e-02, -2.1275e-02], [-2.9758e-02, 3.3075e-02, 3.4555e-02, -9.5503e-03, -3.7708e-02], [1.7208e-02, -2.2608e-02, -3.8017e-02, -3.8513e-02, 3.6860e-02], [-2.5387e-02, 1.6212e-02, -1.8951e-02, -1.6967e-02, -3.5452e-02]]],	B
Conv(1,6)	[[[6.2404e-03, 3.6737e-02, -3.0952e-02, 1.7836e-02, -1.3794e-02], [-2.3046e-03, -7.4019e-03, 2.7439e-02, -6.7033e-03, 1.3265e-02], [2.5971e-03, -1.3865e-02, -1.4494e-02, -1.5182e-02, 1.4534e-02], [2.1527e-02, -3.2548e-02, -1.3516e-02, 1.0287e-02, 2.3148e-03], [-1.7734e-02, -7.4336e-03, -2.2842e-02, 2.0157e-03, 1.6148e-02]],	R
	[[2.1144e-02, -1.9653e-02, 5.2045e-03, -2.1007e-02, -1.8186e-03], [2.3860e-03, 1.2678e-02, 1.1592e-02, 3.2063e-02, -3.5388e-02], [2.9454e-02, -3.7332e-02, -9.1750e-03, -6.3340e-03, 1.3085e-02], [-1.7844e-02, -7.2246e-03, 3.8863e-02, -3.4902e-02, 1.0664e-02], [3.2504e-02, -1.5321e-02, 3.4310e-02, -6.4388e-03, 1.2881e-02]],	G
	[[-3.9276e-02, -1.0605e-02, -3.3806e-02, -3.0661e-02, 2.8134e-03], [2.9698e-02, -2.1744e-02, -1.8989e-02, 1.9458e-03, 5.8158e-03], [-3.1983e-02, 2.6221e-02, 4.6968e-04, 9.0706e-03, 3.3329e-02], [-1.2740e-02, -1.4794e-02, -3.2827e-02, 1.4619e-02, -3.3684e-02], [4.4150e-03, 2.4252e-02, -3.9325e-02, 1.8008e-02, 1.2518e-03]]],	B

Konvolüsyon 1. katman çıktıların matris gösterimi		
Maske	Conv1_ çıktısı : torch.Size([6,236,316])	Renk Modeli
Conv(1,1)	[2.5775e+00, 2.4448e+00, 2.5080e+00, ..., -1.3647e+01, - 1.3487e+01, -1.3330e+01], [4.9239e+00, 5.9347e+00, 7.0364e+00, ..., -1.3647e+01, - 1.3487e+01, -1.3330e+01], [1.1099e+01, 1.3790e+01, 1.6670e+01, ..., -1.3647e+01, - 1.3487e+01, -1.3330e+01],	R
	...,	G
	[-1.0999e+00, -1.0999e+00, -1.0999e+00, ..., -1.1515e+01, - 1.1381e+01, -1.1326e+01], [-9.5579e-01, -9.5579e-01, -9.5579e-01, ..., -1.1515e+01, - 1.1381e+01, -1.1326e+01], [-9.0575e-01, -9.0575e-01, -9.0575e-01, ..., -1.1515e+01, - 1.1381e+01, -1.1326e+01]],	B
Conv(1,2)	[[4.2198e+02, 4.2091e+02, 4.1983e+02, ..., 1.9912e+03, 1.9882e+03, 1.9885e+03], [4.3879e+02, 4.1061e+02, 3.9135e+02, ..., 1.9912e+03, 1.9882e+03, 1.9885e+03], [4.8727e+02, 5.1879e+02, 5.6917e+02, ..., 1.9912e+03, 1.9882e+03, 1.9885e+03],	R
	...,	G
	[1.1321e+02, 1.1321e+02, 1.1321e+02, ..., 1.6675e+03, 1.6615e+03, 1.6552e+03], [1.0189e+02, 1.0189e+02, 1.0189e+02, ..., 1.6675e+03, 1.6615e+03, 1.6552e+03],	B

Havuzlama 1. katman çıktılarının matris gösterimi		
Maske	maksimum havuzlama1_cikti: torch.Size([6, 3, 2,2])	Renk Modeli
Maxpool1 (1,1)	tensor([[[[5.9347e+00,8.1774e+00, 1.0703e+01, ..., -1.3865e+01,- 1.3647e+01, -1.3330e+01], [2.4671e+01, 2.6385e+01, 2.7974e+01, ..., -1.3865e+01, - 1.3647e+01, -1.3330e+01], [2.2907e+01, 1.9192e+01, 1.0084e+01, ..., -1.3422e+01, - 1.3149e+01, -1.3147e+01], ..., ..., [-9.6703e-01, -9.6703e-01, -8.8663e-01, ..., -1.1172e+01, -1.1472e+01, - 1.1387e+01], [-9.8276e-01, -9.8276e-01, -9.5061e-01, ..., -1.1135e+01, - 1.1412e+01, -1.1326e+01], [-9.0575e-01, -9.0575e-01, -8.0465e-01, ..., -1.1390e+01, - 1.1412e+01, -1.1326e+01]],	R
	...	G
	...	B
Maxpool1 (1,2)	[[4.3879e+02, 4.2133e+02, 4.2969e+02, ..., 2.0223e+03, 2.0003e+03, 1.9885e+03], [1.0322e+03, 1.2666e+03, 1.3941e+03, ..., 2.0223e+03, 2.0003e+03, 1.9885e+03], [2.1477e+03, 2.3861e+03, 2.5524e+03, ..., 2.0291e+03, 2.0120e+03, 1.9987e+03], ..., ..., [1.2341e+02, 1.2341e+02, 1.2510e+02, ..., 1.6852e+03, 1.6822e+03, 1.6695e+03], [1.1321e+02, 1.1321e+02, 1.1581e+02, ..., 1.6774e+03, 1.6742e+03, 1.6615e+03], [1.0189e+02, 1.0189e+02, 1.0531e+02, ..., 1.6792e+03, 1.6742e+03, 1.6615e+03]],	R
	...	G
	...	B
Maxpool1 (1,3)	[[7.5081e+00, -1.9961e-01, -4.9881e+00, ..., 1.1664e+02, 1.1660e+02, 1.1021e+02], [-9.6094e+01, -6.3774e+01, -1.4728e+01, ..., 1.1664e+02, 1.1660e+02, 1.1021e+02], [2.7851e+01, 4.4842e+01, 7.8845e+01, ..., 1.1324e+02, 1.1514e+02, 1.1210e+02], ..., ..., [7.8653e+00, 7.8653e+00, 7.0419e+00, ..., 9.1061e+01, 9.6159e+01, 9.5853e+01], [7.5858e+00, 7.5858e+00, 6.3849e+00, ..., 9.1974e+01, 9.5395e+01, 9.5088e+01], [6.3079e+00, 6.3079e+00, 5.1888e+00, ..., 9.4444e+01, 9.5395e+01, 9.5088e+01]],	R
	...	G
	...	B
Maxpool1 (1,4)	[[-2.2814e+01, -2.3131e+01, -2.3777e+01, ..., -7.4941e+01, -7.4235e+01, -7.3959e+01], [-4.5587e+01, -5.3431e+01, -6.2179e+01, ..., -7.4941e+01, - 7.4235e+01, -7.3959e+01], [-9.3320e+01, -9.9318e+01, -1.0589e+02, ..., -7.5251e+01, - 7.4613e+01, -7.4264e+01], ..., ...	R
	...	G

	<p>[-4.1194e+00, -4.1194e+00, -4.1449e+00, ..., -6.2269e+01, -6.2057e+01, -6.1573e+01],</p> <p>[-3.7515e+00, -3.7515e+00, -3.7094e+00, ..., -6.2407e+01, -6.2151e+01, -6.1667e+01],</p> <p>[-3.4001e+00, -3.4001e+00, -3.3958e+00, ..., -6.2539e+01, -6.2151e+01, -6.1667e+01]],</p> <p>....</p>	B
Maxpool1 (1,5)	<p>[[[-2.5523e+01, -2.6008e+01, -2.7213e+01, ..., -8.9044e+01, -8.8453e+01, -8.8236e+01],</p> <p>[-5.3865e+01, -6.1240e+01, -6.6747e+01, ..., -8.9044e+01, -8.8453e+01, -8.8236e+01],</p> <p>[-9.8312e+01, -1.0552e+02, -1.1035e+02, ..., -8.9493e+01, -8.8982e+01, -8.8613e+01],</p> <p>....</p>	R
		G
	<p>[-4.9389e+00, -4.9389e+00, -5.0609e+00, ..., -7.4158e+01, -7.3848e+01, -7.3459e+01],</p> <p>[-4.4680e+00, -4.4680e+00, -4.6194e+00, ..., -7.4154e+01, -7.3845e+01, -7.3455e+01],</p> <p>[-4.1004e+00, -4.1004e+00, -4.2798e+00, ..., -7.4331e+01, -7.3845e+01, -7.3455e+01]],</p> <p>....</p>	B
Maxpool1 (1,6)	<p>[[[-3.7545e+00, -2.9861e+00, -1.4741e+00, ..., -2.0737e+01, -2.0377e+01, -2.0304e+01],</p> <p>[-3.2552e+00, -4.4784e+00, -8.1267e+00, ..., -2.0737e+01, -2.0377e+01, -2.0304e+01],</p> <p>[-1.4442e+01, -1.4361e+01, -1.5020e+01, ..., -2.0667e+01, -2.0298e+01, -2.0281e+01],</p> <p>....</p>	R
		G
	<p>[-1.2779e+00, -1.2779e+00, -1.2377e+00, ..., -1.7166e+01, -1.7215e+01, -1.7042e+01],</p> <p>[-1.1456e+00, -1.1456e+00, -1.1249e+00, ..., -1.7145e+01, -1.7148e+01, -1.6974e+01],</p> <p>[-1.0140e+00, -1.0140e+00, -9.7258e-01, ..., -1.7190e+01, -1.7148e+01, 1.6974e+01]]],</p> <p>....</p>	B

Conv1_max1 (1,1) ispatı	
Conv1_max1 (1,1)	
<p>tensor([[2.5775, 2.4448, 2.5080, 2.8157, 3.2716, 2.9482],</p> <p>[4.9239, 5.9347, 7.0364, 8.1774, 9.6808, 10.7027],</p> <p>[11.0994, 13.7904, 16.6704, 18.6157, 19.7824, 22.4511],</p> <p>[22.2927, 24.6711, 26.0775, 26.3850, 27.9735, 25.2384]],</p>	<p>tensor([[[[5.9347e+00, 8.1774e+00, 1.0703e+01, ..., -1.3865e+01, -1.3647e+01, -1.3330e+01],</p> <p>[2.4671e+01, 2.6385e+01, 2.7974e+01, ..., -1.3865e+01, -1.3647e+01, -1.3330e+01],</p>

Aktivasyon 1. katman çıktılarının matris gösterimi		
Maske	RELU1 çıktı : torch.Size([6, 3, 2,2])	Renk Modeli
RELU (1,1)	<p>tensor([[[[5.9347, 8.1774, 10.7027, ..., 0.0000, 0.0000, 0.0000],</p> <p>[24.6711, 26.3850, 27.9735, ..., 0.0000, 0.0000, 0.0000],</p>	R

	[22.9067, 19.1920, 10.0840, ..., 0.0000, 0.0000, 0.0000], ..., ...,	G
	[0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000], [0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000], [0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000]], ..., ...	B
RELU (1,2)	[[438.7896, 421.3314, 429.6882, ..., 2022.3013, 2000.3180, 1988.5315], [1032.2408, 1266.5627, 1394.0509, ..., 2022.3013, 2000.3180, 1988.5315], [2147.7410, 2386.1128, 2552.4294, ..., 2029.0959, 2011.9968, 1998.6558], ..., ...,	R
	..., ...	G
	[123.4080, 123.4080, 125.1038, ..., 1685.1622, 1682.1801, 1669.5095], [113.2135, 113.2135, 115.8116, ..., 1677.3590, 1674.2085, 1661.5377], [101.8887, 101.8887, 105.3149, ..., 1679.1959, 1674.2085, 1661.5377]], ..., ...	B
RELU (1,3)	[[7.5081, 0.0000, 0.0000, ..., 116.6430, 116.5981, 110.2065], [0.0000, 0.0000, 0.0000, ..., 116.6430, 116.5981, 110.2065], [27.8510, 44.8424, 78.8452, ..., 113.2431, 115.1448, 112.0950], ..., ...,	R
	..., ...	G
	[7.8653, 7.8653, 7.0419, ..., 91.0612, 96.1589, 95.8525], [7.5858, 7.5858, 6.3849, ..., 91.9737, 95.3948, 95.0885], [6.3079, 6.3079, 5.1888, ..., 94.4442, 95.3948, 95.0885]], ..., ...	B
RELU (1,4)	[[0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000], [0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000], [0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000], ..., ...,	R
	..., ...	G
	[0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000], [0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000], [0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000]], ..., ...	B
RELU (1,5)	[0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000], [0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000],	R

	[0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000],	
	...,	G
	[0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000],	B
	[0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000],	
	[0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000],	
	...,	
RELU (1,6)	[0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000],	R
	[0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000],	
	[0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000],	
	...,	
	...,	G
	[0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000],	B
	[0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000],	
	[0.0000, 0.0000, 0.0000, ..., 0.0000, 0.0000, 0.0000],	
	...,	

Konvolüsyon 2. katman maskelerinin matris gösterimi	
Maske	Conv2_kernel : torch.Size([8, 6, 5, 5])
Conv(2,1)	tensor([[[[-2.0355e-02, -1.2098e-02, -1.4757e-02, 4.4881e-03, 6.4913e-03], [7.1501e-03, 6.7534e-03, -3.4954e-05, -1.5406e-02, 2.9992e-03], [2.1904e-03, 1.1171e-02, -1.3158e-02, 5.9075e-03, -1.2584e-02], [2.2261e-02, -1.8504e-02, -2.0047e-02, 2.2520e-04, 2.0823e-02], [-1.6935e-02, 9.6953e-03, -1.8361e-02, 1.3287e-02, -3.2003e-03]]], [[1.6191e-01, 1.1744e-01, -2.1782e-01, -2.6491e-01, -4.1859e-03], [1.6712e-01, 1.0712e-01, -1.9758e-01, -2.4844e-02, 1.4285e-01], [1.2809e-01, 1.1548e-01, -3.8725e-02, 2.4075e-01, 3.1382e-01], [1.2540e-01, -8.2391e-02, -1.7659e-01, 1.0450e-01, 2.1668e-01], [-2.2104e-02, -3.3989e-01, -4.0495e-01, -1.5052e-01, -7.6098e-03]]], [[4.1158e-02, 7.6874e-02, 5.2558e-02, -8.5684e-02, -3.6246e-02], [1.1509e-02, 6.4531e-02, 1.2981e-02, -5.0828e-02, -1.9453e-02], [3.8118e-02, 2.8617e-02, 4.2332e-02, -5.4167e-02, 5.5688e-02], [4.6401e-02, 2.8726e-02, 2.5911e-03, -3.8670e-03, 5.6570e-02], [6.0245e-02, 2.9680e-02, -7.6306e-02, -5.7005e-02, 8.4485e-03]]], [[-5.1922e-03, -1.6310e-02, -1.9238e-02, -1.6579e-02, 3.7262e-03], [-2.0612e-03, -4.1504e-05, -8.4557e-03, -1.4251e-02, -1.0442e-02], [1.4378e-02, -1.5333e-02, 7.2278e-03, 1.4277e-02, 2.0622e-02], [8.0581e-03, -6.3207e-03, -1.2842e-02, 1.1009e-02, 1.5779e-02], [2.5380e-03, 1.7680e-02, 2.6260e-02, 1.1310e-02, -1.7531e-02]]], [[-8.0231e-03, 1.5349e-02, -1.0426e-02, -1.0550e-03, -2.0265e-02], [-2.3389e-02, 2.3500e-02, 2.7770e-02, 2.0052e-02, 2.4760e-02], [-1.9453e-02, -1.2283e-03, -3.6807e-03, -1.9328e-03, 2.3280e-02], [1.2358e-02, 2.7344e-02, -4.4679e-03, 8.5191e-03, -2.7727e-03], [1.5885e-02, 5.1653e-03, 1.0151e-02, 2.3022e-02, -1.5315e-02]]], [[-1.0435e-02, 1.4466e-02, -2.2577e-04, 2.2549e-02, -1.7524e-02], [2.3233e-02, 2.6780e-02, 3.2363e-03, -1.4830e-02, 1.4241e-02], [-2.4103e-02, -1.5755e-02, -2.4391e-03, 1.7666e-02, 1.6229e-03], [3.6353e-03, -3.4429e-03, 4.2836e-03, -6.6951e-03, 1.0782e-02], [-3.8426e-03, 2.5358e-02, -1.2994e-02, 1.6103e-02, 2.7470e-02]]]]],
Conv(2,2)	[[[1.9805e-03, 2.3675e-02, 3.9660e-03, 2.2944e-03, 5.4889e-03], [-1.2379e-02, 1.9329e-02, -2.3209e-03, 2.9475e-02, -2.0952e-02],

	[1.3524e-02, 2.2679e-02, 2.0525e-02, -2.3058e-02, -1.0106e-02], [2.3116e-02, -1.8198e-02, 1.8541e-02, 6.6086e-03, 1.1003e-03], [-1.5906e-02, -2.7106e-02, -1.9287e-02, 5.9043e-03, -1.1439e-02]],
	[[-4.7731e-02, -1.4439e-01, 3.7608e-02, 1.8806e-01, 2.6090e-01], [-6.8936e-02, -3.6749e-02, 2.2607e-02, 2.9547e-01, 3.0624e-01], [2.2122e-01, 7.5425e-02, 4.6174e-02, 2.2326e-01, 2.1488e-01], [3.2050e-01, -8.8040e-02, -2.1612e-01, -1.4512e-01, 2.7208e-02], [6.8485e-02, -4.4615e-01, -5.0146e-01, -2.2992e-01, -6.7512e-02]],
	[[1.6895e-02, 1.4510e-02, -4.0697e-02, -4.1150e-02, 2.9523e-02], [-3.5001e-02, -2.5615e-02, -1.3263e-02, -1.1110e-02, -4.6519e-03], [3.6115e-02, 5.9868e-02, 2.9553e-04, -2.3705e-02, -7.9804e-03], [1.0155e-01, 6.0471e-02, -2.1872e-02, -8.0408e-02, 3.6526e-03], [6.4533e-02, 7.0058e-02, -8.1830e-02, -8.0915e-02, -1.2030e-02]],
	[[4.3320e-03, -6.1162e-03, 2.1778e-02, -2.0398e-02, 1.2264e-02], [1.6497e-02, -9.3982e-03, -6.4021e-03, 8.0361e-03, -2.8001e-03], [-4.7831e-03, 1.9291e-02, 2.7357e-02, 1.7997e-04, -9.5436e-03], [-1.3138e-02, -6.0343e-03, 1.8523e-02, -2.5874e-02, -2.5433e-02], [2.6834e-03, -2.7891e-02, -1.2453e-02, 1.5825e-02, - 3.7890e-03]],
	[[-2.3401e-02, 5.6405e-03, -1.5882e-02, -2.6948e-02, -2.5482e-02], [2.1489e-03, -2.2715e-02, -1.1115e-02, 5.5475e-03, -1.1649e-03], [-2.5358e-02, 2.0444e-02, 2.6087e-02, 2.6071e-02, -1.4036e-02], [-1.2450e-02, 2.2536e-02, 3.1154e-03, -1.8396e-02, -1.1967e-03], [2.7466e-02, 1.5672e-02, 4.3353e-03, 5.1220e-03, 1.0472e-02]],
	[[-1.3380e-02, -2.0749e-02, 1.7510e-02, -7.9883e-03, 2.2598e-02], [-1.9109e-02, -1.1454e-02, -1.4521e-02, 1.9398e-02, -1.9951e-03], [2.7411e-02, 5.4738e-03, -5.1859e-03, -2.7110e-02, -6.3012e-03], [4.7263e-03, 1.1802e-02, -1.2115e-02, 2.7443e-03, -2.4053e-02], [-1.0799e-02, 2.6907e-02, -5.3518e-03, -1.3473e-02, -1.2407e-02]]],
Conv(2,3)	[[[-2.9605e-02, -1.3839e-02, -1.1868e-02, -4.9514e-03, -3.1924e-02], [-1.5543e-02, -1.5205e-02, -1.8217e-02, -1.2231e-02, -4.1344e-03], [-4.3361e-03, -2.8220e-02, 8.7284e-03, -1.0626e-02, 2.6766e-02], [-3.3062e-02, -2.0694e-02, 2.3623e-02, 2.0289e-02, -4.3157e-03], [1.5762e-02, 1.3326e-02, 1.4297e-02, -7.7605e-03, -2.5359e-02]],
	[[-7.1250e-01, -4.1595e-01, -6.9329e-02, 1.4729e-01, -3.0424e-01], [-7.8313e-01, -3.1516e-01, 1.9118e-01, 9.3723e-01, 6.5548e-01], [-3.8996e-01, -2.2054e-01, 3.1770e-01, 1.1998e+00, 9.1032e-01], [-2.8626e-01, -5.6496e-01, 3.9435e-01, 7.3005e-01, 1.3963e-01], [-3.6806e-01, -1.7665e-01, 2.4622e-01, 7.8686e-02, -7.7758e-01]],
	[[1.1291e-01, -2.3450e-01, -1.0746e-01, 1.1559e-01, 1.5991e-01], [4.9171e-02, -2.3567e-01, -1.5330e-01, 8.5569e-02, 2.7219e-01], [1.4342e-01, -1.7430e-01, -8.1163e-02, 9.0542e-02, 2.7644e-01], [2.3826e-01, -1.1081e-01, -1.5638e-01, 8.3091e-02, 2.1147e-01], [2.2061e-01, -4.4225e-02, 2.0228e-03, -5.9079e-02, 5.6982e-02]],
	[[-2.7578e-02, -7.7125e-03, -7.6383e-03, 1.8458e-02, 2.3474e-02], [1.1592e-02, 8.1827e-03, 5.2262e-03, -2.0337e-02, 1.8900e-02], [-1.3922e-02, 1.9789e-02, 5.7035e-03, -9.2313e-03, 1.5757e-02], [2.0333e-02, 1.9535e-02, 2.0440e-03, -1.4722e-02, -5.9525e-03], [-1.8061e-02, -2.4671e-02, 1.5183e-03, -2.7516e-02, -2.7423e-02]],
	[[4.0501e-03, -7.1614e-04, -1.6568e-02, -2.5801e-02, -2.5624e-02], [7.1105e-03, 2.7888e-02, -2.2014e-02, -1.7459e-02, 1.6717e-02], [-5.2710e-03, 1.0884e-02, 2.4387e-02, 5.5646e-03, 5.0129e-03], [2.1382e-02, 1.0947e-02, 2.6477e-02, -7.3691e-03, -2.2498e-02], [-2.1556e-02, -1.6078e-02, -1.1519e-02, 2.9573e-02, -2.3128e-03]],
	[[-1.8740e-02, 1.0542e-03, 2.7248e-02, 1.3716e-02, 1.6614e-02], [-7.9068e-03, 1.7253e-02, 2.1176e-02, 9.9838e-03, 1.2372e-02], [-7.7820e-03, -2.2889e-02, 4.1670e-03, 1.9720e-02, 2.4879e-02], [-1.2400e-02, -7.7728e-04, -2.1170e-02, 6.0655e-03, 2.3946e-02], [-2.2876e-02, 2.0286e-02, 2.2602e-02, -5.6833e-03, -1.8037e-02]]],

Conv(2,4)

Conv(2,5)

Conv(2,6)	[[[-1.5727e-02, 1.4098e-02, -2.5947e-02, 2.4610e-02, -8.5704e-03], [-2.5226e-02, -9.1544e-03, 1.1602e-02, 1.0868e-02, 2.6043e-02], [-2.2315e-02, 4.4134e-03, -1.2964e-02, 8.7896e-03, -9.0685e-03], [-1.0354e-02, -9.8738e-03, -8.3354e-03, -7.7655e-03, -2.3478e-02], [1.8412e-02, -2.7153e-03, -3.5884e-03, 1.4148e-02, 7.6397e-03]],
	[[3.6858e-02, 5.2897e-02, -1.9267e-03, 1.5187e-01, 3.0667e-02], [3.2949e-01, 3.2606e-01, 2.6078e-01, 2.0037e-01, 1.3002e-01], [2.7734e-01, 2.1943e-01, -9.6954e-03, -2.2490e-01, -3.2974e-01], [4.2493e-02, 3.1610e-02, -1.5705e-01, -5.9958e-01, -6.9921e-01], [4.1574e-02, 1.0052e-01, -3.2934e-03, -2.0600e-01, -4.2237e-01]],
	[[-3.2293e-02, 5.8227e-02, 3.6866e-02, 2.1842e-02, 3.6456e-02], [3.0994e-02, 1.0849e-01, 8.8856e-02, 8.9965e-02, 8.3017e-02], [1.8548e-02, 7.7095e-02, 5.9722e-02, 5.1231e-02, -3.8699e-02], [-7.3487e-02, -5.5378e-03, 6.2366e-02, 9.1252e-03, -1.1622e-01], [-1.6450e-02, 9.0409e-04, 2.5629e-02, 5.0538e-02, -6.3842e-02]],
	[[-1.6440e-02, -2.5733e-02, 1.9935e-02, 4.9024e-03, -4.3036e-03], [3.2604e-03, 4.0668e-03, -4.6524e-03, 2.0301e-02, 6.1225e-03], [1.1657e-02, -1.7274e-03, -2.1504e-02, -1.7515e-02, 2.7903e-02], [2.3301e-02, 2.1944e-02, 3.3585e-03, 5.9324e-03, -2.1644e-02], [-2.3756e-02, 4.8538e-03, 2.2055e-02, -1.5835e-03, 2.1334e-03]],
	[[2.5270e-02, 2.4593e-02, 2.2319e-02, 1.6119e-02, 1.1814e-02], [-2.5255e-02, 2.5816e-02, -9.7181e-03, -1.7681e-02, -1.9156e-02], [1.5127e-03, 2.6079e-02, 2.2119e-02, -5.4546e-03, -1.5623e-02], [-7.0227e-03, 1.0690e-02, -8.6402e-03, -2.6404e-02, 1.3622e-02], [-2.6547e-02, 2.4656e-02, -8.0690e-03, -1.9565e-02, 1.5856e-03]],
	[[-1.2717e-02, 1.9487e-02, 1.7966e-02, -1.4397e-02, -2.1629e-02], [1.4339e-03, 3.0448e-03, 1.0715e-02, 1.4169e-02, -1.4294e-02], [-1.4257e-02, 2.0725e-04, -3.0407e-03, -1.1126e-02, 2.0843e-02], [-1.4045e-02, 2.1497e-02, 2.1303e-02, 1.2965e-03, -1.4162e-02], [-6.4177e-03, -1.2292e-02, 5.9578e-03, -1.5790e-02, 9.8719e-03]]],
	[[[-8.5619e-03, -2.1091e-02, -8.5108e-03, 1.3417e-02, 1.4494e-02], [-2.5063e-02, -4.0130e-03, 2.0094e-02, 5.9003e-03, -1.3092e-02], [1.2602e-02, 8.9379e-03, 8.1024e-03, 4.0452e-03, 6.1752e-03], [-3.5254e-03, 4.1155e-03, 1.6043e-02, 1.9986e-02, -2.7190e-02], [-6.8064e-03, -4.4502e-03, 4.5652e-03, -2.6863e-02, 2.4986e-02]],
[[5.4702e-02, 4.8700e-02, -1.4575e-02, -1.7218e-02, -6.3924e-02], [2.0778e-02, 5.3473e-02, 2.8444e-02, -4.8657e-02, -2.5290e-02], [-3.9483e-03, 3.3003e-02, 1.9305e-02, -2.9330e-02, 1.5701e-03], [-3.3710e-02, 1.7667e-02, -4.2973e-03, -1.7336e-02, -6.4685e-03], [-4.0927e-02, -1.5668e-02, 6.5049e-03, -3.9234e-03, -3.0732e-02]],	
[[1.8530e-02, 3.1696e-02, 3.3459e-02, -1.0681e-02, -3.3092e-02], [-1.1579e-02, 3.4001e-02, -1.8745e-03, 2.7273e-02, -3.5638e-03], [3.0855e-03, 1.7056e-02, -2.4814e-03, -1.7095e-02, 1.8036e-02], [-3.4168e-02, -1.0027e-02, -2.6195e-03, -3.3400e-03, -7.4301e-03], [-2.8438e-02, 1.7547e-02, 1.0185e-02, -2.7302e-02, 8.8751e-03]],	
Conv(2,7)	[[[-8.5619e-03, -2.1091e-02, -8.5108e-03, 1.3417e-02, 1.4494e-02], [-2.5063e-02, -4.0130e-03, 2.0094e-02, 5.9003e-03, -1.3092e-02], [1.2602e-02, 8.9379e-03, 8.1024e-03, 4.0452e-03, 6.1752e-03], [-3.5254e-03, 4.1155e-03, 1.6043e-02, 1.9986e-02, -2.7190e-02], [-6.8064e-03, -4.4502e-03, 4.5652e-03, -2.6863e-02, 2.4986e-02]],
	[[5.4702e-02, 4.8700e-02, -1.4575e-02, -1.7218e-02, -6.3924e-02], [2.0778e-02, 5.3473e-02, 2.8444e-02, -4.8657e-02, -2.5290e-02], [-3.9483e-03, 3.3003e-02, 1.9305e-02, -2.9330e-02, 1.5701e-03], [-3.3710e-02, 1.7667e-02, -4.2973e-03, -1.7336e-02, -6.4685e-03], [-4.0927e-02, -1.5668e-02, 6.5049e-03, -3.9234e-03, -3.0732e-02]],
	[[1.8530e-02, 3.1696e-02, 3.3459e-02, -1.0681e-02, -3.3092e-02], [-1.1579e-02, 3.4001e-02, -1.8745e-03, 2.7273e-02, -3.5638e-03], [3.0855e-03, 1.7056e-02, -2.4814e-03, -1.7095e-02, 1.8036e-02], [-3.4168e-02, -1.0027e-02, -2.6195e-03, -3.3400e-03, -7.4301e-03], [-2.8438e-02, 1.7547e-02, 1.0185e-02, -2.7302e-02, 8.8751e-03]],

	<pre> [[[-1.6676e-02, 2.3746e-02, -2.5501e-02, -2.6855e-02, -1.3947e-02], [2.2545e-02, 1.6113e-02, 1.3115e-02, 7.4563e-03, 1.9262e-02], [-1.2570e-02, -8.4198e-03, -9.0929e-03, 1.6043e-02, 1.0299e-02], [2.1301e-02, 9.5527e-03, 1.5673e-02, 2.2064e-02, -1.6607e-02], [1.6077e-02, -2.2760e-02, 1.3365e-02, -1.4252e-02, -1.2718e-02]], [[6.5951e-03, -3.6190e-03, 5.8563e-03, 9.0998e-03, 4.1059e-03], [2.5334e-02, -2.2384e-02, 6.7148e-03, 4.1619e-03, 1.9902e-03], [2.6590e-02, -7.8618e-03, 2.2178e-02, 1.9164e-03, -7.0030e-03], [-2.1713e-02, 2.5577e-02, -1.6736e-02, -2.9125e-03, 1.1818e-02], [8.3261e-03, -1.2319e-02, -1.3580e-02, 2.1804e-02, -1.5081e-02]], [[[-3.5212e-03, 1.2902e-02, -2.0142e-02, 2.2419e-02, -1.0583e-02], [2.6728e-03, -1.8797e-02, -9.9086e-03, 1.1752e-03, -7.5107e-03], [9.5095e-03, -2.4455e-02, 1.2431e-02, -5.4799e-03, -1.4084e-02], [-2.3680e-03, 2.3178e-02, 1.1355e-02, 2.3794e-02, -2.3432e-02], [7.8643e-03, -2.0233e-02, 1.1206e-02, 2.7880e-02, -1.1708e-02]]], Conv(2,8) [[[-8.5653e-03, -1.5497e-02, 1.7205e-02, 2.3868e-02, -7.7401e-03], [1.4049e-02, -2.3623e-03, -1.0810e-02, -1.6916e-02, -1.0923e-02], [1.1254e-02, 2.1092e-02, -1.3545e-02, -1.6617e-02, 1.4068e-02], [2.5692e-02, 1.0267e-02, 2.5693e-02, -2.2861e-02, 1.3978e-02], [2.3776e-02, 2.9598e-02, 2.7891e-04, 1.1753e-02, 2.0138e-02]], [[[-1.2971e-01, -1.8548e-01, -1.7361e-01, -1.1552e-01, -1.5195e-01], [-1.1999e-01, -1.2838e-01, -6.5802e-02, 2.7963e-02, -2.0560e-02], [-4.2746e-02, -1.0059e-01, -1.1012e-01, 8.6702e-02, 1.0426e-01], [-8.8178e-02, -1.2954e-01, 1.8389e-02, 2.9881e-01, 2.7267e-01], [-2.0557e-02, -8.4887e-02, 8.6938e-02, 3.5829e-01, 2.1041e-01]], [[[-3.1504e-03, 6.8682e-03, -1.7716e-02, -5.5886e-03, -3.5652e-03], [1.3147e-04, -1.2342e-02, -2.3557e-02, -3.0411e-03, 3.5415e-02], [4.4210e-02, -1.9930e-02, -1.3065e-02, 1.8920e-03, 5.0882e-02], [2.0563e-02, -2.0529e-02, -1.6111e-02, -6.9488e-03, 9.0894e-02], [4.2522e-03, 6.3638e-04, -5.3430e-02, 1.3672e-02, 6.5759e-02]], [[1.7702e-02, -1.2983e-02, -6.3968e-03, 1.5746e-03, -1.7270e-02], [-2.2132e-02, 9.3913e-03, -4.9298e-03, -1.0233e-02, 1.4042e-02], [4.3020e-03, 7.6342e-04, -7.1625e-04, -7.7097e-03, 1.7022e-02], [2.1082e-02, 1.0446e-02, 8.2507e-03, 2.0248e-02, -2.4820e-02], [-1.4966e-02, 2.1329e-02, -2.3733e-02, -2.2999e-02, -2.7209e-02]], [[2.2340e-03, -8.9278e-03, 2.5905e-02, 2.4627e-02, 2.3400e-03], [2.4145e-02, -1.6237e-02, 1.8368e-02, 8.5851e-03, 1.8428e-02], [-1.6672e-02, -6.7764e-03, 2.4102e-02, 5.2496e-03, 2.1344e-02], [1.3955e-02, 1.7059e-02, -1.0775e-02, -1.0392e-02, -1.1134e-02], [-9.7830e-03, 6.4508e-03, -2.1914e-02, -2.6712e-02, -1.0675e-02]], [[5.8668e-03, 1.1596e-02, 2.5441e-02, -2.1265e-02, 2.7095e-03], [-2.7863e-02, -1.4904e-02, -8.2492e-04, 2.5926e-02, -3.4055e-04], [1.9374e-02, -4.6030e-04, -4.4902e-03, 2.0107e-02, 1.8273e-02], [-1.5343e-02, 8.1552e-03, 1.5424e-02, 3.1187e-03, -1.8632e-02], [1.1701e-02, 1.7289e-02, -4.2567e-03, 2.5681e-02, -2.0117e-02]]], </pre>
--	--

Konvolüsyon 2. katman çıktılarının matris gösterimi	
Maske	Conv2 çıktısı : torch.Size([8,6,5,5])
Conv(2,1)	<pre> tensor([[[[3.1268e+01, 1.6295e+02, 1.9634e+02, ..., 4.4751e+01, 4.1107e+01, 2.6162e+01], [8.5725e+01, 1.3980e+02, 1.4227e+02, ..., 5.5107e+01, 5.7020e+01, 4.3898e+01], [-3.7101e+01, 4.4783e-02, 2.9592e+01, ..., 5.7649e+01, 6.1038e+01, 4.8488e+01], </pre>

	<p>...</p> <p>[2.1394e+01, 1.8948e+01, 2.2151e+01, ..., 5.1526e+01, 5.0605e+01, 3.9927e+01],</p> <p>[1.5766e+01, 8.7587e+00, 2.0828e+01, ..., 3.5306e+01, 4.2870e+01, 4.1843e+01],</p> <p>[1.2889e+01, 4.6143e+00, 1.7948e+01, ..., 2.5495e+01, 3.1349e+01, 2.9199e+01]],</p>
Conv(2,2)	<p>[[-6.5096e+02, -4.1029e+02, -1.1057e+02, ..., 6.1215e+02, 6.0332e+02, 5.7789e+02],</p> <p>[5.3473e+02, 6.0736e+02, 6.9652e+02, ..., 6.2286e+02, 6.2096e+02, 6.0061e+02],</p> <p>[8.7829e+02, 8.6836e+02, 8.5877e+02, ..., 6.3459e+02, 6.4111e+02, 6.2764e+02],</p> <p>...</p> <p>[9.9364e+01, 9.9110e+01, 1.0784e+02, ..., 5.4886e+02, 5.4079e+02, 5.2522e+02],</p> <p>[9.5076e+01, 9.3811e+01, 9.9779e+01, ..., 5.3196e+02, 5.3479e+02, 5.3078e+02],</p> <p>[8.3981e+01, 8.0936e+01, 8.8372e+01, ..., 5.1627e+02, 5.3236e+02, 5.3338e+02]],</p>
Conv(2,3)	<p>[[4.4983e+03, 4.4347e+03, 4.5582e+03, ..., 1.1277e+03, 1.1352e+03, 1.1202e+03],</p> <p>[4.1886e+03, 3.7120e+03, 3.4267e+03, ..., 1.1608e+03, 1.1849e+03, 1.1810e+03],</p> <p>[2.3017e+03, 1.9847e+03, 1.8549e+03, ..., 1.1762e+03, 1.2067e+03, 1.2056e+03],</p> <p>...</p> <p>[1.3053e+02, 1.2687e+02, 1.0644e+02, ..., 1.0109e+03, 1.0069e+03, 9.9557e+02],</p> <p>[1.3058e+02, 1.2514e+02, 8.7905e+01, ..., 1.0231e+03, 1.0219e+03, 1.0176e+03],</p> <p>[1.0381e+02, 8.6418e+01, 5.4785e+01, ..., 1.0164e+03, 1.0142e+03, 1.0084e+03]],</p>
Conv(2,4)	<p>...</p> <p>...</p> <p>...</p>
Conv(2,5)	<p>...</p> <p>...</p> <p>...</p>
Conv(2,6)	<p>[[-3.8238e+03, -3.5299e+03, -3.2710e+03, ..., -7.8455e+02, -7.8724e+02, -7.7677e+02],</p> <p>[-1.9295e+03, -1.6912e+03, -1.5457e+03, ..., -7.8564e+02, -7.9035e+02, -7.7758e+02],</p> <p>[-1.1970e+03, -1.1462e+03, -1.1340e+03, ..., -7.6582e+02, -7.6221e+02, -7.4444e+02],</p> <p>...</p> <p>[-2.5856e+01, -1.6325e+01, -1.0087e+01, ..., -6.3919e+02, -6.4478e+02, -6.3063e+02],</p> <p>[-1.1575e+01, -7.2284e-01, -1.5042e+00, ..., -6.4229e+02, -6.4444e+02, -6.2826e+02],</p> <p>[-2.8449e+00, 1.0238e+01, 9.5858e-01, ..., -6.4899e+02, -6.3669e+02, -6.1661e+02]],</p>
Conv(2,7)	<p>[[-3.0058e+02, -3.0626e+02, -3.2940e+02, ..., -1.4044e+02, -1.4035e+02, -1.3781e+02],</p> <p>[-2.9033e+02, -2.7214e+02, -2.7202e+02, ..., -1.4035e+02, -1.4062e+02, -1.3808e+02],</p>

	[-2.4552e+02, -2.2609e+02, -2.1485e+02, ..., -1.4027e+02, - 1.4085e+02, -1.3855e+02],

	[-9.8125e+00, -8.4594e+00, -8.3199e+00, ..., -1.1699e+02, - 1.1596e+02, -1.1456e+02],
	[-9.6843e+00, -9.0541e+00, -7.8285e+00, ..., -1.1826e+02, - 1.1725e+02, -1.1549e+02],
	[-8.1396e+00, -7.4822e+00, -6.8041e+00, ..., -1.1718e+02, - 1.1930e+02, -1.1770e+02]],
Conv(2,8)	[[1.7769e+03, 1.6589e+03, 1.5462e+03, ..., -4.0654e+02, -3.9491e+02, - 3.9846e+02],
	[6.6895e+02, 4.9205e+02, 3.5276e+02, ..., -4.0807e+02, - 3.9695e+02, -4.0118e+02],
	[-3.1303e+02, -4.0759e+02, -4.5337e+02, ..., -4.1762e+02, - 4.1216e+02, -4.1915e+02],

	[-6.6476e+01, -7.3734e+01, -7.8899e+01, ..., -3.5029e+02, -3.4411e+02, - 3.4816e+02],
	[-5.8533e+01, -7.0243e+01, -7.4639e+01, ..., -3.4540e+02, - 3.4501e+02, -3.4956e+02],
	[-5.6731e+01, -6.8196e+01, -6.8159e+01, ..., -3.3609e+02, - 3.4336e+02, -3.5111e+02]]],

Havuzlama 2. katman çıktılarının matris gösterimi	
Maske	maksimum havuzlama2 çıktı : torch.Size([8,6,2,2])
Maxpool2 (2,1)	tensor([[[[1.6295e+02, 1.9634e+02, 5.0156e+01, ..., 6.6065e+01, 5.5107e+01, 5.7020e+01], [8.2959e+01, 8.3910e+01, 7.8342e+01, ..., 6.7365e+01, 5.7649e+01, 6.1038e+01], [9.5033e+01, 1.0068e+02, 9.0132e+01, ..., 6.4144e+01, 4.6575e+01, 4.4289e+01], [3.0889e+01, 3.7981e+01, 3.1752e+01, ..., 1.8295e+01, 6.2446e+01, 5.3007e+01], [2.4517e+01, 3.3804e+01, 3.6654e+01, ..., -8.2910e+01, 6.0878e+01, 5.0605e+01], [1.5766e+01, 3.6140e+01, 3.0761e+01, ..., -2.8614e+02, 3.5306e+01, 4.2870e+01]],
Maxpool2 (2,2)	[[6.0736e+02, 7.7136e+02, 9.0319e+02, ..., 6.4806e+02, 6.2286e+02, 6.2096e+02], [8.7829e+02, 8.7326e+02, 8.8876e+02, ..., 6.5965e+02, 6.3880e+02, 6.4887e+02], [9.0950e+02, 9.0859e+02, 8.9428e+02, ..., 6.6111e+02, 6.3523e+02, 6.4318e+02], [2.3957e+02, 3.3796e+02, 4.1059e+02, ..., 4.6706e+02, 5.4710e+02, 5.4995e+02], [1.0580e+02, 1.2779e+02, 1.4248e+02, ..., 4.5009e+02, 5.5256e+02, 5.4601e+02], [9.5076e+01, 1.1566e+02, 1.1746e+02, ..., 7.4548e+02, 5.3196e+02, 5.3479e+02]],
Maxpool2 (2,3)	[[4.4983e+03, 4.8075e+03, 4.9043e+03, ..., 1.2407e+03, 1.1987e+03, 1.1849e+03],

	[2.3017e+03, 1.8549e+03, 1.7874e+03, ..., 1.2616e+03, 1.2149e+03, 1.2067e+03], [1.7159e+03, 1.6949e+03, 1.7481e+03, ..., 1.2607e+03, 1.2049e+03, 1.1429e+03], ..., ..., [1.1991e+02, 1.2947e+02, 2.3752e+02, ..., 3.5950e+03, 1.0055e+03, 1.0391e+03], [1.3415e+02, 1.3052e+02, 2.1591e+02, ..., 4.8340e+03, 1.0109e+03, 1.0069e+03], [1.3058e+02, 9.3604e+01, 1.7375e+02, ..., 5.6084e+03, 1.1639e+03, 1.0219e+03]], ..., ..., ...
Maxpool2 (2,4)	..., ..., ..., ..., ...
Maxpool2 (2,5)	..., ..., ..., ..., ...
Maxpool2 (2,6)	[[-1.6912e+03, -1.4461e+03, -1.3249e+03, ..., -7.9565e+02, -7.8297e+02, - 7.7677e+02], [-1.0971e+03, -1.1026e+03, -1.1170e+03, ..., -7.9444e+02, - 7.4932e+02, -7.1484e+02], [-1.0376e+03, -1.0476e+03, -1.0915e+03, ..., -7.9387e+02, - 7.4491e+02, -7.0750e+02], ..., ..., [5.2433e+01, 7.0962e+01, 9.7401e+01, ..., -6.7779e+02, -6.3909e+02, - 6.1721e+02], [-1.5555e+01, -1.0087e+01, -3.2697e+01, ..., -8.1073e+02, - 6.2898e+02, -6.1746e+02], [1.0238e+01, 9.5858e-01, -2.8087e+01, ..., -1.1910e+03, - 6.4229e+02, -6.1661e+02]], ..., ..., ...
Maxpool2 (2,7)	[[-2.7214e+02, -2.7202e+02, -2.7938e+02, ..., -1.4310e+02, -1.4035e+02, - 1.3781e+02], [-1.9344e+02, -1.9561e+02, -2.0015e+02, ..., -1.4391e+02, - 1.4027e+02, -1.3855e+02], [-1.9210e+02, -1.9308e+02, -1.9405e+02, ..., -1.4480e+02, - 1.3934e+02, -1.3722e+02], ..., ..., [-7.5118e+00, -8.4832e+00, -1.6156e+01, ..., -6.4932e+01, -1.1135e+02, - 1.1748e+02], [-8.3539e+00, -8.3199e+00, -1.2011e+01, ..., -5.8245e+01, - 1.0808e+02, -1.1456e+02], [-7.4822e+00, -6.8041e+00, -9.9575e+00, ..., -6.4756e+01, - 9.3598e+01, -1.1549e+02]], ..., ..., ...
Maxpool2 (2,8)	[[1.7769e+03, 1.5462e+03, 1.2026e+03, ..., -3.9814e+02, -4.0654e+02, - 3.9491e+02], [-3.1303e+02, -4.5337e+02, -4.8404e+02, ..., -4.0007e+02, - 4.1337e+02, -4.1216e+02], [-5.5939e+02, -5.3998e+02, -5.2873e+02, ..., -4.0468e+02, - 4.2115e+02, -4.3474e+02], ..., ..., ...

	<p>[-8.9262e+01, -9.0695e+01, -8.0166e+01, ..., 9.9184e+01, -3.4730e+02, -3.4226e+02],</p> <p>[-6.6476e+01, -7.5042e+01, -6.6186e+01, ..., 4.1198e+02, -3.5029e+02, -3.4411e+02],</p> <p>[-5.6731e+01, -5.7690e+01, -4.8846e+01, ..., 6.9856e+02, -3.0395e+02, -3.4336e+02]]],</p> <p>...</p>
--	---

Aktivasyon 2. katman çıktılarının matris gösterimi	
Maske	RELU2 çıktı : torch.Size([8,6,2,2])
RELU (2,1)	<p>tensor([[[[1.6295e+02, 1.9634e+02, 5.0156e+01, ..., 6.6065e+01, 5.5107e+01, 5.7020e+01],</p> <p>[8.2959e+01, 8.3910e+01, 7.8342e+01, ..., 6.7365e+01, 5.7649e+01, 6.1038e+01],</p> <p>[9.5033e+01, 1.0068e+02, 9.0132e+01, ..., 6.4144e+01, 4.6575e+01, 4.4289e+01],</p> <p>...</p> <p>...</p> <p>[3.0889e+01, 3.7981e+01, 3.1752e+01, ..., 1.8295e+01, 6.2446e+01, 5.3007e+01],</p> <p>[2.4517e+01, 3.3804e+01, 3.6654e+01, ..., 0.0000e+00, 6.0878e+01, 5.0605e+01],</p> <p>[1.5766e+01, 3.6140e+01, 3.0761e+01, ..., 0.0000e+00, 3.5306e+01, 4.2870e+01]],</p> <p>[1.5766e+01, 3.6140e+01, 3.0761e+01, ..., -2.8614e+02, 3.5306e+01, 4.2870e+01]],</p> <p>...</p>
RELU (2,2)	<p>[[6.0736e+02, 7.7136e+02, 9.0319e+02, ..., 6.4806e+02, 6.2286e+02, 6.2096e+02],</p> <p>[8.7829e+02, 8.7326e+02, 8.8876e+02, ..., 6.5965e+02, 6.3880e+02, 6.4887e+02],</p> <p>[9.0950e+02, 9.0859e+02, 8.9428e+02, ..., 6.6111e+02, 6.3523e+02, 6.4318e+02],</p> <p>...</p> <p>...</p> <p>[2.3957e+02, 3.3796e+02, 4.1059e+02, ..., 4.6706e+02, 5.4710e+02, 5.4995e+02],</p> <p>[1.0580e+02, 1.2779e+02, 1.4248e+02, ..., 4.5009e+02, 5.5256e+02, 5.4601e+02],</p> <p>[9.5076e+01, 1.1566e+02, 1.1746e+02, ..., 7.4548e+02, 5.3196e+02, 5.3479e+02]],</p> <p>...</p>
RELU (2,3)	<p>[[4.4983e+03, 4.8075e+03, 4.9043e+03, ..., 1.2407e+03, 1.1987e+03, 1.1849e+03],</p> <p>[2.3017e+03, 1.8549e+03, 1.7874e+03, ..., 1.2616e+03, 1.2149e+03, 1.2067e+03],</p> <p>[1.7159e+03, 1.6949e+03, 1.7481e+03, ..., 1.2607e+03, 1.2049e+03, 1.1429e+03],</p> <p>...</p> <p>...</p> <p>[1.1991e+02, 1.2947e+02, 2.3752e+02, ..., 3.5950e+03, 1.0055e+03, 1.0391e+03],</p> <p>[1.3415e+02, 1.3052e+02, 2.1591e+02, ..., 4.8340e+03, 1.0109e+03, 1.0069e+03],</p> <p>[1.3058e+02, 9.3604e+01, 1.7375e+02, ..., 5.6084e+03, 1.1639e+03, 1.0219e+03]],</p>

	...
RELU (2,4)	...
	...
RELU (2,5)	...
	...
RELU (2,6)	[[0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00], [0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00], [0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00], ... [5.2433e+01, 7.0962e+01, 9.7401e+01, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00], [0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00], [1.0238e+01, 9.5858e-01, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00],
RELU (2,7)	[[0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00], [0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00], [0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00], ... [0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00], [0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00], [0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00],
RELU (2,8)	[[1.7769e+03, 1.5462e+03, 1.2026e+03, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00], [0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00], [0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 0.0000e+00, 0.0000e+00, 0.0000e+00], ... [0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 9.9184e+01, 0.0000e+00, 0.0000e+00], [0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 4.1198e+02, 0.0000e+00, 0.0000e+00], [0.0000e+00, 0.0000e+00, 0.0000e+00, ..., 6.9856e+02, 0.0000e+00, 0.0000e+00]]]] ...

Konvolüsyon 3. katman maskelerinin matris gösterimi	
Maske	Conv3_kernel : torch.Size([10, 8, 4,4])

Conv3(3,1)	tensor([[[[5.2493e-04, -6.9858e-02, 1.3517e-01, 3.7427e-02], [7.7644e-02, -1.2696e-01, -2.8541e-02, 4.1557e-02], [-6.2381e-02, -1.8372e-03, 5.5946e-02, -5.5637e-02], [-3.4693e-02, -6.9528e-02, 6.5649e-02, -8.0269e-03]]],
	[[[-6.5016e-03, -4.2413e-02, -2.3636e-02, -2.4206e-02], [6.0431e-02, -1.2720e-01, 1.3943e-01, -2.1297e-02], [1.4594e-01, -5.5990e-02, -2.0522e-01, -1.7026e-01], [1.4672e-01, -7.7074e-02, 1.6954e-02, -4.2512e-02]]],
	[[[-2.7278e-02, 1.0649e-01, 1.1785e-01, -1.5737e-01], [2.5094e-01, -2.3979e-01, -7.3627e-01, -1.5577e-01], [2.9896e-01, -8.2542e-01, -3.2387e-01, 2.4933e-02], [-4.0288e-01, 2.4216e-01, 4.8282e-02, 8.9877e-02]]],
	[[[-5.9266e-02, -2.9237e-02, -6.1685e-02, -8.3168e-02], [-3.5847e-02, -6.6811e-02, 2.6454e-02, -3.5646e-02], [6.3931e-02, 3.1426e-02, 7.7382e-02, -2.3069e-03], [-4.8106e-02, -1.2087e-01, -3.7324e-02, -1.1573e-02]]],
	[[[-2.9835e-01, 8.8530e-02, 2.5269e-01, 6.7942e-02], [7.4500e-02, 3.5899e-02, -2.7847e-02, -1.4196e-01], [-8.5719e-02, 1.9127e-01, -2.8233e-01, -2.2636e-01], [-1.4578e-01, -2.3032e-03, 2.8112e-01, 2.3650e-01]]],
	[[[-1.0240e-01, 1.0665e-01, -7.3485e-02, -1.1386e-01], [-1.7284e-01, 1.2040e-02, 2.9327e-01, -8.9081e-05], [-1.3662e-01, -8.0300e-02, -2.6541e-01, -1.8302e-01], [-1.1911e-01, -1.1091e-01, 4.0164e-03, 8.0085e-04]]],
	[[[1.8325e-02, 7.1680e-03, -3.1526e-03, 1.3234e-02], [-1.7600e-02, -2.1112e-02, -3.6204e-02, -6.1322e-03], [-2.3254e-02, -2.3453e-02, 7.6926e-03, -1.0631e-02], [-2.6725e-02, -1.9203e-02, 4.0822e-03, 3.3611e-02]]],
	[[[-1.6202e-01, -1.8233e-02, 1.6243e-02, -6.2961e-02], [-1.7888e-03, -1.1654e-02, -2.3132e-02, -1.1775e-01], [-7.7438e-03, -2.4551e-02, -7.5492e-02, -4.0979e-02], [3.6198e-02, -4.3957e-02, -1.5048e-01, -7.1331e-02]]],
	Conv3(3,2)
[[[-4.9676e-03, 7.1117e-02, 3.6946e-02, -3.9051e-02], [-2.7460e-01, -3.5221e-02, -8.8808e-02, -8.9512e-02], [5.2943e-02, 1.7735e-02, -1.2248e-01, -7.7215e-02], [6.9065e-02, 4.5244e-02, -9.0644e-03, 1.7831e-02]]],	
[[[3.3234e-01, 1.1479e-01, 1.8142e-01, -8.9601e-03], [-5.9429e-01, -4.2788e-01, -2.2718e-02, -1.9801e-01], [-2.6427e-01, -2.4389e-01, -1.3987e-01, -8.6684e-03], [9.5096e-02, -3.5999e-01, 4.9467e-01, -3.4150e-01]]],	
[[[-2.8101e-01, -1.1686e-01, -8.9315e-02, -5.2978e-02], [-2.0822e-01, -6.3298e-02, -5.2187e-02, -8.1420e-02], [-1.2225e-02, -2.1160e-02, -7.5984e-02, -1.9454e-02], [-3.2074e-02, 1.6037e-02, -2.5540e-02, -4.2196e-02]]],	
[[[4.2630e-02, 9.2653e-02, 9.4241e-02, 2.1943e-02], [-2.0369e-01, -3.7541e-01, -1.2220e-01, -1.7666e-04], [2.5370e-01, -1.5248e-01, -5.1786e-02, -1.3801e-01], [6.7153e-02, 4.1339e-02, 2.4899e-01, 2.0111e-01]]],	
[[[2.5899e-02, -1.5519e-02, -1.0215e-01, -5.1179e-02], [3.6755e-02, 6.8417e-02, -3.9529e-02, -4.9822e-02], [6.9517e-02, 6.3325e-02, -2.0752e-02, 7.2885e-03], [-1.0579e-01, -3.5527e-02, 9.6202e-03, 1.2608e-01]]],	
[[[1.4586e-02, 1.1853e-02, -3.2604e-02, 2.5866e-02], [-9.7485e-04, -1.8634e-03, 1.4284e-02, 1.2237e-02],	

	[8.1167e-04, 2.1770e-02, 1.1381e-02, -1.9622e-02], [-2.7037e-02, 4.9521e-04, -1.8434e-02, 3.5803e-02]],
	[[-2.0792e-02, 5.0182e-03, 5.8071e-02, 3.0509e-03], [-2.9184e-02, -4.6148e-02, -8.9363e-02, 1.0752e-03], [5.8410e-02, -1.9969e-02, 3.7516e-02, 1.1719e-02], [1.2927e-02, -2.7045e-02, -1.5681e-02, -4.3861e-03]]],
Conv3(3,3)	tensor([[[-5.0247e-03, -1.1606e-03, -4.9450e-02, -2.8416e-02], [1.3930e-03, 4.6840e-03, 1.8522e-02, 1.9722e-02], [6.0029e-02, 3.2922e-02, 4.4081e-02, -4.8691e-02], [-4.4204e-02, -5.8023e-02, 1.9445e-02, 3.4626e-02]],
	[[7.7961e-03, 1.5123e-03, -1.0443e-02, 3.6323e-02], [-6.9917e-03, 6.4341e-03, 4.2650e-02, -3.5323e-02], [1.4378e-02, -3.4885e-02, 2.0656e-02, -6.9232e-02], [-5.3799e-02, 3.0601e-02, 8.8258e-02, 7.5415e-02]],
	[[1.2779e-02, -2.0305e-01, 7.4668e-02, 1.1387e-01], [-2.5079e-01, 6.2021e-02, -3.9211e-02, -1.4561e-01], [8.9918e-03, -4.3642e-02, -2.7448e-01, -2.2712e-02], [1.0424e-01, -4.6312e-02, -2.4876e-02, 6.2425e-02]],
	[[-2.6347e-02, -4.7257e-02, -2.8997e-02, -8.2448e-03], [-3.1454e-02, -1.3292e-02, -5.8432e-02, -5.0194e-04], [-2.7363e-02, 2.9020e-02, -1.4802e-03, -2.3004e-02], [-3.6830e-02, -6.5230e-03, 5.5002e-03, -8.0133e-03]],
	[[-3.2493e-02, -8.2345e-02, -1.5970e-01, -4.5682e-02], [5.1089e-02, 2.0533e-02, 6.0284e-02, -9.7975e-03], [1.0561e-01, 1.3736e-01, -9.0820e-02, -7.8108e-02], [-1.3359e-01, -7.9453e-03, 6.7661e-02, 2.0705e-01]],
	[[7.1548e-02, 1.0546e-01, 4.5494e-02, 6.2474e-02], [-5.3430e-02, 1.4037e-02, 4.5714e-02, -2.4838e-03], [-6.3137e-02, 8.8549e-03, -7.5969e-02, -6.8745e-02], [-7.1541e-02, -6.4070e-02, -4.8401e-02, -1.3007e-03]],
	[[-2.9657e-02, 2.7589e-02, -6.7856e-03, 1.7083e-02], [1.7527e-02, -8.9093e-03, -5.0052e-03, 2.7467e-02], [-6.0011e-03, 1.7078e-02, 2.4705e-02, -8.5599e-03], [-1.6097e-02, 8.5417e-03, 9.8501e-04, 1.5923e-03]],
	[[-1.6115e-02, -1.7359e-02, 1.4773e-02, 3.4469e-02], [1.8931e-02, 8.5587e-03, 6.1650e-03, -2.6581e-04], [-2.6663e-02, -2.7406e-02, 2.4939e-02, -6.1747e-03], [-9.7874e-03, -2.5715e-02, -1.2794e-03, -5.3656e-02]]],
Conv3(3,,4)	tensor([[[-0.0326, -0.0250, -0.0044, -0.0086], [-0.0369, 0.0010, -0.0245, -0.0146], [0.0238, 0.0212, 0.0058, 0.0412], [-0.0081, 0.0057, 0.0094, 0.0329]],
	[[-0.0428, 0.0116, -0.0120, -0.0411], [-0.0324, -0.0027, -0.0325, -0.0155], [-0.0190, -0.0351, -0.0118, 0.0320], [0.0388, -0.0047, 0.0354, 0.0367]],
	[[-0.0717, -0.0025, -0.1000, -0.0558], [-0.0791, -0.0009, -0.0642, -0.0759], [-0.0928, 0.0916, 0.0542, -0.0224], [-0.0937, 0.0338, 0.0624, -0.1222]],
	[[-0.0057, -0.0229, 0.0158, 0.0232], [0.0103, -0.0059, -0.0300, 0.0193], [0.0179, -0.0051, -0.0245, 0.0004], [-0.0061, -0.0245, -0.0124, -0.0371]],
	[[-0.0881, -0.0038, -0.0260, -0.0868], [0.0217, -0.0447, -0.0408, -0.0735], [0.0487, 0.0436, 0.0068, 0.0579], [0.0639, -0.0011, 0.0331, -0.0213]],

	[[0.0289, -0.0184, 0.0295, 0.0075], [-0.0156, -0.0060, 0.0200, -0.0238], [-0.0378, -0.0332, -0.0346, -0.0408], [-0.0205, -0.0250, 0.0188, -0.0458]],
	[[[-0.0008, 0.0070, 0.0117, -0.0272], [-0.0253, 0.0275, 0.0110, -0.0146], [-0.0106, 0.0226, -0.0200, 0.0041], [-0.0084, 0.0122, 0.0242, 0.0237]],
	[[[-0.0225, -0.0176, -0.0335, -0.0062], [-0.0336, 0.0069, 0.0167, 0.0287], [-0.0404, -0.0021, -0.0168, 0.0260], [0.0062, -0.0132, -0.0240, -0.0016]]]
Conv3(3,5)	tensor([[[[0.0427, -0.1293, -0.0488, 0.0029], [-0.0162, -0.0509, -0.0688, 0.2238], [-0.0166, 0.0475, -0.0156, 0.0848], [0.0017, -0.0241, -0.0626, 0.0349]],
	[[[0.0556, -0.0976, -0.0454, 0.1467], [-0.1271, -0.1353, 0.0978, 0.3142], [-0.0278, -0.1780, 0.0863, 0.0304], [0.0480, -0.0555, -0.0123, 0.0857]],
	[[[0.0059, -0.1883, -0.0600, -0.1231], [0.2187, -0.0098, -0.1601, -0.4122], [0.3304, 0.2477, -0.4500, -0.2900], [-0.3574, 0.0509, -0.3323, 0.2271]],
	[[[0.0644, 0.0083, -0.0511, -0.0683], [-0.0058, -0.0480, -0.0148, -0.0673], [-0.0672, -0.1374, 0.0648, -0.0683], [-0.0725, -0.1394, -0.0532, 0.0433]],
	[[[-0.0419, -0.3619, -0.3554, -0.0912], [0.1040, 0.1436, 0.1417, 0.2021], [0.1266, 0.0813, 0.1220, 0.0639], [-0.0394, -0.0773, -0.1179, -0.0179]],
	[[[0.1257, -0.0236, 0.1346, -0.0031], [-0.0456, -0.1510, 0.0560, 0.1259], [-0.0847, -0.0695, -0.0036, 0.0309], [-0.0938, -0.1906, -0.1787, -0.2754]],
	[[[-0.0062, -0.0119, 0.0129, -0.0137], [0.0014, -0.0079, 0.0115, 0.0123], [0.0376, 0.0091, -0.0680, -0.0113], [-0.0069, 0.0106, -0.0480, 0.0038]],
	[[[-0.0318, -0.0545, -0.0159, -0.0320], [-0.0408, -0.0058, -0.0006, 0.0023], [-0.0545, 0.0051, 0.0083, -0.0549], [-0.1015, -0.0761, -0.0498, -0.1327]]],
Conv3(3,6)	tensor([[[[-2.4009e-02, 1.6642e-02, -3.7511e-02, 2.2305e-02], [-4.2905e-02, 1.1547e-02, 1.1702e-01, 7.4380e-03], [6.1616e-03, -1.6248e-01, 3.6586e-02, -1.2472e-01], [2.3706e-02, -4.0844e-02, 1.3682e-01, -6.7327e-02]],
	[[[-6.9341e-03, -4.6341e-03, 6.8870e-02, 3.5806e-02], [3.0049e-02, 9.3094e-03, 1.0950e-01, -7.9702e-04], [-9.0820e-02, -1.4579e-01, -2.1960e-02, -1.2192e-01], [1.1007e-01, -2.3468e-02, -3.7250e-03, 9.8352e-02]],
	[[[2.0379e-01, 1.6599e-01, -2.8706e-01, 1.7429e-01], [-7.1481e-02, -6.0293e-02, -2.2392e-01, -2.0604e-01], [-2.8036e-01, -3.1459e-01, -2.5215e-01, 1.8519e-01], [3.4009e-01, -6.8340e-01, 2.9498e-01, -2.7441e-02]],
	[[[-1.8266e-02, -4.2005e-02, -7.9884e-02, -1.1240e-02], [-2.6615e-02, -3.4687e-02, -3.8518e-02, -5.4668e-02],

	[-7.9027e-02, -6.6136e-02, -8.0376e-03, -3.4901e-02], [-1.7209e-02, -2.1613e-02, -2.7913e-02, 3.1891e-02]],
	[[[-1.5969e-01, 8.1723e-02, -5.2741e-02, -1.5279e-03], [-1.9832e-01, 9.7865e-02, 2.4376e-01, 1.6632e-02], [-2.2161e-01, -2.3743e-01, -3.1462e-01, -1.6588e-01], [8.7623e-02, 2.6127e-01, -8.0776e-02, 1.8421e-01]],
	[[[-1.1097e-01, 8.7027e-04, 4.2201e-02, 2.8447e-02], [-5.0325e-02, 7.6867e-02, 3.0579e-02, -1.4968e-01], [-1.0328e-01, -1.1043e-01, -1.7936e-01, -1.5429e-01], [-7.1279e-02, 4.2731e-02, -5.0716e-02, -4.1867e-02]],
	[[1.3020e-03, 5.4698e-03, 6.2744e-03, -2.9939e-02], [-5.1113e-03, -7.9945e-03, -2.1403e-03, 1.8488e-02], [1.1548e-03, -2.6567e-02, -2.7622e-02, 1.8254e-02], [-1.3974e-02, -8.8078e-03, 1.5311e-02, 2.9737e-02]],
	[[[-2.7965e-02, -6.5026e-03, -1.7818e-02, -4.3000e-02], [-1.3832e-02, 4.6318e-04, -7.9930e-02, -1.0465e-01], [4.7529e-02, -7.8074e-02, 7.3491e-02, -5.8588e-02], [5.4430e-03, -6.5797e-02, -8.9101e-02, -1.4656e-01]],
Conv3(3,7)	tensor([[[[-9.5769e-02, -7.1026e-02, -1.5433e-01, -6.8618e-03], [1.8584e-01, -1.6470e-01, -2.4132e-01, 9.0358e-02], [1.3596e-01, -3.6972e-02, -2.8100e-02, -1.2516e-02], [8.6903e-02, -2.2123e-01, 3.6144e-03, 4.2992e-02]],
	[[1.9810e-01, -2.5012e-02, -2.2180e-01, -5.2453e-03], [6.7218e-02, -2.3311e-01, -2.0129e-01, -4.5792e-02], [2.5093e-01, -1.2757e-01, -6.5473e-02, 1.2551e-01], [4.4150e-02, 4.5276e-02, 2.0162e-01, -1.2002e-01]],
	[[[-1.2876e-01, -3.2997e-01, -5.4368e-01, 6.6969e-02], [-6.7679e-01, 8.2343e-01, 6.9750e-02, -2.1628e-01], [-2.5212e-01, 1.8562e-01, 5.7003e-01, -5.8599e-01], [9.4891e-02, -3.4285e-01, -2.2369e-02, -1.4287e-02]],
	[[2.0552e-02, 3.2166e-02, 8.9730e-02, 2.4236e-02], [-3.3981e-02, 3.5911e-04, -1.0090e-01, -4.0843e-02], [2.7131e-02, -1.0319e-01, -2.4320e-01, -8.4847e-02], [-4.3677e-02, -5.8326e-02, -4.7431e-02, -7.0203e-02]],
	[[[-2.1066e-01, 1.3144e-01, -3.0953e-01, -3.0996e-01], [-1.6933e-02, -1.8619e-02, 3.2249e-02, -1.8540e-01], [-1.0034e-01, 2.7511e-01, 1.5591e-01, 2.6875e-01], [-7.5101e-02, -1.1503e-01, 2.9013e-01, 2.1383e-02]],
	[[1.9741e-02, -2.1365e-01, 1.5217e-01, 8.3692e-02], [-1.0810e-02, -2.3530e-01, 1.0578e-01, 9.7367e-02], [-2.9603e-01, -2.5898e-01, 8.0805e-02, -1.8103e-02], [-1.0096e-02, -3.2560e-02, -1.8768e-01, -2.5775e-01]],
	[[[-3.9640e-02, -3.4032e-03, 2.4088e-02, 3.8676e-04], [8.8666e-03, 1.1758e-02, -9.2892e-03, 9.3857e-04], [-1.4327e-02, -8.7190e-03, 6.0082e-04, -1.5475e-02], [-7.2264e-03, -1.1093e-02, -5.4509e-03, -1.0230e-02]],
	[[[-8.7975e-02, -5.1545e-02, -8.3984e-02, -1.1260e-01], [-1.3494e-01, -1.3297e-01, -6.3084e-02, -4.6507e-02], [-1.7131e-01, -1.3476e-01, 8.3439e-04, -4.1192e-02], [4.1716e-02, 2.0447e-02, -8.6384e-02, -1.0933e-01]],
Conv3(3,8)	tensor([[[[-4.5257e-02, 1.4784e-02, 1.9777e-02, -2.7285e-02], [-8.3760e-02, -3.4832e-02, 5.5716e-02, 1.5144e-01], [7.6296e-02, -1.1487e-01, -1.0627e-01, -9.0475e-02], [-1.1251e-01, 3.0721e-02, 1.4326e-03, -4.0808e-02]]
	[[[-4.8957e-02, -8.9340e-02, 4.3706e-02, 2.4152e-01], [1.7844e-02, -6.3016e-03, 6.3435e-02, -9.4867e-02], [6.5410e-02, -2.5045e-01, -1.8803e-01, 6.2175e-02], [5.3191e-02, 9.9147e-02, -3.9600e-02, 9.8503e-03]],

	[[1.5530e-01, -8.5607e-02, 9.8353e-03, -6.4291e-01], [4.0807e-01, -8.8154e-02, 2.5817e-01, -1.4512e-01], [-6.5319e-01, -2.2861e-01, -3.1843e-01, 3.9988e-01], [-5.9622e-02, 3.6409e-01, -1.8617e-01, -1.3755e-01]],
	[[6.6681e-03, -1.5339e-01, -6.2942e-02, 3.4116e-02], [-9.9039e-02, -4.9143e-02, 8.7577e-03, -7.4770e-02], [1.1639e-03, -1.4064e-01, 1.9222e-02, -4.0270e-02], [1.3292e-01, 5.4942e-02, -1.1926e-02, -3.9174e-02]],
	[[-1.3529e-02, 1.1201e-01, 1.0523e-01, -1.2459e-02], [-2.6761e-01, -9.6352e-02, 3.3063e-01, 8.8854e-02], [2.2451e-01, -1.9987e-01, -4.0216e-01, -3.2292e-01], [-1.6687e-01, 1.7270e-01, 2.5999e-01, -1.8307e-01]],
	[[-5.5780e-02, -4.6319e-02, -3.2907e-02, 2.7068e-02], [1.3098e-02, -1.0457e-01, 1.9483e-02, -2.2231e-01], [-4.2132e-02, -1.8771e-01, -4.0310e-02, -6.3785e-02], [8.5198e-02, 5.1509e-02, 9.0450e-02, 1.3133e-01]],
	[[6.2935e-03, -2.7362e-02, 3.9711e-03, -6.0163e-03], [-2.4526e-02, -5.7843e-03, -1.3115e-04, 9.4863e-03], [-6.8465e-03, -1.1726e-02, 1.1434e-02, -1.2117e-03], [4.9237e-03, 2.8968e-03, 3.4649e-02, 1.0035e-02]],
	[[-8.8864e-02, -6.9044e-02, -6.7175e-02, -6.3770e-02], [2.2298e-02, 8.3761e-03, 3.2280e-02, -3.6906e-02], [2.2747e-02, -1.0359e-02, -4.4401e-05, -9.3023e-02], [-1.1911e-01, -8.8486e-02, -5.0259e-02, -7.0593e-02]],
Conv3(3,9)	tensor([[[1.9468e-01, 9.4218e-02, -1.2822e-01, -1.6460e-01], [-1.0584e-01, -6.5339e-02, -4.8817e-02, 6.5240e-02], [6.7509e-02, -1.5468e-01, 8.0455e-02, 1.4525e-01], [4.9095e-02, -1.9922e-01, -1.8558e-01, -2.0781e-01]],
	[[2.6918e-01, 7.5484e-03, -5.7506e-02, -1.9679e-01], [-7.6387e-02, -1.6666e-01, 1.4174e-01, -9.1192e-02], [1.2930e-01, 1.4133e-02, 8.9656e-02, 5.1560e-03], [7.2973e-02, -2.4556e-01, -2.2564e-01, -2.2902e-01]],
	[[2.5452e-01, -1.5293e-01, 4.9847e-02, -1.2403e-01], [-9.9060e-01, 4.8411e-01, -8.1549e-02, -1.6890e-01], [1.5153e-01, 1.8424e-01, 1.9395e-01, 4.2340e-01], [-9.8081e-02, -3.5211e-01, -2.6828e-01, 8.4373e-02]],
	[[-1.1903e-01, -4.5608e-04, 2.0663e-02, -1.5603e-02], [4.8926e-03, -3.6316e-02, -1.6354e-02, -2.3081e-01], [-3.8221e-02, -1.7854e-01, -1.8807e-01, -1.9242e-01], [-6.6944e-02, -2.2884e-01, -1.1177e-01, -1.6396e-01]],
	[[4.9812e-01, 1.9848e-01, -2.5617e-01, -3.1593e-01], [-4.1388e-01, -2.7882e-01, 9.2050e-02, 1.9394e-01], [2.8163e-01, -1.2040e-01, 2.5953e-01, 2.4164e-01], [1.2289e-02, 5.5640e-03, -7.3470e-01, -3.6094e-01]],
	[[-1.5069e-01, -1.2245e-01, -3.4833e-01, -1.9623e-02], [-1.7837e-01, -8.8310e-02, -1.6673e-01, 5.6936e-02], [7.2775e-02, -7.8803e-03, 6.6183e-04, -4.7179e-03], [-6.4876e-02, -2.9498e-01, -1.7045e-01, -2.4913e-01]],
	[[-3.1476e-02, -4.2677e-02, -3.5860e-02, 4.0383e-02], [-9.8567e-03, -7.2670e-03, -5.6991e-03, 2.5276e-02], [-1.2400e-02, 2.4788e-02, -2.5094e-02, -2.2748e-02], [-2.0395e-02, 1.1875e-02, 2.5240e-02, -3.1431e-02]],
	[[-2.5539e-03, -3.1875e-02, 3.2459e-02, 9.8706e-03], [-2.4985e-02, -6.2585e-02, -2.0484e-02, -9.2937e-02], [3.5099e-03, 8.4229e-02, 1.4125e-02, -3.9197e-02], [-3.0292e-02, 3.1007e-02, -4.4928e-02, -1.3239e-01]],
	tensor([[[0.0524, -0.0735, -0.2040, 0.0361], [0.0056, -0.1928, -0.0882, -0.1189],

	[0.0092, -0.0190, 0.1218, 0.0838], [-0.0868, 0.0014, 0.0818, -0.1139]],
	[[0.1701, 0.0296, -0.0226, 0.0625], [-0.2115, -0.2231, -0.2669, -0.2479], [-0.0037, -0.0197, -0.0074, 0.0109], [-0.0634, 0.1784, 0.1055, -0.0935]],
	[[[-0.4684, -0.1998, -0.3056, -0.1872], [0.1212, 0.0955, -0.8388, 0.6857], [-0.0369, 0.6210, -0.4730, 0.0633], [-0.2534, 0.2073, -0.1440, -0.2354]],
	[[[-0.0584, -0.0838, 0.0844, -0.0051], [-0.0808, -0.0762, -0.0828, -0.0733], [-0.1694, -0.1720, -0.0261, -0.1619], [-0.1688, -0.0248, -0.1164, -0.0821]],
	[[0.2863, 0.0152, -0.3649, -0.1560], [-0.0593, -0.1381, -0.4451, -0.4744], [0.0771, 0.3537, 0.1745, 0.0758], [-0.4129, 0.2263, 0.1624, 0.1107]],
	[[0.0116, 0.2066, 0.1236, -0.1141], [-0.2354, -0.1566, -0.1115, -0.0719], [-0.2792, -0.1731, -0.1373, -0.1137], [-0.0988, 0.0532, -0.1229, -0.1721]],
	[[[-0.0024, 0.0072, -0.0460, 0.0023], [-0.0351, 0.0122, -0.0049, 0.0033], [0.0223, 0.0181, -0.0062, 0.0079], [-0.0137, -0.0343, -0.0315, -0.0323]],
	[[0.0321, 0.0293, -0.1186, -0.0970], [-0.1334, -0.0288, -0.1484, -0.0497], [-0.1599, -0.0551, -0.1015, -0.0693], [-0.0430, -0.0368, 0.0257, 0.0204]],

Konvolüsyon 3. katman çıktılarının matris gösterimi	
Maske	Conv3 çıktı : torch.Size([10, 8, 4,4])
Conv3(3,1)	tensor([[[[-1.3002e+03, -1.3967e+03, -1.2040e+03, ..., -6.9029e+02, - 8.0700e+02, -8.6715e+02], - [-1.3133e+03, -1.2629e+03, -1.0888e+03, ..., -7.2707e+02, - 8.0184e+02, -8.3673e+02], - [-1.2842e+03, -1.2239e+03, -1.1701e+03, ..., -8.0127e+02, - 8.6674e+02, -8.8504e+02], - ..., - ..., - ..., - ..., - [-5.0963e+03, -6.6255e+03, -7.0985e+03, ..., -8.7853e+03, -3.1030e+03, - 9.2093e+01], - [-9.4338e+02, -1.0571e+03, -8.3803e+02, ..., -7.7928e+03, - 5.0378e+03, -8.7599e+02], - [-6.0897e+02, -2.9222e+02, -3.4501e+00, ..., -7.4961e+03, - 7.3777e+03, -1.8594e+03]], - ..., -
Conv3(3,2)	[[6.5226e+02, 7.9785e+02, 4.4282e+02, ..., -8.7597e+02, -7.3127e+02, - 8.1989e+02], - [-1.0248e+03, -1.2109e+03, -1.2550e+03, ..., -8.8121e+02, - 7.5285e+02, -8.0633e+02], -

	[-1.1595e+03, -1.2415e+03, -1.2619e+03, ..., -8.4530e+02, - 7.9990e+02, -8.3300e+02], ..., ..., ..., ..., ..., ..., [-6.3094e+03, -7.1900e+03, -8.0220e+03, ..., -3.1914e+03, -7.3835e+03, - 4.6616e+03], [-9.6577e+01, -1.1946e+03, -2.3523e+03, ..., -2.1721e+03, - 6.0396e+03, -6.2868e+03], [-1.5265e+03, -1.1274e+03, -8.5962e+02, ..., -2.0781e+03, - 4.9270e+03, -7.9849e+03]], ..., ...
Conv3(3,3)	[[[-6.2653e+02, -6.9939e+02, -7.3575e+02, ..., -2.7980e+02, -2.8056e+02, - 3.0638e+02], [-4.7389e+02, -4.3380e+02, -4.4536e+02, ..., -2.8324e+02, - 2.7382e+02, -2.9215e+02], [-4.1396e+02, -4.5617e+02, -4.9266e+02, ..., -2.9133e+02, - 2.9960e+02, -3.1344e+02], ..., ..., ..., ..., ..., ..., [-2.0191e+03, -2.0972e+03, -1.7279e+03, ..., -2.8087e+03, -2.6243e+03, - 9.5778e+02], [-9.3695e+02, -1.2007e+03, -1.5490e+03, ..., -2.4440e+03, - 3.3072e+03, -1.1302e+03], [-2.4408e+02, -5.5258e+02, -7.8086e+02, ..., -2.3116e+03, - 3.4108e+03, -1.3520e+03]], ..., ...
Conv3(3,4)	..., ..., ..., ..., ..., ..., ..., ..., ...
Conv3(3,5)	..., ..., ..., ..., ..., ..., ..., ...
Conv3(3,6)	[[[-2.5810e+03, -2.1665e+03, -1.4747e+03, ..., -5.8015e+02, -5.7428e+02, - 5.6945e+02], [-8.4791e+02, -8.9359e+02, -8.0571e+02, ..., -5.6516e+02, - 6.1532e+02, -5.9233e+02], [-8.9286e+02, -8.0790e+02, -8.5614e+02, ..., -5.3540e+02, - 6.0538e+02, -5.8245e+02], ..., ..., ..., ..., ..., ..., [-1.6072e+03, -2.5290e+03, -3.2846e+03, ..., -3.2079e+03, -1.2322e+03, - 7.5471e+02], ...

	<p>[-5.2472e+02, 5.0460e+00, -1.2836e+02, ..., -2.2187e+03, -3.0470e+03, -797e+02],</p> <p>[-4.9321e+03, -4.4512e+03, -3.1571e+03, ..., -1.2198e+03, -5.8677e+03, 9.2093e+01],</p> <p>[-2.9222e+02, -3.4501e+00, 6.1705e+02, ..., -4.3317e+02, -3.6401e+03, -8.7599e+02]],</p> <p>....</p>
Maxpool3 (2,2)	<p>[[7.9785e+02, 4.4282e+02, -2.7931e+02, ..., -8.0076e+02, -8.4071e+02, -7.3127e+02],</p> <p>[-1.1595e+03, -1.2419e+03, -1.1281e+03, ..., -7.4122e+02, -7.3901e+02, -7.7253e+02],</p> <p>[-1.1666e+03, -1.1779e+03, -1.1663e+03, ..., -8.6851e+02, -7.6934e+02, -8.0050e+02],</p> <p>....</p> <p>....</p> <p>....</p> <p>....</p> <p>[1.1447e+03, 1.1882e+03, 9.8150e+02, ..., -1.8442e+03, -2.8775e+03, -8.6051e+02],</p> <p>[-2.7009e+03, -1.4847e+03, -9.2905e+01, ..., -7.3109e+02, -2.7664e+03, -3.3806e+03],</p> <p>[-9.6577e+01, -5.3959e+02, -8.8547e+02, ..., -3.5250e+02, -2.0781e+03, -4.9270e+03]],</p> <p>....</p>
Maxpool3 (2,3)	<p>[[-4.3380e+02, -4.4536e+02, -4.1406e+02, ..., -2.7538e+02, -2.6774e+02, -2.7382e+02],</p> <p>[-4.1396e+02, -4.2893e+02, -3.9725e+02, ..., -2.6254e+02, -2.7561e+02, -2.7617e+02],</p> <p>[-4.3445e+02, -4.3002e+02, -4.3748e+02, ..., -3.0124e+02, -2.9267e+02, -2.7821e+02],</p> <p>....</p> <p>....</p> <p>....</p> <p>....</p> <p>....</p> <p>[4.0872e+02, 2.8858e+02, -1.3988e+01, ..., -3.3637e+02, -2.3414e+03, 2.6254e+01],</p> <p>[-8.6820e+02, -4.8746e+02, -1.8259e+01, ..., -4.7519e+01, -1.6293e+03, -6.6607e+02],</p> <p>[-2.4408e+02, -7.8031e+02, -9.0522e+02, ..., -4.3902e+01, -4.7479e+02, -1.1302e+03]],</p> <p>....</p>
Maxpool3 (2,4)	<p>....</p> <p>....</p> <p>....</p>
Maxpool3 (2,5)	<p>....</p> <p>....</p> <p>....</p>
Maxpool3 (2,6)	<p>[[-8.4791e+02, -8.0571e+02, -8.1133e+02, ..., -5.0813e+02, -5.3778e+02, -5.6945e+02],</p> <p>[-8.0790e+02, -8.5614e+02, -8.2159e+02, ..., -5.3327e+02, -5.3540e+02, -4.9876e+02],</p> <p>[-8.0732e+02, -7.8842e+02, -7.0899e+02, ..., -5.1556e+02, -5.0675e+02, -4.6818e+02],</p> <p>....</p> <p>....</p> <p>....</p>

	<p>...</p> <p>[2.4317e+01, 6.8001e+02, 1.9662e+02, ..., -2.3683e+03, -1.4786e+03, -6.3350e+02],</p> <p>[-1.6072e+03, -3.2846e+03, -2.8824e+03, ..., -1.5068e+03, -2.5048e+03, -7.5471e+02],</p> <p>[-6.7823e+02, -7.4314e+02, -1.2698e+03, ..., -2.3945e+02, -2.9850e+03, -6.1526e+02]],</p> <p>...</p>
Maxpool3 (2,7)	<p>[[2.2155e+03, 2.2500e+03, 2.2184e+03, ..., 1.5744e+03, 1.4710e+03, 1.4845e+03],</p> <p>[1.9295e+03, 1.9497e+03, 1.7675e+03, ..., 1.5156e+03, 1.5056e+03, 1.4951e+03],</p> <p>[1.7390e+03, 1.7990e+03, 1.8523e+03, ..., 1.5138e+03, 1.4576e+03, 1.4565e+03],</p> <p>...</p> <p>...</p> <p>...</p> <p>...</p> <p>...</p> <p>[7.6454e+02, 1.1309e+03, 1.1609e+03, ..., 6.8842e+02, 1.9085e+03, 7.7047e+02],</p> <p>[-1.7197e+03, -1.4398e+03, -9.2677e+02, ..., -5.3539e+02, 1.5540e+03, -1.7309e+03],</p> <p>[-1.7001e+03, -2.6103e+03, -9.5233e+02, ..., 3.8105e+01, 1.7325e+03, -7.4292e+02]],</p> <p>...</p>
Maxpool3 (2,8)	<p>[[[-1.9490e+02, 1.4167e+02, 3.7462e+01, ..., 2.1735e+02, 2.8622e+02, 1.9976e+02],</p> <p>[2.1909e+01, 1.1401e+02, 1.4134e+02, ..., 2.9879e+02, 2.8723e+02, 1.6097e+02],</p> <p>[-3.9264e+01, -4.5688e+01, -1.4360e+01, ..., 2.5631e+02, 2.5769e+02, 8.8260e+01],</p> <p>...</p> <p>...</p> <p>...</p> <p>...</p> <p>...</p> <p>[-4.6704e+02, -4.9169e+02, -4.1790e+01, ..., -6.4119e+02, -7.2004e+02, 1.5859e+02],</p> <p>[-1.0562e+03, 7.5752e+01, 2.2402e+02, ..., -3.3420e+02, -6.8335e+03, 5.8470e+02],</p> <p>[-9.1430e+02, -2.4109e+03, -4.6876e+03, ..., 1.4962e+02, 3.4061e+03, 4.5642e+02]]],</p> <p>...</p>

Senaryo 4 sonucunda oluşan ağız ve göz ağırlık matrisleri	
agiz ağırlıklar torch.Size([128, 9990])	<p>tensor([[4.3458e-05, 2.4755e-03, -1.9520e-03, ..., -1.7473e-03, -3.0986e-03, 1.5034e-03],</p> <p>[-1.8673e-03, 3.0306e-03, -5.1187e-04, ..., -2.4505e-03, -3.0601e-03, 1.3790e-02],</p> <p>[-2.1532e-03, 6.5274e-05, -2.4392e-03, ..., -2.3357e-03, 4.4448e-03, -1.1944e-02],</p> <p>...</p> <p>[-2.0874e-03, -7.8884e-04, 2.6814e-03, ..., 1.1254e-03, -8.9197e-03, 1.6763e-02],</p>

	[-2.3131e-03, -2.9796e-03, -2.8082e-03, ..., -1.6967e-03, 1.6867e-03, 3.0036e-03], [-3.3579e-03, 2.8891e-03, 2.8328e-03, ..., 6.0240e-04, -6.7362e-04, 3.4955e-03]], requires_grad=True)
göz ağırlıklar torch.Size([128, 9990])	tensor([[-0.0002, 0.0002, 0.0016, ..., -0.0006, 0.0049, 0.0112], [0.0010, -0.0020, 0.0027, ..., -0.0043, -0.0011, -0.0020], [-0.0022, -0.0026, 0.0011, ..., -0.0010, -0.0123, 0.0076], ..., [0.0018, -0.0031, -0.0017, ..., 0.0003, -0.0056, -0.0043], [-0.0004, -0.0031, 0.0023, ..., -0.0022, 0.0007, 0.0008], [0.0009, -0.0022, 0.0010, ..., 0.0085, 0.0010, -0.0035]], requires_grad=True)
Senaryo 4 sonucunda elde edilen tahmini ağız ve göz değerleri	
Prediction_ağız	tensor([[-912.2397, 911.1942]],
Prediction_göz	tensor([[-819.5477, 908.8188]],



KİŞİSEL YAYIN VE ESERLER

Kır B., Öz C., & Gülbağ A. (2001). Yapay Sinir Ağlarında Negative Correlation Learning Metodu Kullanarak Optik Karakter Tanıma. Elektrik-Elektronik Bilgisayar Sempozyumu (FEEB 2011), Elazığ, 105-109.

Kır B., Öz C., & Gülbağ A. (2012, April). Leaf recognition using K-NN classification algorithm. In 2012 20th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.

Kır B., Öz C., & Gülbağ A. (2013, November). The Application of optical character recognition for mobile device via artificial neural networks with negative correlation learning algorithm. In 2013 International Conference on Electronics, Computer and Computation (ICECCO), 2013, Elazığ, Türkiye.

Kır Savaş B., and Becerikli Y, Behavior-Based Driver Fatigue Detection System with Deep Belief Network, Neural Computing and Applications (gönderildi).

Kır Savaş B., and Becerikli Y, BSD-ConNN: a driver fatigue behavioral state detection algorithm based on convolutional neural network, turkish journal of electrical engineering & computer sciences (Ocak 2021 gönderildi)

Kır Savaş B., and Becerikli Y., Real Time Driver Fatigue Detection System Based on Multi-Task ConNN, IEEE Access, 2020, **8**(2020), 12491–12498.

Kır Savaş B., and Becerikli Y., Real Time Driver Fatigue Detection Based on SVM Algorithm, 6th International Conference on Control Engineering & Information Technology, CEIT), 25-27 Ekim 2018, İstanbul, 2018.

Kır Savaş B., Becerikli Y., Development of Driver Fatigue Detection System By Using Video Images, Akıllı Sistemlerde Yenilikler Ve Uygulamaları (ASYU 2017), Alanya-Antalya, Türkiye, 5-7 Ekim 2017.

Kır B., Altuncu M. A., & Sahin Ş. (2013, May). FPGA based implementation of CORDIC using different number format. In 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAECE) (pp. 444-448). IEEE.

Kır B., Kurt M., & Urhan, O., Local binary pattern based fast digital image stabilization. *IEEE Signal Processing Letters*, 2014, **22**(3), 341-345.

Pehlivanoğlu M. K., **Savaş B. K.**, & Duru N. (2015, May). LSB based steganography over video files using Koblitz's Method. In 2015 23rd Signal Processing and Communications Applications Conference (SIU) (pp. 1034-1037). IEEE.

Savaş B. K., Duman O., & Şahin, S., Smart Devices' Contribution to the Development of Virtual Museology in Turkey. *International Journal of Computer Applications*, 2016, **1**(6), 21-24.

Savaş B. K., İlkin S., Hangişi S., & Şahin S., Gölge Tespitinde Kullanılan Bayes Sınıflandırma, Otsu Bölütleme ve Histogram Dağılımı Yöntemlerinin Karşılaştırılması. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 2017, **5**(2), 345-355.

Şahin S., & **Savaş B. K.**, Hardware implementation of generic CORDIC algorithm on FPGA. *Sakarya Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 2017, **21**(1), 16-25.

Ünal H. B., Vural E., **Savaş B. K.**, & Becerikli, Y. (2020, October). Fruit Recognition and Classification with Deep Learning Support on Embedded System (fruitnet). In 2020 Innovations in Intelligent Systems and Applications Conference (ASYU) (pp. 1-5). IEEE.



ÖZGEÇMİŞ

Burcu KIR SAVAŞ ilk ve orta öğrenimini Gülsüm Sami Kefeli İlköğretim Okulu'nda, lise öğrenimini Samsun Anadolu Lisesi'nde tamamladı. Dumlupınar Üniversitesi'nde Bilgisayar Mühendisliği bölümü ve çift anadal olarak ta Elektrik-Elektronik Mühendisliği bölümlerini okudu. 2009 yılında Bilgisayar Mühendisliği Bölümünden, 2010 yılında Elektrik-Elektronik Mühendisliği bölümünden mezun oldu. 2010 yılı içinde Sakarya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar ve Bilişim Mühendisliği Anabilim Dalı'nda başladığı Yüksek Lisans eğitimini 2012 yılında tamamladı. 2012 yılından beri Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Bilgisayar Mühendisliği Anabilim Dalı'nda doktora eğitimine devam etmektedir. Aynı zamanda 2010 yılından itibaren Kocaeli Üniversitesi Bilgisayar Mühendisliği bölümünde Araştırma Görevlisi olarak çalışmaktadır.