



**T.C.
DÜZCE ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**

**EL HAREKETLERİNİN DERİN ÖĞRENME TABANLI
TANINMASI VE ARTIRILMIŞ GERÇEKLİK UYGULAMASI
GELİŞTİRME**

OSMAN GÜLER

**DOKTORA TEZİ
ELEKTRİK-ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI**

**DANIŞMAN
PROF. DR. İBRAHİM YÜCEDAĞ**

DÜZCE, 2021

T.C.
DÜZCE ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

EL HAREKETLERİNİN DERİN ÖĞRENME TABANLI
TANINMASI VE ARTIRILMIŞ GERÇEKLİK UYGULAMASI
GELİŞTİRME

Osman GÜLER tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Lisansüstü Eğitim Enstitüsü Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Prof. Dr. İbrahim YÜCEDAĞ

Düzce Üniversitesi

Jüri Üyeleri

Prof. Dr. İbrahim YÜCEDAĞ

Düzce Üniversitesi

Prof. Dr. Uğur GÜVENÇ

Düzce Üniversitesi

Doç. Dr. Alper BAYRAK

Abant İzzet Baysal Üniversitesi

Doç. Dr. Abdullah Talha KABAKUŞ

Düzce Üniversitesi

Doç. Dr. Nihat DALDAL

Abant İzzet Baysal Üniversitesi

Tez Savunma Tarihi: 28/07/2021

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

28 Temmuz 2021

Osman GÜLER



TEŐEKKÖR

Doktora öđrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı çok değerli hocam Prof. Dr. İbrahim YÜCEDAĞ'a en içten dileklerle teşekkür ederim. Tez çalışması süresince yönlendirmeleri ile çalışmalarımıza olumlu katkılar sağlayan tik üyesi hocalarım Prof. Dr. Uđur GÜVENÇ ve Doç. Dr. Alper BAYRAK hocama teşekkür ederim.

Bu çalışma boyunca yardımlarını ve desteklerini esirgemeyen dostum ve çalışma arkadaşım Dr. Serkan SAVAŐ'a sonsuz teşekkürlerimi sunarım.

Tez çalışmam süresince hep yanımda olan, her zaman bana destek olan sevgili eşim ve aileme sonsuz teşekkürlerimi sunarım.

28 Temmuz 2021

Osman GÜLER

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ	vii
ÇİZELGE LİSTESİ	x
KISALTMALAR.....	xi
ÖZET	xii
ABSTRACT	xiii
EXTENDED ABSTRACT	xiv
1. GİRİŞ.....	1
1.1. ÇALIŞMANIN KAPSAMI	2
1.2. LİTERATÜRE KATKI.....	3
1.3. TEZİN YAPISI.....	3
2. ARTIRILMIŞ GERÇEKLİK.....	5
2.1. ARTIRILMIŞ GERÇEKLİK TÜRLERİ	6
2.1.1. İşaretleyici Tabanlı Artırılmış Gerçeklik.....	6
2.1.2. İşaretleyici Olmayan Artırılmış Gerçeklik	7
2.1.2.1. Konum Tabanlı Artırılmış Gerçeklik.....	7
2.1.2.2. Yansıtma Tabanlı Artırılmış Gerçeklik	8
2.1.2.3. Anahat Tabanlı Artırılmış Gerçeklik.....	8
2.1.2.4. Üstüne Bindirme Tabanlı Artırılmış Gerçeklik	9
2.2. ARTIRILMIŞ GERÇEKLİK İÇİN KULLANILAN ÇEVRE BİRİMLERİ... 9	
2.2.1. Donanım Altyapısı.....	9
2.2.2. Yazılım Altyapısı	10
2.3. ARTIRILMIŞ GERÇEKLİK VE EL İLE ETKİLEŞİM ÜZERİNE YAPILAN ÇALIŞMALAR.....	10
3. DERİN ÖĞRENME	30
3.1. EVRİŞİMSEL SİNİR AĞLARI.....	32
3.1.1. Evrişim Katmanı	33
3.1.2. Havuzlama Katmanı	34
3.1.3. Tam Bağlı Katman	35
3.1.4. DropOut	35
3.1.5. Aktivasyon fonksiyonu.....	36
3.1.5.1. Sigmoid Fonksiyonu	36
3.1.5.2. Hiperbolik Tanjant (tanh) Fonksiyonu.....	37
3.1.5.3. Doğrultulmuş Lineer Birim (Rectified Linear Unit - ReLU) Fonksiyonu.....	37
3.1.5.4. Sızıntı (Leaky) ReLU Fonksiyonu.....	37
3.1.5.5. Basamak Fonksiyonu	38
3.1.5.6. Doğrusal Fonksiyon	38
3.1.5.7. Swish Fonksiyon.....	39
3.1.6. Sınıflandırma katmanı.....	39
3.2. KAPSÜL AĞLAR VE DİNAMİK YÖNLENDİRME	39
3.3. DERİN ÖĞRENME MODELLERİ.....	41
3.3.1. LeNet	41

3.3.2. AlexNet	42
3.3.3. ZFNet.....	42
3.3.4. ResNet.....	43
3.3.5. VGGNet.....	44
3.3.6. GoogLeNet	45
3.3.7. DenseNet.....	48
3.3.8. MobileNet.....	48
3.3.9. InceptionV2/InceptionV3	50
3.3.10. CapsNet	50
4. ÖNERİLEN MODEL VE VERİSETİ.....	52
4.1. VERİ SETİ	57
5. TEST SONUÇLARI.....	60
6. ARTIRILMIŞ GERÇEKLİK UYGULAMASI.....	72
7. SONUÇ	79
8. KAYNAKLAR.....	83
9. EKLER	93
9.1. EK 1: HG14 VERİ SETİ BAĞLANTI ADRESİ.....	93
9.2. EK 2: GENERATE_ARUCO.PY KODLAMA DOSYASI	94
9.3. EK 3: EL TANIMA.PY KODLAMA DOSYASI.....	96
ÖZGEÇMİŞ.....	99

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1. AG çalışma sistemi.	5
Şekil 2.2. AG Türleri.	6
Şekil 2.3. İşaretleyici ve eksenler.	7
Şekil 2.4. Konum Tabanlı AG.	7
Şekil 2.5. Yansıtma Tabanlı AG.	8
Şekil 2.6. Anahat Tabanlı AG.	8
Şekil 2.7. Üstüne Bindirme Tabanlı AG.	9
Şekil 2.8. İşaretleyicili Eldiven ve AG Satranç oyunu	11
Şekil 2.9. EnhancedDesk AG tabanlı masa sistemi [.	12
Şekil 2.10. Dış ortam AG sistemi	12
Şekil 2.11. AG Kontrol paneli kutucuğu	13
Şekil 2.12. HMC giriş görüntüsü	13
Şekil 2.13. VITA AG sistemi ve etkileşim araçları	14
Şekil 2.14. AG menü çubuğu	14
Şekil 2.15. Şehir planlama gösterimi	15
Şekil 2.16. Artırılmış köpük	15
Şekil 2.17. Sanal basketbol oyunu.	16
Şekil 2.18. HandyAR nesne seçimi	16
Şekil 2.19. Parmaklara takılan kızılötesi hedefler ve blok tutma	17
Şekil 2.20. Masaüstü AG sistemi ve el ile tutulan ekran.	17
Şekil 2.21. Jenga ve balık tutma oyunu	18
Şekil 2.22. Avuçiçi tanıma AG uygulaması	18
Şekil 2.23. MagicCup AG giriş aracı.	19
Şekil 2.24. İşaretleyicisiz el tanıma sistemi	19
Şekil 2.25. AG sanal nesne döndürme ve boyutlandırma	20
Şekil 2.26. El tanıma ile montaj işlemi	20
Şekil 2.27. El ile etkileşim metodu işlem basamakları	21
Şekil 2.28. AG sistemi ve uygulaması	21
Şekil 2.29. Miragetable etkileşimli masaüstü AG uygulaması	22
Şekil 2.30. Parmakucu tanıma ve sistem prototipi	23
Şekil 2.31. AG sistemi çalışması	23
Şekil 2.32. Etkileşimli AG uygulaması şeması	24
Şekil 2.33. AG kart oyunu ve nesne etkileşimi	24
Şekil 2.34. Çoklu el izleme sistemi.	25
Şekil 2.35. AR araba boyutlandırma animasyonu	25
Şekil 2.36. Avuçiçi tespiti	26
Şekil 2.37. El tanıma ve sanal nesne gösterimi	26
Şekil 2.38. El ile tutulan işaretleyici, Serbest el etkileşimi, Tuş takımı denetleyicisi	27
Şekil 2.39. RemoAct AG sistemi	27
Şekil 2.40. AG Satranç oyunu	28
Şekil 2.41. AG ev aletleri kontrolü blok diyagramı	28
Şekil 2.42. Yüzey çizimi ve hava çizimi yöntemleri	29
Şekil 3.1. Evrişimsel sinir ağı örnek model.	33
Şekil 3.2. Evrişim işlemi.	33
Şekil 3.3. Havuzlama işlemi ile boyut azaltma.	34
Şekil 3.4. Tam bağlı katman yapısı.	35

Şekil 3.5. Dropout katmanı.	36
Şekil 3.6. Sigmoid Fonksiyonu	36
Şekil 3.7. Hiperbolik Tanjant Fonksiyonu	37
Şekil 3.8. ReLU Fonksiyonu	37
Şekil 3.9. Sızıntı ReLU Fonksiyonu	38
Şekil 3.10. Basamak Fonksiyonu	38
Şekil 3.11. Doğrusal Fonksiyonu	39
Şekil 3.12. Swish Fonksiyonu	39
Şekil 3.13. Kapsül ağı örnek model.	40
Şekil 3.14. LeNet mimarisi	42
Şekil 3.15. ALexNet mimarisi.	42
Şekil 3.16. ZFNet mimarisi	43
Şekil 3.17. Kalıntı modülü,34 katmanlı Klasik CNN modeli ile ResNet mimarisi	44
Şekil 3.18. VGGNet mimarisi	45
Şekil 3.19. Inception modülü	46
Şekil 3.20. Boyutu düşürme sonrası inception modülü	46
Şekil 3.21. GoogLeNet mimarisi	47
Şekil 3.22. DenseNet mimarisi	48
Şekil 3.23. Ayrılabilir ve noktasal evrişim yapısı	49
Şekil 3.24. MobileNet mimarisi	49
Şekil 3.25. InceptionV2 mimarisi.	50
Şekil 3.26. CapsNet mimarisi	51
Şekil 4.1. Önerilen CNN & Kapsül ağı modeli.	52
Şekil 4.2. Önerilen CNN & Kapsül ağı yapay sinir ağı modeli.	57
Şekil 4.3. HG14- El Hareketi Veri seti.	58
Şekil 4.4. Fashion-MNIST veri seti.	59
Şekil 4.5. CIFAR-10 veri seti.	59
Şekil 5.1. VGG16 Modeli CIFAR-10 veri seti eğitim sonuçları.	61
Şekil 5.2. ResNet50 Modeli CIFAR-10 veri seti eğitim sonuçları.	61
Şekil 5.3. DenseNet121 Modeli CIFAR-10 veri seti eğitim sonuçları.	62
Şekil 5.4. MobilNet Modeli CIFAR-10 veri seti eğitim sonuçları.	62
Şekil 5.5. InceptionV3 Modeli CIFAR-10 veri seti eğitim sonuçları.	62
Şekil 5.6. CapsNet Modeli CIFAR-10 veri seti eğitim sonuçları.	62
Şekil 5.7. CCNN Modeli CIFAR-10 veri seti eğitim sonuçları.	63
Şekil 5.8. VGG16 Modeli Fashion-MNIST veri seti eğitim sonuçları.	64
Şekil 5.9. ResNet50 Modeli Fashion-MNIST veri seti eğitim sonuçları.	64
Şekil 5.10. DenseNet121 Modeli Fashion-MNIST veri seti eğitim sonuçları.	65
Şekil 5.11. MobileNet Modeli Fashion-MNIST veri seti eğitim sonuçları.	65
Şekil 5.12. InceptionV3 Modeli Fashion-MNIST veri seti eğitim sonuçları.	65
Şekil 5.13. CapsNet Modeli Fashion-MNIST veri seti eğitim sonuçları.	65
Şekil 5.14. CCNN Modeli Fashion-MNIST veri seti eğitim sonuçları.	66
Şekil 5.15 VGG16 Modeli HG14 veri seti eğitim sonuçları.	67
Şekil 5.16 ResNet50 Modeli HG14 veri seti eğitim sonuçları.	68
Şekil 5.17 DenseNet121 Modeli HG14 veri seti eğitim sonuçları.	68
Şekil 5.18 MobileNet Modeli HG14 veri seti eğitim sonuçları.	68
Şekil 5.19 InceptionV3 Modeli HG14 veri seti eğitim sonuçları.	68
Şekil 5.20 CapsNet Modeli HG14 veri seti eğitim sonuçları.	69
Şekil 5.21 CCNN Modeli HG14 veri seti eğitim sonuçları.	69
Şekil 6.1. AG Uygulaması Akış Diyagramı.	73
Şekil 6.2. ArUco 1 numaralı Referans İşaretçiler.	73

Şekil 6.3. ArUco Referans İşaretçi Tespiti Akış Diyagramı.....	74
Şekil 6.4. ArUco 4x4 1-2-3-4 numaralı Referans İşaretçiler.....	75
Şekil 6.5. AG uygulaması ekran görüntüsü.	76
Şekil 6.6. AG uygulaması 1. Video gösterimi.	77
Şekil 6.7. AG uygulaması 2. Video gösterimi.	77
Şekil 6.8. AG uygulaması program sonlandırma.....	78



ÇİZELGE LİSTESİ

Sayfa No

Çizelge 3.1. El tanıma üzerine derin öğrenme modelleri ile yapılan çalışmalar.	31
Çizelge 4.1. Önerilen CCNN modeli mimarisi.	54
Çizelge 4.2. CCNN V1 mimarisi.	54
Çizelge 4.3. CCNN V2 mimarisi.	54
Çizelge 4.4. CCNN V3 mimarisi.	55
Çizelge 4.5. CCNN V4 mimarisi.	55
Çizelge 4.6. CCNN mimarileri eğitim doğruluk oranları.	55
Çizelge 5.1. CIFAR-10 veri seti test sonuçları.	61
Çizelge 5.2. CIFAR-10 veri seti üzerine yapılan çalışmalar.	63
Çizelge 5.3. Fashion-MNIST veri seti test sonuçları.	64
Çizelge 5.4. Fashion-MNIST veri seti üzerine yapılan çalışmalar.	66
Çizelge 5.5. HG14 veri seti test sonuçları.	67
Çizelge 5.6. CIFAR-10, Fashion-MNIST ve HG14 veri seti eğitim zamanı sonuçları..	70
Çizelge 5.7. Tek görüntü tahmin süreleri.	70
Çizelge 5.8. HG14 veri seti eğitime girmeyen veri test doğruluk oranı.	71

KISALTMALAR

2B	2 Boyutlu
3B	3 Boyutlu
6DOF	Six Degrees Of Freedom
AG	Artırılmış Gerçeklik
CAD	Computer Aided Design
CCNN	Convolutional Capsule Neural Network
CNC	Computer Numerical Control
CNN	Convolutional Neural Network
CSS	Cascading Style Sheets
DDR4	Double Data Rate 4
FC	Fully Connected
GB	Gigabyte
GPS	Global Positioning System
HG14	Handgesture14
HMC	Head Mounted Camera
HMD	Helmet Mounted Display
HTML	Hyper Text Markup Language
ILSVRC	Imagenet Large Scale Visual Recognition Challenge
İBE	İnsan Bilgisayar Etkileşimi
İOAG	İşaretleyici Olmayan Artırılmış Gerçeklik
İTAG	İşaretleyici Tabanlı Artırılmış Gerçeklik
LSTM	Long Short-Term Memory
MS	Milisaniye
OPENCV	Open Source Computer Vision
RAM	Random Access Memory
ReLU	Rectified Lineer Unit
RFID	Radio Frekanslı Tanımlama
RGB	Red-Green-Blue
RNN	Recurrent Neural Network
SDK	Software Development Kit
SLAM	Simultane Lokalizasyon Ve Haritalama Teknolojisi
SN	Saniye
SSD	Solid-State Disk
VGG	Visual Geometry Group
VITA	Visual Interaction Tool For Archaeology
WLAN	Wireless Local Area Network

ÖZET

EL HAREKETLERİNİN DERİN ÖĞRENME TABANLI TANINMASI VE ARTIRILMIŞ GERÇEKLİK UYGULAMASI GELİŞTİRME

Osman GÜLER

Düzce Üniversitesi

Lisansüstü Eğitim Enstitüsü, Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim
Dalı

Doktora Tezi

Danışman: Prof. Dr. İbrahim YÜCEDAĞ

Temmuz 2021, 98 sayfa

Son yıllarda gelişen teknoloji ile birlikte insan-bilgisayar etkileşimi çalışmaları yaygınlaşmıştır. Özellikle artırılmış gerçeklik uygulamaları insan bilgisayar etkileşimi alanında yaygınlaşan bir çalışma alanı olmuştur. Görüntü işleme alanında ise derin öğrenme yöntemlerinin kullanılmasıyla birlikte son yıllarda önemli gelişmeler yaşanmıştır. Nesne tanıma, nesne sınıflandırma, yüz tanıma, hareket tanıma ve el hareketi tanıma uygulamaları, derin öğrenme yöntemleri sayesinde artık günlük hayatta kullanılan cihazlarda etkin bir şekilde kullanılmaktadır. Bu çalışmada, insan-bilgisayar etkileşiminde önemli bir rol oynayan artırılmış gerçeklik teknolojileri araştırılmıştır. Derin öğrenme ile el hareketi tanıma modeli geliştirilmiş ve el hareketleri ile kontrol edilebilen bir artırılmış gerçeklik uygulaması tasarlanmıştır. El hareketi tanıma ve sınıflandırması için evrişimli sinir ağı ile kapsül ağ algoritması kullanılarak hibrit bir model oluşturulmuştur. 14 farklı el hareketi içeren toplam 14000 görüntüden oluşan HandGesture14 (HG14) adlı bir veri seti oluşturulmuştur. Önerilen modelin nesne tanımadaki başarısını ölçmek için HG14, Fashion-MNIST ve CIFAR-10 veri setleri üzerinde derin öğrenme önerilen hibrit model ile VGG16, ResNet50, DenseNet121, MobileNet, InceptionV3 ve CapsNet modelleri kullanılarak eğitimler gerçekleştirilmiştir. Eğitimlerin sonuçları karşılaştırılmış ve doğruluk oranları değerlendirilmiştir. Önerilen hibrit model, HG14 veri setinde %90, Fashion-MNIST veri setinde %93.88 ve CIFAR-10 veri setinde %81.42 ile doğruluk oranlarına ulaşmıştır.

Anahtar sözcükler: Artırılmış gerçeklik, Derin öğrenme, El hareketi tanıma, İnsan bilgisayar etkileşimi, Yapay zekâ.

ABSTRACT

DEEP LEARNING BASED RECOGNITION OF HAND GESTURES AND DEVELOPMENT OF AUGMENTED REALITY APPLICATION

Osman GÜLER

Düzce University

Institute of Graduate Studies, Department of Electrical-Electronic and Computer
Engineering

Doctoral Thesis

Supervisor: Prof. Dr. İbrahim YÜCEDAĞ

July 2021, 98 pages

In recent years, human-computer interaction has become widespread with the developing technology. In particular, augmented reality applications have become a widespread field of study in the field of human-computer interaction. In the field of image processing, there have been significant developments in recent years with the use of deep learning methods. Object recognition, object classification, face recognition, gesture recognition and hand gesture recognition applications developed with deep learning methods are now effectively used in devices used in daily life. In this study, augmented reality technologies that play an important role in human-computer interaction were investigated. A hand gesture recognition model has been developed with deep learning and an augmented reality application that can be controlled by hand gestures has been designed. A hybrid model was created for hand gesture recognition and classification by using convolutional neural network and capsule network algorithm. A dataset called HandGesture14 (HG14) was created, consisting of a total of 14000 images containing 14 different hand gestures. In order to measure the success of the proposed model in object recognition, trainings were carried out on the HG14, Fashion-MNIST and CIFAR-10 datasets using the hybrid model proposed for deep learning and VGG16, ResNet50, DenseNet121, MobileNet, InceptionV3 and CapsNet models. The results of the trainings were compared and the accuracy rates were evaluated. The proposed hybrid model achieved accuracy rates of %90 in the HG14 dataset, %93.88 in the Fashion-MNIST dataset and %81.42 in the CIFAR-10 dataset.

Keywords: Artificial intelligence, Augmented reality, Deep learning, Hand gesture recognition, Human computer interaction.

EXTENDED ABSTRACT

DEEP LEARNING BASED RECOGNITION OF HAND GESTURES AND DEVELOPMENT OF AUGMENTED REALITY APPLICATION

Osman GÜLER

Düzce University

Institute of Graduate Studies, Department of Electrical-Electronic and Computer
Engineering

Doctoral Thesis

Supervisor: Prof. Dr. İbrahim YÜCEDAĞ

July 2021, 98 pages

1. INTRODUCTION

In this study, a hand gesture recognition model with deep learning is proposed to interact using the natural hand gesture recognition method in augmented reality applications. The difference of the proposed model from the studies in the literature is that it is a combination of convolutional neural networks and capsule neural networks. Within the scope of the study, a hand gesture dataset consisting of 14000 images captured from the first person viewpoint was developed to train with hand gesture recognition. On the developed hand gestures dataset, trainings were made with the proposed deep learning model and hand gesture recognition was performed. As a result, in augmented reality applications, using the weights obtained as a result of deep learning, interaction with hand movements is provided.

2. MATERIAL AND METHODS

A dataset containing 14 different hand gestures was developed for the training and testing processes of the proposed deep learning model in this study. For each gesture, 1000 images were captured from 17 different people from different camera angles, using different colored backgrounds. The dataset consists of 14000 colored images.

In order to compare the results of the proposed hybrid model in the HG14, Fashion-MNIST and CIFAR-10 datasets, trainings were conducted with the models in the literature, VGG16, ResNet50, DenseNet121, MobileNet, InceptionV3 and CapsNet. The results obtained from the trainings were compared. In order to be standard in all trainings, test parameters were taken as Epoch number 50 and Batchsize 10. Since random parameters were used, the same test was repeated 30 times for each model and dataset, the averages were taken, and the standard deviations were calculated in order to

increase the reliability of the training results.

Within the scope of the thesis, a marker based augmented reality application was developed using the Python programming language and OpenCV support. ArUco markers were used as reference markers. In the augmented reality application, the weights obtained as a result of deep learning trainings were used to provide transition between videos with hand movements.

3. RESULTS AND DISCUSSIONS

To test and compare the success of the proposed hybrid model in image recognition, 30 training sessions were conducted with VGG16, ResNet50, DenseNet121, MobileNet, InceptionV3 and CapsNet models on the HG14, Fashion-MNIST and CIFAR-10 datasets and their averages were calculated. According to the average results of the trainings, the proposed model achieved the accuracy rates with %81.42 in the CIFAR-10 dataset, %93.88 in the Fashion-MNIST dataset, and %90 in the HG14 dataset. The accuracy rates achieved by the VGG16 model were %71.24 in the CIFAR-10 dataset, %92.14 in the Fashion-MNIST dataset, and %83.93 in the HG14 dataset. The accuracy rates achieved by the ResNet50 model are %56.46 in the CIFAR-10 dataset, %87.51 in the Fashion-MNIST dataset, and %30.32 in the HG14 dataset. The accuracy rates achieved by the DenseNet121 model were %65.60 in the CIFAR-10 dataset, %89.78 in the Fashion-MNIST dataset, and %72.86 in the HG14 dataset. The accuracy rates achieved by the InceptionV3 model were %75.00 in the CIFAR-10 dataset, %91.10 in the Fashion-MNIST dataset, and %91.46 in the HG14 dataset. The accuracy rates achieved by the CapsNet model were %66.48 in the CIFAR-10 dataset, %90.59 in the Fashion-MNIST dataset, and %64.00 in the HG14 dataset. We can say that our proposed model is successful in image recognition and classification. The proposed model differs from other models with its convolutional and capsule neural network structure. It can be said that the proposed model structure is successful because it combines the success of the Convolutional Neural Network (CNN) model in object feature extraction and classification processes, and the success of the CNN model in learning images taken from different angles and light environments with the capsule network.

4. CONCLUSION AND OUTLOOK

In this study, a data set called HG14, consisting of 14 different hand gestures, was

created to be used in augmented reality applications especially for human-computer interaction. The difference of the dataset we developed from other datasets in the literature is that it consists of images taken directly with the camera, not with special depth or infrared cameras. Hand gesture images from a first-person view can be used specifically for use with augmented reality applications and wearable technologies. A hybrid model was created using deep learning architectures CNN and capsule networks to recognize hand gestures. Due to its high success in image processing, images are first trained on the CNN network and the new values obtained are given as input values to the capsule network, which is more successful in different angular images, and the training is completed. The training was conducted on the proposed model and the VGG16, ResNet50, DenseNet121, MobileNet, InceptionV3 and CapsNet models, as well as the HG14, Fashion-MNIST and CIFAR-10 datasets. Comparing the results of the training, it is seen that the proposed hybrid model achieved the accuracy rates of %90 in the HG14 dataset, %93.88 in the Fashion-MNIST dataset, and %81.42 in the CIFAR-10 dataset. Considering the test results obtained, it can be said that the proposed deep learning model is successful.

1. GİRİŞ

İnsan-Bilgisayar Etkileşimi (İBE), kullanıcı ile bilgisayar arasında teknolojinin daha kullanılabilir olmasını sağlayan arayüzler ve uygulamalar geliştirmeyi amaçlayan çalışma alanıdır. İBE, hem insan hem bilgisayar faktörü içerdiği için bilişim teknolojileri, yazılım, tasarım, insan psikolojisi ve insan davranışları vb. alanlarla ilişkili olduğu için disiplinler arası bir çalışma alanıdır. İBE alanında çalışanları tasarımcılar ve araştırmacılar olmak üzere iki çalışma dalında inceleyebiliriz. İBE alanında çalışan tasarımcılar, yeni teknoloji tasarımı, arayüz geliştirme kısmında görev alırken, araştırmacılar ise insanların bu teknolojiler ve bilgisayarlar ile etkileşimini, yeni etkileşim yöntem ve tekniklerini, kullanılan teknolojilerin kullanılabilirlik ve verimliliğini araştırırlar.

Teknoloji alanında yaşanan gelişmeler ile birlikte İBE alanında da yeni etkileşim yöntemleri ve teknolojileri geliştirilmiştir. 1980'li yıllarda basit ofis programları, diyalog kutuları ve hata mesajları ile başlayan İBE çalışmaları, o yıllardan günümüze kadar internetin gelişmesi, mobil ve taşınabilir aygıtların geliştirilmesi, dokunmatik ekranların, görüntü, hareket ve his algılama sensörlerinin kullanılması ile yaygınlaşmıştır. Günümüzde en çok arayüz tasarımı yapılan İBE çalışma alanları mobil aygıtlar, dokunmatik ekranlar, sesli komut işleme, insan hareketi hesaplama, görüntü işleme, sensörler, ve giyilebilir teknolojiler kullanılarak geliştirilen etkileşimli sistemlerdir.

İnsan hareketi tanıma ve el hareketi ile kontrol sistemleri, insan-bilgisayar etkileşimini artırmak için bilgisayarlı görme ve makine öğrenmesi alanında çalışan araştırmacılar için önemli bir çalışma konusu olmuştur [1]. Hareket tanıma çalışmalarının amacı ilk olarak belirli hareketlerin tanımlanması ve devamında bu hareketler ile bir cihazı kontrol eden ya da bilgi ileten sistemler oluşturmaktır [2]. El tanıma sistemlerinin bilgisayarlı sistemler, ara yüz kontrolü, akıllı araç kontrolü, sanal gerçeklik uygulamaları, bilgisayar oyunları, robotik sistemler, işaret dili tanıma, multimedya sistemleri vb. alanlarda çok çeşitli uygulamaları vardır [2-4].

İBE alanında geliştirilen arayüzler, eğitim, sağlık, eğlence, endüstri, askeri, turistik,

iletiřim, robotik vb. alanlarda yaygın olarak kullanılmaktadır. Eđitim alanında yeni teknolojiler kullanarak etkileřimli ve kullanıcı tarafından kontrol edilebilen eđitim materyalleri tasarlanmaktadır. Sađlık sektööründe kullanıcının günlük olarak nabız, tansiyon, kalp atıřı, řeker vb. verilerini takip etmesini sađlayan sistemler, uzaktan ve robotik sistemler kullanarak ameliyatlar olmasını sađlayan sistemler geliřtirilmiřtir. Eđlence sektööründe kullanıcı hareketlerini tanıyan dijital oyunlar ve sanal ortamlar tasarlanmıřtır. Endüstri alanında son yıllarda yařanan geliřmeler ile birlikte tüm iřleyiř dijital ortamlarda takip ve kontrol edilebilmektedir. Askeri alanda özellikle silahlı eđitim, savunma ve saldırma sistemleri simülasyonlar ile verilmektedir. Turistik alanda özellikle müze gezileri sanal ortamlarda yapılmaktadır. İletiřim alanında iřaret dili tanıma, dil çevirme sistemleri ile insanlar arasında köprü kurulmaktadır. Robotik alanlarda ise birçok sistem hareket ve sesli kontrol ile kullanıcılar tarafından kontrol edilmektedir. İBE alanında geliřtirilen arayüzler hayatımızın her alanında etkin olarak kullanılmaya bařlamıřtır.

El hareketi tanımlama iřleminde kullanılan yöntemler temel olarak Sensör Tabanlı ve Görüř Tabanlı (Dođal) olmak üzere ikiye ayrılır [5]. Sensör Tabanlı sistemlerde el ve parmak hareketlerini tanımlamak için özel olarak geliřtirilmiř, üzerinde sensör veya iřaretleyiciler bulunan eldivenler kullanılmaktadır. Kullanılan donanımın pahalı olması ve her türlü ortamda kullanılamaması bu sistemin bir dezavantajıdır. Görüř tabanlı sistemlerde ise sadece bir kamera kullanılarak, yazılımsal ya da donanımsal olarak tanımlanan bir yapay görme sistemi ile el hareketi tanıma iřlemi gerçekleştirilmektedir. Görüntülerden el hareketi tespitinde ellerin řekli, ten rengi farklılıđı, arka plan karmařıklıđı, ortamın ıřık řiddeti, kamera açısı vb. gürültüler dikkate alındıđında, el tanıma uygulamaları için kararlı ve hızlı çalıřan bir sistem geliřtirilmesinin önemi artmaktadır.

1.1. ÇALIřMANIN KAPSAMI

Bu tez çalıřmasında, derin öđrenme tabanlı el hareketleri ile kontrol edilebilen etkileřimli bir artırılmıř gerçeklik (AG) uygulaması tasarlanmıřtır. Yapılacak AG çalıřmasında referans iřaretleyici sistem kullanılmıř ve dođal el tanıma yöntemi ile etkileřim sađlanmıřtır. Dođal el tanıma için derin öđrenme modelleri kullanılmıřtır. Derin öđrenme modelleri ile el hareketi tanıma yöntemi için evriřimsel sinir ađları(Convolutional Neural Network-CNN) ile kapsül ađlar ve dinamik yönlendirme

algoritması kullanılarak evrişimsel kapsül sinir ağı (Convolutional Capsule Neural Network-CCNN) adı verilen hibrit bir model oluşturulmuştur. HandGesture14(HG14) adı verilen 14 farklı el hareketinden oluşan bir veri seti oluşturulmuş ve el hareketlerinin sınıflandırılması ve tanınması üzerine çalışılmıştır. CNN, kapsül ağı ve dinamik yönlendirme mimarileri hakkında bilgi verilerek, oluşturulan hibrit model, kullanılan yöntem, yazılım ve donanımlar, veri seti ve özellikleri belirtildikten sonra elde edilen sonuçlar değerlendirilmektedir.

1.2. LİTERATÜRE KATKI

Tez çalışmasında AG uygulamalarında kullanılmak üzere derin öğrenme tabanlı el tanıma ile kontrol edilebilen bir AG uygulaması geliştirilmiştir. El hareketi tanıma için derin öğrenme modelleri kullanılarak hibrit bir model önerilmiştir. Önerilen model evrişimsel sinir ağı ile kapsül ağlar yapısının birleşiminden oluşmaktadır. Evrişimsel Kapsül Sinir Ağı (Convolutional Capsule Neural Network-CCNN) adı verilen hibrit bir model ile literatüre yeni bir derin öğrenme modeli önerisi ile katkı sağlanmıştır. Önerilen model görüntü işleme, nesne sınıflandırma vb. alanlarda derin öğrenme ile çalışacak olan araştırmacılara örnek olacaktır. Çalışma kapsamında el hareketi tanıma için 14 farklı el hareketini içeren ve 14000 el hareketi görüntüsünden oluşan HandGesture14 (HG14) adı verilen bir veri seti oluşturulmuş ve literatüre katılmıştır. El ve hareket tanıma üzerine çalışma yapacak araştırmacıların kullanımını ile literatürde önemli bir yer alacağı düşünülmektedir.

1.3. TEZİN YAPISI

Tez çalışması aşağıda verilen bölümler şeklinde organize edilmiştir.

Tezin birinci bölümünde insan bilgisayar etkileşimi ve el hareketi tanıma üzerine genel bilgiler verilmiştir. Daha sonra çalışmanın amacı ve kapsamı ile literatüre olan katkısından bahsedilmiştir.

Tezin ikinci bölümünde AG, AG türleri, AG uygulamalarında ihtiyaç duyulan çevre birimleri, donanımlar ve yazılımlar ile literatürde AG ile yapılan el tanıma çalışmalarından bahsedilmiştir.

Tezin üçüncü bölümünde derin öğrenme, önerilen modelde kullanılan CNN ve kapsül ağlar yapısı, derin öğrenme ile el tanıma üzerine yapılan çalışmalar ve literatürde yer alan derin öğrenme modelleri açıklanmıştır.

Tezin dördüncü bölümünde CCNN adında yeni bir hibrit derin öğrenme modeli önerilmiş, modelin yapısı ve özellikleri anlatılmıştır. HG14 adında el hareketi veri seti geliştirilmiş ve geliştirilen veri seti açıklanmıştır.

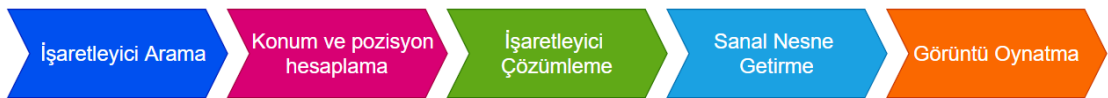
Tezin beşinci bölümünde literatürde yer alan derin öğrenme modelleri ve önerilen CCNN model ile HG14, CIFAR-10 ve Fashion-MNIST veri setleri üzerine yapılan testlerin sonuçlar açıklanmıştır.

Tezin altıncı bölümünde derin öğrenme modeli ve veri seti ile yapılan eğitimler sonucu elde edilen ağırlıklar kullanılarak geliştirilen, el hareketi ile kontrol edilebilen AG uygulaması, geliştirilme adımları ve kullanılan yazılımlar anlatılmıştır.

Tezin yedinci bölümünde ise çalışmanın sonuçlarından bahsedilmiş, çalışmanın literatüre katkısı ve gelecekte yapılabilecek çalışmalara öneriler yer almıştır.

2. ARTIRILMIŞ GERÇEKLİK

Artırılmış gerçeklik; gerçek dünya görüntüsü üzerine çeşitli görüntü işleme ve nesne tanıma yöntemleri kullanılarak dijital ortamda hazırlanmış sanal nesnelerin gösterildiği gerçeklik ortamıdır [6]. Gerçek nesnelerin görüntüsü üzerine dijital ortamda hazırlanan video, ses, animasyon vb. sanal nesnelerin yerleştirilerek, zenginleştirilen canlı, doğrudan veya dolaylı fiziksel görüntüdür. AG teknolojisi, dijital ortamda hazırlanan görüntü, grafik, video, ses, yazı ve küresel konumlama sistemi (Global Positioning System-GPS) gibi sanal içeriklerin kamera yardımı ile alınan gerçek dünya görüntüsü üzerinde önceden tanımlanan hedef noktalarda gerçek zamanlı olarak görüntülenebilmesine olanak sağlar. AG teknolojisi sanal ortam ile gerçek ortamı birleştirdiği için, gerçek dünyanın zenginleştirilmesini, daha faydalı ve işlevsel hale getirilmesini sağlamaktadır [7]. AG teknolojisinde kullanılan dijital nesneler 3 boyutlu (3B) nesneler başta olmak üzere video, animasyon, ses, görüntü, yazı vb. dosyalardan oluşturulabilmektedir. AG teknolojisi kullanılarak oluşturulan sahnelerde genel olarak gerçek görüntü ve sanal bilgi birlikte verilir, sistem eş zamanlı olarak çalışır ve tasarlanan sistem etkileşimli sistemdir, oluşturulan sahneler 3B ortamda görüntülenir [8]. Şekil 2.1 'de AG uygulamalarının çalışma sistemine ait akış diyagramı verilmiştir.

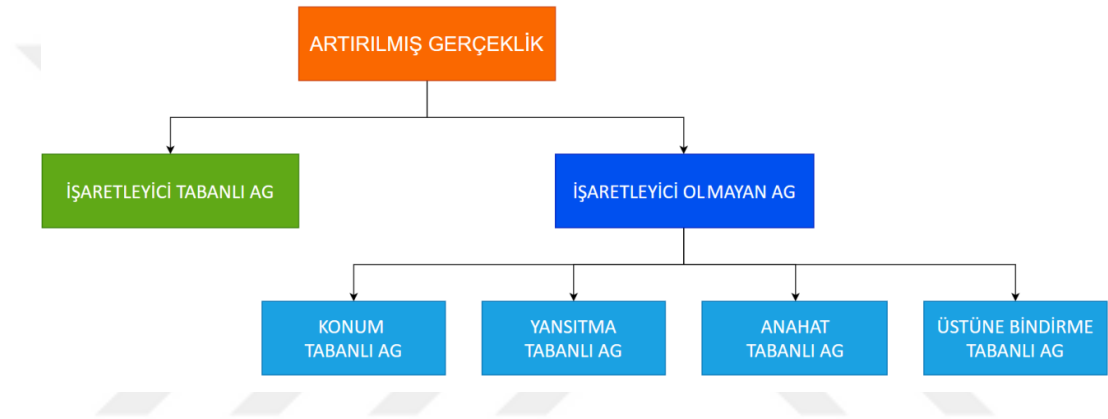


Şekil 2.1. AG çalışma sistemi.

AG teknolojileri ilk olarak askeri alanda savaş uçağı pilotlarına ön ekranda uçuş bilgilerinin gösterilmesi ile kullanılmaya başlanılmıştır. Daha sonra gelişen teknoloji ile birlikte hayatımızın her alanında kullanılmaya başlanılmıştır. Günümüzde birçok alanda yaygınlaşan AG teknolojisi askeri, endüstri, tıp, eğitim, savunma sanayi, ticaret, pazarlama, turizm, mimari, sanat ve eğlence alanlarında etkin olarak kullanılmaktadır [9].

2.1. ARTIRILMIŞ GERÇEKLİK TÜRLERİ

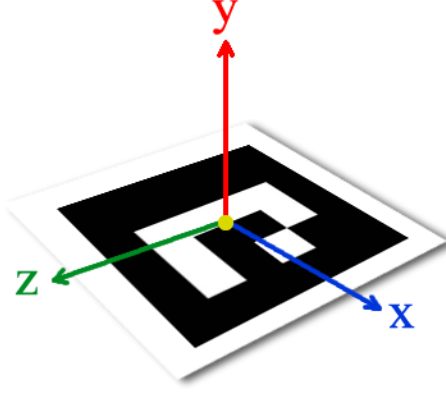
AG teknolojisinin kullanılan donanım ve yazılımlara göre çeşitleri bulunmaktadır. AG sistemlerinde görüntü işleme yöntemleri kullanılır. Çalışma sistemine göre AG sistemleri ikiye ayrılır. Gerçek görüntü üzerine eklenecek sanal nesnelerin ekran üzerindeki gösterim yeri sistemde daha önceden tanıtılan referans işaretçiler kullanılarak yapılıyorsa, İşaretleyici Tabanlı AG (İTAG), referans işaretçi kullanımına ihtiyaç olmadan sanal nesnenin ekranda herhangi bir noktaya yerleştir işlemleri yapılıyorsa İşaretleyici Olmayan AG (İOAG) olarak isimlendirilmektedir [10]. Şekil 2.2’de AG türleri gösterilmiştir.



Şekil 2.2. AG Türleri.

2.1.1. İşaretleyici Tabanlı Artırılmış Gerçeklik

İTAG uygulamalarında temel fikir, bir cihaza ait (telefon, tablet, vb.) kameranın kullanılarak alınan çevreye ait görüntüde daha önceden belirlenen bir referans işaretleyicinin tanınması ve sanal nesnenin işaretleyici üzerinde gösterilmesidir. İTAG uygulamalarında gerçek ortam ile sanal nesneleri birleştirmek için referans işaretçiler kullanılır [11]. İTAG sisteminde kullanılan işaretleyici bir kare kod, resim vb. bir görüntü olabilir. AG uygulamalarında kullanılan referans işaretleyiciler sayesinde sanal nesnenin görüntüleneceği koordinat sistemini oluşturan x, y ve z eksenleri elde edilmektedir (Şekil 2.3). Bu sayede uygulamaya gerçek dünyada bulunan referans işaretleyicinin konum bilgisinin aktarılması sağlanmaktadır [12]. Kameranın referans işaretleyici etrafında hareket ettirilmesi ile ekranda görüntülenen sanal nesnenin de etrafında hareket edilir.



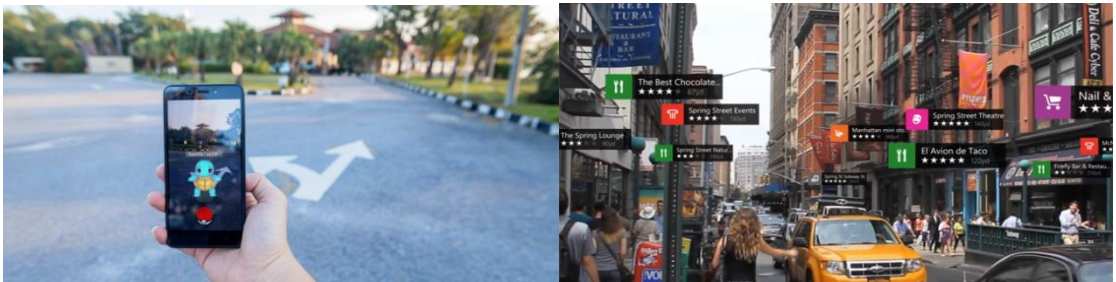
Şekil 2.3. İşaretleyici ve eksenler.

2.1.2. İşaretleyici Olmayan Artırılmış Gerçeklik

AG teknolojisinin en yaygın uygulanan çeşitlerinden biri olan İOAG, coğrafi konum veya mekânsal tanıma gibi farklı tetikleyici türleri ile tetiklenecektir. İOAG'nin temelini oluşturan Eşzamanlı yerelleştirme ve haritalama teknolojisi (Simultaneous localization and mapping - SLAM) AG uygulamalarında kullanılan en ileri teknolojilerden biridir. Bir grup karmaşık algoritmayı işleyerek ve sensör verilerini yöneterek, bilinmeyen bir alanı eşler ve aynı zamanda kendi yerelleştirmesini tanımlar. İOAG uygulamalarının da kullanıldığı alana göre çeşitleri vardır. Bunlar:

2.1.2.1. Konum Tabanlı Artırılmış Gerçeklik

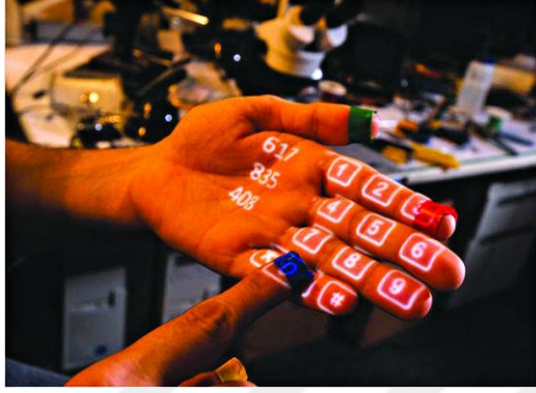
Konum tabanlı AG uygulamalarında kullanıcının konumunun tespit edilmesi için Kablosuz Ağ (Wireless Local Area Network-WLAN), GPS, Radio Frekanslı Tanımlama (RFID), dijital pusula, hızölçer, ivmeölçer, sensör vb. teknolojiler kullanılır. Kullanıcının tespit edilen konumuna göre kamera görüntüsü ile alınan gerçek görüntü, sanal verilerin eklenmesiyle zenginleştirilir [13].



Şekil 2.4. Konum Tabanlı AG.

2.1.2.2. Yansıtma Tabanlı Artırılmış Gerçeklik

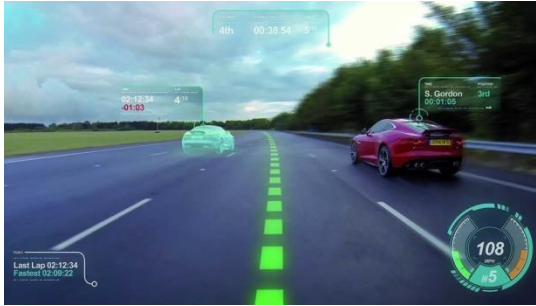
Yansıtma Tabanlı AG, gerçek bir nesne yüzeyine projeksiyon vb. bir cihaz ile görüntü yansıtılması ve yansıtılan görüntü ile insan etkileşiminin algılanması prensibine ile çalışmaktadır [14].



Şekil 2.5. Yansıtma Tabanlı AG.

2.1.2.3. Anahat Tabanlı Artırılmış Gerçeklik

İnsan gözünün dünyanın en iyi kamerası olduğu bilinmesine rağmen, algılayamayacağı durumlar olabilmektedir. Çok uzun süre bir noktaya bakamaz, düşük ışık koşullarında iyi göremez ve kızılötesi görüntüleri ve dalgaları algılayamaz. Anahat Tabanlı AG, bu gibi durumların algılanması için geliştirilen özel kameralar ile farklı açılardan alınan görüntülerin ana hatlarının çizilerek kullanıcıya gösterilmesi temeline dayanır [13]. Karanlık veya sisli bir havada görünmeyen yol şeritleri ve işaretlerinin anahatlarının belirlenerek kullanıcıya gösterilmesi veya binaların ve destekleyici sütunlarının anahatlarının özetlenmesi gibi uygulamalar örnek olarak verilebilir.



Şekil 2.6. Anahat Tabanlı AG.

2.1.2.4. Üstüne Bindirme Tabanlı Artırılmış Gerçeklik

Üstüne bindirme tabanlı AG, nesnenin orijinal görünümünü kısmen veya tamamen aynı nesnenin zenginleştirilmiş yeni görünümü ile değiştirir.



Şekil 2.7. Üstüne Bindirme Tabanlı AG.

2.2. ARTIRILMIŞ GERÇEKLİK İÇİN KULLANILAN ÇEVRE BİRİMLERİ

AG uygulamaların çalışabilmesi ve kullanıcı tarafından etkili bir şekilde kullanılması için hem donanım hem de yazılım alt yapısına ihtiyaç vardır.

2.2.1. Donanım Altyapısı

AG uygulamaları, çevreden alınan görüntülerin üzerinde dijital ortamda hazırlanan sanal nesnelerin gösterildiği bir ortam olduğu için, görüntü alma ve oluşturulan görüntünün kullanıcıya gösterilmesi için çeşitli donanımlara ihtiyaç vardır. Alan B. Craig, Artırılmış Gerçekliği Anlamak: Kavramlar ve Uygulamalar adlı kitabında AG sistemlerinde kullanılan donanımsal bileşenlerin algılayıcılar, işlemciler ve görüntüleyicilerden oluştuğunu; AG'nin içeriğini uygulama, içerik, etkileşim, teknoloji, gerçek dünya ve kullanıcı bileşenlerinin oluşturduğunu belirtmektedir [15].

Algılayıcılar, gerçek dünyadan kamera, sensör, GPS vb. araçlar kullanılarak, sıcaklık, basınç, ışık düzeyi, konum vb. verileri okuyarak, AG uygulamasıyla iletişim kuran bileşenlerdir [15].

İşlemciler ise gerçek dünyadan alınan verileri yorumlayarak sonuç üreten bilgisayar, tablet, mobil telefon gibi AG sisteminin temel bileşenleridir.

Ekranlar (görüntüleyiciler) ise, üretilen sonuçları kullanıcıya aktaran bilgisayar, tablet, mobil telefon, başa takılı ekran (Helmet Mounted Display –HMD), akıllı gözlük vb.

cihazlardır.

2.2.2. Yazılım Altyapısı

AG uygulamaları geliştirmek ve çalıştırmak için gerçek dünyadan alınan görüntüyü işleyerek, elde edilen sonuca göre dijital ortamda hazırlanan sanal görüntüleri kullanıcıya gösterecek bir arayüz yazılımı gerekmektedir. Donanım ve yazılım geliştiricileri için sunulan, AG uygulamalarının çalışmasını sağlayan ve farklı yazılım geliştirme araçlarını içeren yazılım geliştirme araçlarına SDK (Software Development Kit) denir. Bu yazılım geliştirme paketleri, çeşitli yazılım şirketleri tarafından windows, unix, android, ios vb. platformlarda kullanılmak için geliştirilir ve yapılarında bulunan Javascript, Html (Hyper Text Markup Language), CSS (Cascading Style Sheets) ve JQuery dosyaları sayesinde yazılım istenildiği şekilde değiştirilip düzenlenebilir. AG uygulaması için en çok kullanılan yazılım geliştirme araçlarına Vuforia, ARToolKit, DeepAR, Wikitude, EasyAR, NyARToolkit, Kudan, ve Maxst örnek olarak verilebilir.

2.3. ARTIRILMIŞ GERÇEKLİK VE EL İLE ETKİLEŞİM ÜZERİNE YAPILAN ÇALIŞMALAR

AG uygulamalarında gelişen teknoloji ile yazılımsal ve donanımsal olarak kullanıcı ile etkileşim sağlayan çeşitli yöntemler vardır. İnsan hareketlerini ve eylemleri tanıma işlemi en çok kullanılan etkileşim yöntemidir. AG uygulamalarında özellikle el hareketleri ile kontrol üzerine yazılımsal ve donanımsal araçlar kullanılarak etkileşim sağlanmıştır. AG uygulamalarında sanal nesne ile etkileşim kurmak, sanal nesneyi uygun şekilde kontrol etmek için sanal nesne ile insan arasındaki el etkileşimi önemli bir çalışma konusudur. AG teknolojisinde kullanılan 2 boyutlu (2B) görüntü tanıma ve etkileşim tekniği, kullanıcı ile sanal nesne arasında doğal bir etkileşim gerçekleştirmek için bir sınırlamaya sahiptir.

El hareketleri tanıma ve sınıflandırma sistemleri AG ve İBE çalışmaları başta olmak üzere çeşitli görüntü işleme çalışmalarında yaygın olarak araştırılmıştır. İlk olarak 2B düzlemde el takibi olarak çalışılan ve elde edilen bilgi ile bilgisayarda fare kontrolü yapılan el tanıma sistemlerinde, daha sonra masaüstü ortamlarda etkileşimler kurmak için parmak ucu izleme kullanılmıştır. Bir elin 3B uzayda izlenmesi için stereoskopik kameralar, hareket sensörleri vb. algılayıcılar kullanılmıştır. AG uygulamalarında

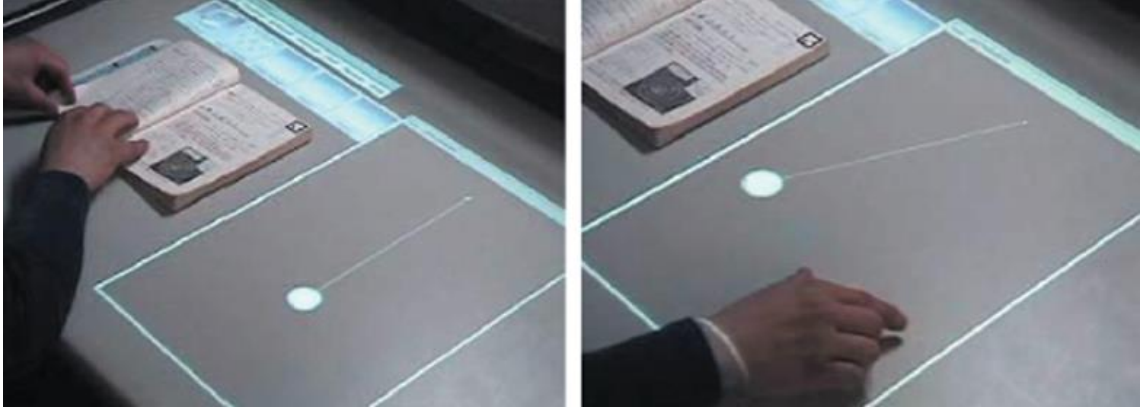
etkileşim sağlamak için özel olarak üzerinde sensörler ve algılayıcılar olan eldivenler tasarlanmıştır. Doğal el tanıma işlemlerinde ise el üzerinde işaretleyiciler veya özel olarak tasarlanan gözlükler ve sensörler kullanılmıştır.

[16]'da etkileşime giren ve tıkanmaya karşı ve doğru, basit, ucuz, hızlı ve sağlam olan bir parmak izleyici üzerine bir çalışma yapılmıştır. Sistem, işaretli bir eldiven, bir stereoskopik izleme sistemi ve insan parmağının kinematik 3B modelini temel almaktadır. Siyah bir eldiven üzerine parmakların eklem yerlerine yerleştirilmiş reflektörlü işaretleyiciler ile bir dış iskelet modeli oluşturulmuştur. İşaretleyiciler kızılötesi ışınlarla algılanmakta ve bu sayede el hareket tanımlama işlemi gerçekleştirilmektedir. Sistem kullanıcının sanal nesnelerle etkileşime girmesine olanak tanıyan bir AG satranç oyununda gösterilmiştir. Geliştirilen AG uygulamasında, kullanıcı 3B olarak gösterilen nesneleri tutabilir, çevirebilir, döndürebilir ve bırakabilir.



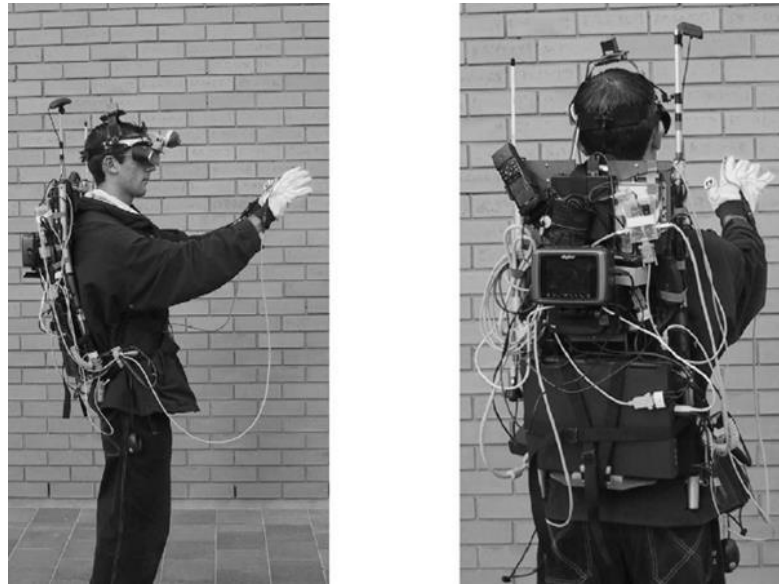
Şekil 2.8. İşaretleyicili Eldiven ve AG Satranç oyunu [16].

[17] ve [18]'de kağıt ve dijital bilgileri düzgün şekilde bir masa üzerine entegre eden EnhancedDesk adlı AG tabanlı masa sistemi geliştirilmiştir. Sistem, kullanıcılara masaüstündeki gerçek nesneler (ör. Kitaplar) ile ilişkilendirilmiş dijital bilgileri görüntüleyen akıllı bir ortam sağlamaktadır. Sistem aynı zamanda kullanıcılara, ellerini ve parmaklarını kullanarak masaüstüne yansıtılan dijital bilgiler ile daha doğal ve daha sezgisel bir etkileşim sağlamaktadır. Oluşturulan sistemde bir projektör, kızılötesi kamera, pan-tilt hareketlerini yapabilen bir kamera kullanılmıştır. Kameralar sisteme tanıtılarak AG tabanlı masa sisteminde gerçek zamanlı parmak izlemesi için yeni bir yöntem geliştirilmiştir.



Şekil 2.9. EnhancedDesk AG tabanlı masa sistemi [17].

[19]'da AG sistemi için dış ortamda giyilebilir eldiven tabanlı kullanıcı arabirimi araçları ile ilgili bir çalışma yapmışlardır. Kullanıcı etkileşiminin ana biçimi el ve kafa hareketlerinin kullanılmasına dayanmaktadır. Sistemde üzerinde işaretleyiciler bulunan bir eldiven ve bir HMD kullanılmıştır. Kask üzerine yerleştirilmiş bir dijital kamera ile alınan görüntüler bilgisayara aktarılmış ve ARtoolkit Kütüphanesi ile AG görüntüsü oluşturulmuştur. Hazırlanan sistemin, etkileşim biçiminin işlevselliğini göstermek için askeri alanda yedek bir helikopter rotoru teslim etme görevi için bir senaryo hazırlanmıştır.



Şekil 2.10. Dış ortam AG sistemi [19].

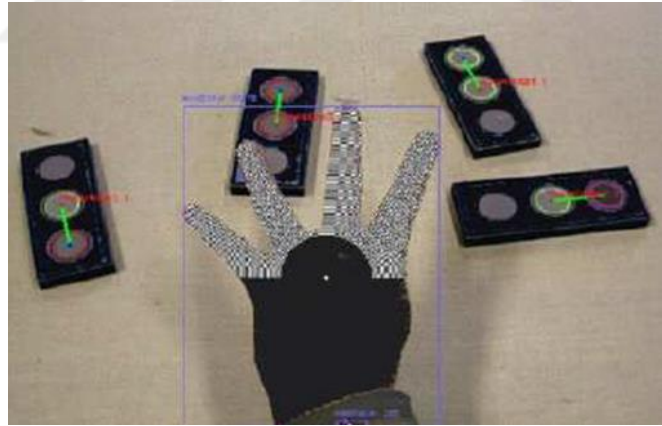
[20]'de etkileşimli AG sistemlerinde hedefin kapanmasının saptanmasını kolaylaştırmak için görüntü sabitlemeyi kullanan bir kapanma siluet çıkarma şemasını üzerine çalışmışlardır. Windows tabanlı olarak geliştirdikleri AG sisteminde, basit el

hareketlerini fare kontrolleri gibi kullanarak sanal bir kontrol paneli ile etkileşim sağlanmaktadır. İşaretleyici üzerinde açılan görüntüye el ile dokunarak, dokunduğu butonun çalışmasını sağlamaktadır. Bu sistemde görüntüsü kapanan buton basılmış olarak algılanmakta ve butona bağlı işlem gerçekleştirilmektedir.



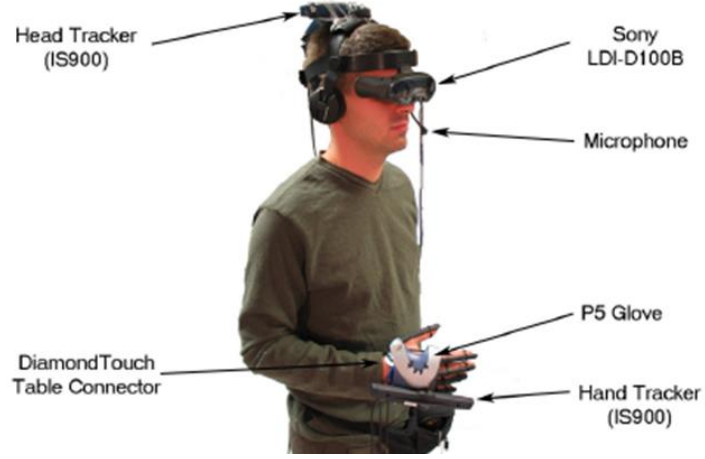
Şekil 2.11. AG Kontrol paneli kutucuğu [20].

[21]'de bir masa toplantısı gibi masa etrafında çok kullanıcılı hareket tanıma tabanlı bir AG sistemi geliştirilmiştir. Geliştirilen sistemde çift HMC (Head Mounted Camera) kullanılmıştır. Bu AG sistemi, masa üstüne yerleştirilen ve konumu takip edilen yer tutucu nesneler, işaret aygıtları ve hareketlere dayalı bir arabirimden oluşmaktadır.



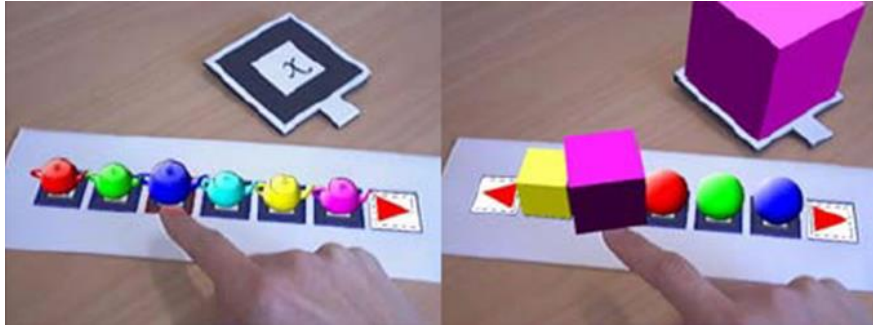
Şekil 2.12. HMC giriş görüntüsü [21].

[22] ve [23]'te arkeolojik kazıların yerinde olmayan görselleştirilmesi için deneysel bir işbirliğine dayalı karma AG sistemi olan Arkeoloji için Görsel Etkileşim Aracı (Visual Interaction Tool For Archaeology –VITA) üzerine bir çalışma yapılmıştır. VITA paletli eldiven, kasklı ekran ve masa üstüne yansıtılan bir görüntüden oluşan çok kullanıcılı ve ses ile kontrol edilebilir bir karma AG sistemidir. Arkeologlar için, kazı alanlarının dijitalleştirilmiş 3B görüntüsü etrafında, karşılıklı olarak incelenmesini, tartışılmasını sağlayan bir sistemdir.



Şekil 2.13. VITA AG sistemi ve etkileşim araçları [22].

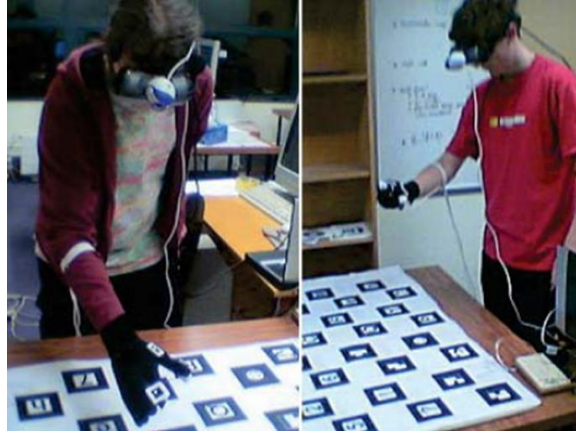
[24]'te AG ortamlarında kullanıcı etkileşimi sağlamak için kapanma tabanlı bir sistem geliştirilmiştir. AG sisteminde kullanıcı el ya da başka bir materyal ile işaretleyicinin üzerine geldiği zaman, sistem belirli bir süre beklemekte ve görüntüsü kapanan işaretleyicileri tanımlamakta ve kapanan işaretleyicilere bağlı olan dijitalleştirilmiş bilgiler gösterilmektedir. Oluşturulan sistemde donanım olarak basit bir bilgisayar ve bilgisayar kamerası kullanılmıştır. Yazılım olarak ise ARtoolkit ve açık grafik kütüphaneleri kullanılmıştır.



Şekil 2.14. AG menü çubuğu [24].

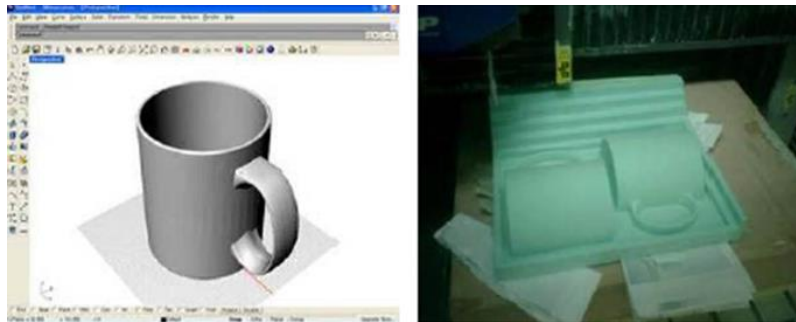
[25]'te AG uygulamalarında sanal nesnelerle etkileşim sağlamak için doğal, parmak ucu tabanlı bir FingARTips adı verilen bir teknik üzerine çalışılmıştır. Çalışmada ARToolKit Kütüphanesi kullanılmıştır. El izleme için özel olarak tasarlanmış üzerinde işaretleyiciler olan bir eldiven ve görüntüleme işlemi için HMD ile HMC kullanılmıştır. Kullanıcının sanal nesneler ile temas ettiğinin anlaması için, eldivenin parmak uçlarına ziller yerleştirilmiştir. Uygulama olarak bir şehir modellemesi yapılmıştır ve kullanıcı, şehir modelinde olan binaları tutarak yerlerini değiştirebilmekte, döndürebilmekte,

işaret ederek yollar oluşturabilmekte ve uygulama içine yerleştirilmiş düğmelere basabilmektedir.



Şekil 2.15. Şehir planlama gösterimi [25].

[26]'da AG uygulamalarında sanal nesnenin el ile etkileşiminde nesne takibinin hata vermesi nedeniyle, 3B Bilgisayar destekli tasarım (Computer Aided Design -CAD) verilerine dayanan dokunmatik ve kavramalı bir arayüz olarak, AG teknolojilerini fiziksel köpüklere uygulayan artırılmış köpük üzerine bir çalışma yapılmıştır. Artırılmış Köpüğü kullanarak, bir köpük kalıbına, 3B üretim için kullanılan aynı CAD modeliyle oluşturulmuş bir 3B sanal nesne yerleştirilmiştir. Uygulamada örnek olarak bir kupa tasarımı ve kontrolü yapılmıştır. AG uygulaması yazılım olarak ARToolKit Kütüphanesi ile geliştirilmiştir. Donanım olarak ise bir kamera ve kasklı ekran kullanılmıştır.



Şekil 2.16. Artırılmış köpük [26].

[27]'de AG oyunlarında kullanıcının çıplak el ile sanal nesnelerle etkileşim sağlaması için bir uygulama geliştirilmiştir. Sanal bir basketbol oyunu ve müzik çalar geliştirmişlerdir. Kullanıcı sanal müzik çalar üzerinde bulunan başlat, durdur vb. düğmelerini eli ile çalıştırabilmektedir. Basketbol oyununda ise kullanıcı sanal

basketbol topuna eli ile kontrol ederek basket atabilmektedir.



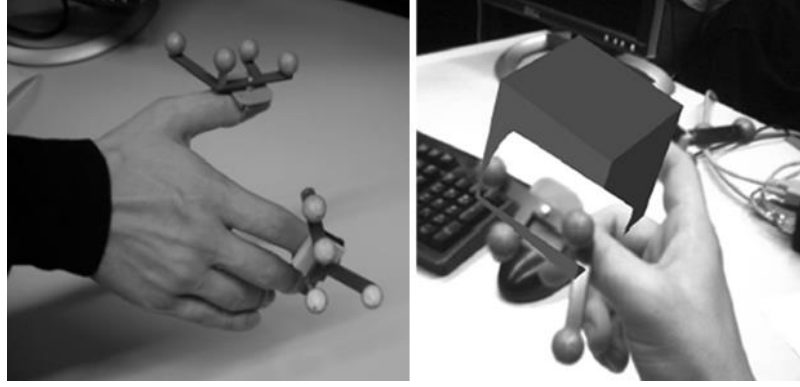
Şekil 2.17. Sanal basketbol oyunu [27].

[28], [29] ve [30]'da AG uygulamalarında nesnelerini kolayca belirlemek için bir işaretleyici yerine insan elini model alan işaretsiz kamera izleme tabanlı kullanıcı arabirimi üzerine Handy AR adı verilen bir çalışma yapılmıştır. Geliştirilen AG uygulamasında insan eli takip edilmekte ve parmak uçları belirlenmektedir. El ile bağlantılı olarak çalıştırılan AG uygulamasında, kullanıcı çeşitli el hareketlerini kullanarak sanal olarak gösterilen 3B nesneleri seçip, yönünü değiştirip tekrar geri bırakabilmektedir. Uygulama ARTag ve OpenCV (Open Source Computer Vision) kütüphaneleri kullanılarak geliştirilmiştir. Donanım olarak ise bir bilgisayar ve bilgisayar kamerası kullanılmıştır.



Şekil 2.18. HandyAR nesne seçimi [28].

[31]'de AG uygulamalarında, statik ve dinamik hareketleri ayırt edebilen otomatik hareket tanıma sistemi ile ilgili bir çalışma yapılmıştır. Geliştirilen sistemde kızılötesi izleme kullanılmaktadır. Kullanıcıların parmaklarına takılan kızılötesi hedefler ile her parmağın konumu ve yönü hakkında bilgi alınmaktadır. Sanal olarak gösterilen 3B blokları tutarak yerini değiştiren bir uygulama geliştirmişlerdir. Sanal görüntüler kasklı bir ekran kullanılarak kullanıcıya aktarılmıştır.



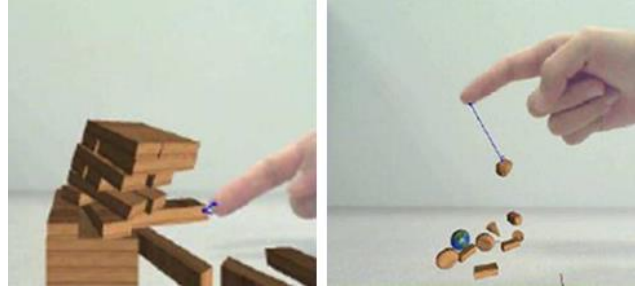
Şekil 2.19. Parmaklara takılan kızılötesi hedefler ve blok tutma [31].

[32]'de çok kullanıcı AG sistemleri için 3B el takip sistemi üzerine çalışmışlar ve 3B görsel tabanlı doğal el etkileşimi yöntemi geliştirilmiştir. Sistem dört adımdan oluşmaktadır: (1) Deri rengi bölümleme, (2) Özellik noktası bulma, (3) El yönü hesaplama ve (4) Basit çarpışma algılama. Sistem OpenCV Kütüphanesi kullanılarak geliştirilmiş ve 3B görüntüyü elde edebilmek için stereo kamera kullanılmıştır. Sisteme ses ile tanıma ve kontrol entegre edilmiştir. AG uygulamasını görüntüleyebilmek için el ile tutulabilir bir ekran kullanılmıştır.



Şekil 2.20. Masaüstü AG sistemi ve el ile tutulan ekran [32].

[33]'de kullanıcının gerçek hayatta olduğu gibi sanal ortamda tasarlanan nesneler ile el ve parmak hareketleri ile etkileşime girebildiği AG uygulaması geliştirilmiştir. Nesnelerle etkileşimde fiziksel olarak gerçekçi bir deneyim sağlamak için sanal nesnelere fizik motoru kullanılarak yerçekimi, çarpışma gibi özellikler eklenmiştir. Sistemdeki algılama ve etkileşim, herhangi bir belirteç veya sensör olmaksızın stereo kamera ile sağlanmaktadır. Sistemin çalışmasını göstermek üzere sanal bloklardan oluşan jenga oyunu ve parmak uçlarıyla balık tutma oyunu olmak üzere iki adet uygulama geliştirilmiştir.



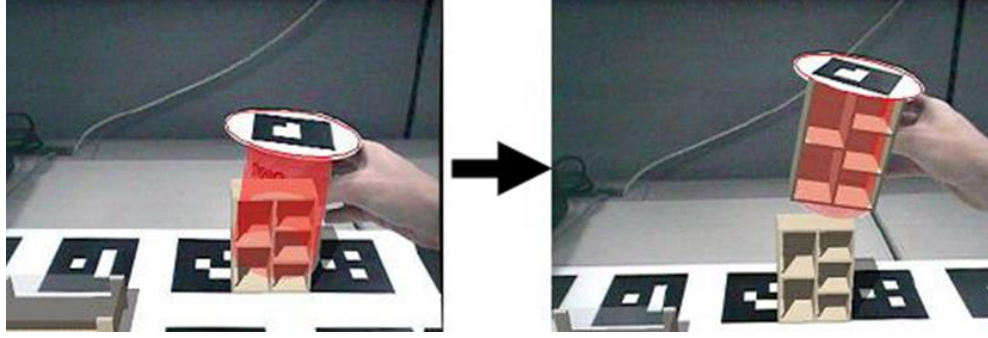
Şekil 2.21. Jenga ve balık tutma oyunu [33].

[34]'de mobil cihazlarda AG ve etkileşim için tek elle cihazın tutulduğu bir yaklaşım sunulmuştur. Önerilen uygulamada, kullanıcının bir eli mobil cihazı tutarken, diğer elini serbest olarak mobil cihazın görüntü alanına yerleştirmesi ile kullanıcının eli tanımlanmakta ve avuç içi belirlenerek sanal nesne oynatılmaktadır. Sistem parmak ucu izleme yöntemi ile avuç içindeki sanal nesne kullanıcının elini açması veya kapatması gibi el hareketlerine tepki vermektedir. Ayrıca, titreşim sensörleri ile dokunsal bir eldiven takarak sanal nesneyle dokunsal etkileşimler sağlanmaktadır.



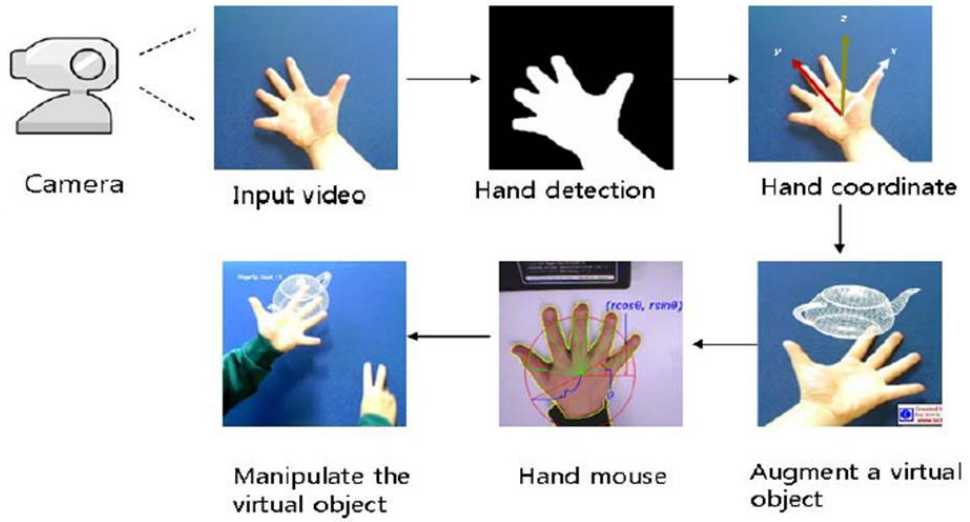
Şekil 2.22. Avuçiçi tanıma AG uygulaması [34].

[35]'de AG uygulamalarında sanal nesne ile kullanıcı etkileşimi, çok kullanıcılı mobil etkileşim gibi daha iyi bir kullanıcı deneyimi sağlamak için MagicCup adını verdikleri bir etkileşim yöntemi üzerine çalışılmıştır. MagicCup, altı boyutlu konumu algılayan ve bilgi veren bir işaretleyiciye sahip, fincan şeklinde portatif bir AG giriş cihazıdır. İşaretleyici ile sanal nesne üzerine geldiği zaman, çeşitli işlemler yapılabilmektedir.



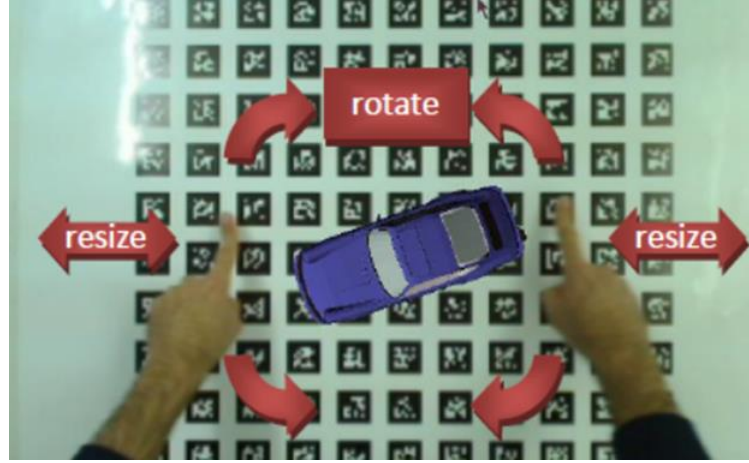
Şekil 2.23. MagicCup AG giriş aracı [35].

[36] ve [37]'de AG uygulamalarında parmak ucu izleme ve el kalıbı tanıma ile sanal nesneleri kontrol etmek için tek kamera kullanan işaretleyici olmayan bir AG sistemi geliştirilmiştir. Sol el, İOAG sisteminde sanal bir işaretleyici olarak kabul edilmekte ve sanal nesneler avuç içinde gösterilmektedir. Sağ el ise fare olarak kullanılmakta ve sağ el ile yapılan işaretlere göre sanal nesne kontrol edilmektedir.



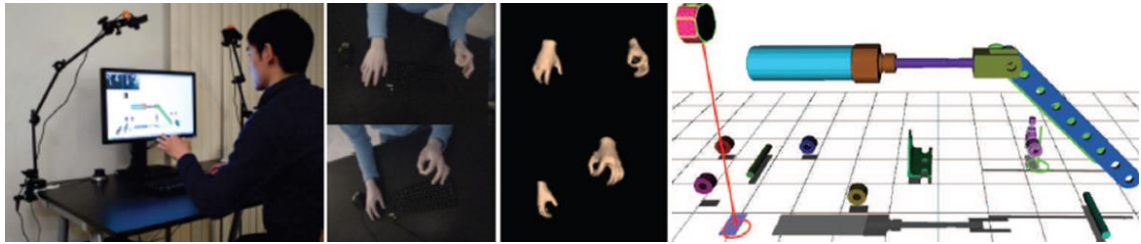
Şekil 2.24. İşaretleyicisiz el tanıma sistemi [36].

[38]'de bir dizi işaretleyicilerden oluşan, ArTag ve OpenCV kütüphanelerini kullanan, tek kamera ile görüntü alan bir masaüstü AG uygulamasında el ile etkileşim yaklaşımı önerilmiştir. Sistem, kullanıcının ellerini haar benzeri özellik sınıflayıcılarını ve Adaboost algoritmasını kullanarak algılamakta ve masadaki konumları sabit işaretleyicilerle ilişkilendirmektedir. Bu sayede, kullanıcıya, çıplak elle masanın üzerinde bulunan sanal nesneleri taşıma, döndürme ve yeniden boyutlandırma imkânı verilmektedir.



Şekil 2.25. AG sanal nesne döndürme ve boyutlandırma [38].

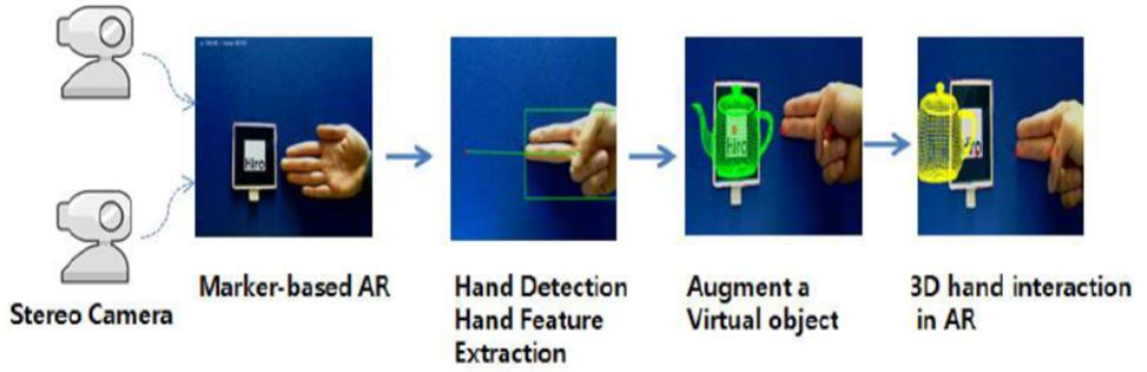
[39]'da CAD uygulamalarında her iki eli de kullanabilen, 3B nesneleri kontrol edebilen bir el takip sistemi üzerine çalışılmıştır. CAD yazılımında klasik olarak kullanılan kısıtlama özelliklerini destekleyen, yalnızca birkaç basit hareketi temel alan, kullanıcıların dirseklerini masalarında dinlendirip klavye ve fare ile birlikte çalışmasını sağlayan bir 3B girdi metaforu tanımlanmıştır. Yaklaşımda, kullanıcının elini gözlemlemek için iki adet web kamerası kullanılmıştır. El takibi sistemini kullanarak, çeşitli CAD modellerinin verimli 3B mekanik montajları gösterilmiştir.



Şekil 2.26. El tanıma ile montaj işlemi [39].

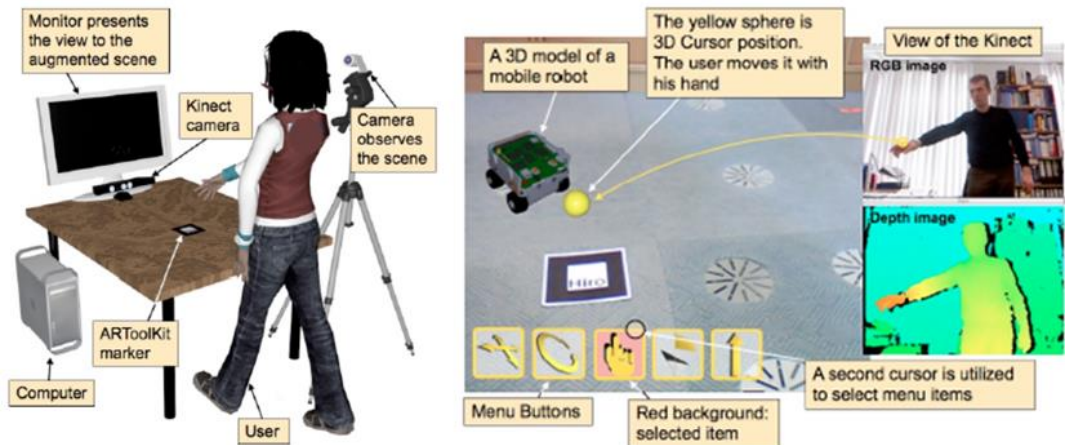
[40]'da AG bir sanal nesne ile insan arasındaki el etkileşimini, stereo görüş tekniğini ve ARToolKit kütüphanesini kullanarak sağlanmıştır. Stereo görme tabanlı el etkileşimi üç büyük aşamadan oluşmaktadır. İlk aşamada, insan eli, cilt rengi modeli ve görüntü bölümlenmesi kullanılarak giriş video görüntüsünden saptanmaktadır. İkinci aşamada, parmak ucu, avucun ortası ve işaretçinin merkezi gibi elin 3B pozisyonları, stereo görme fark haritasını kullanarak değerlendirilmektedir. Üçüncü aşamada, kullanıcı, insan eliyle nesne arasındaki çarpışmayı tespit ederek sanal bir nesneyle etkileşime girebilmektedir. İşaretleyici üzerinde gösterilen 3B sanal nesneye el ile yaklaşıldığı zaman, kullanıcı etkileşimi sağlanır ve nesnenin renk ve şekil gibi özelliklerini

değiştirebilmektedir. Buna ek olarak nesneyi mevcut konumdan iterek veya çekerek nesnenin geometrik konumunu değiştirebilmektedir.



Şekil 2.27. El ile etkileşim metodu işlem basamakları [40].

[41]'de teknik sistemlerde 3B modellerinin etkileşimli montaj işlemleri için Microsoft Kinect video kamera kullanılan bir AG montaj sistemini sunulmuştur. AG montaj sistemini çalıştırmak için el hareket tanıma yazılımı ve bir AG uygulaması kullanılmıştır. Çalışmada el izleme için ARToolKit yazılımı ve desen tabanlı izleme sistemi kullanılmıştır. El hareket tanıma yazılımı OpenCV tabanlı olarak geliştirilmiş ve yumruk, açık el, kapalı el, işaret parmağı ve sallayan bir jesti olmak üzere beş jest algılanmaktadır. Kullanıcı ekrana gelen menüden yapmak istediği hareketi seçebilmekte ve buna bağlı olarak ekrana gelen kursör ile ilgili hareketleri eli ile kontrol edebilmektedir. Çalışmada etkileşim teknikleri bir grup kullanıcı tarafından test edilmiştir ve test sonuçlarına göre etkileşim tekniklerinin sezgisel bir montajı kolaylaştırdığını görülmüştür.



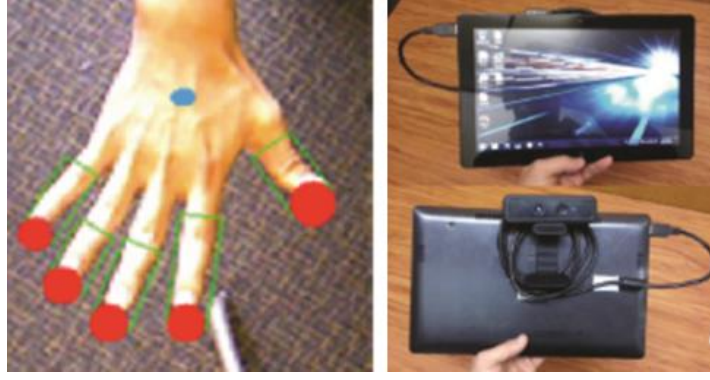
Şekil 2.28. AG sistemi ve uygulaması [41].

[42]'de Microsoft Kinect derinlik kamerası, stereoskopik projektör, 3B shutter gözlük ve kavisli bir ekran ile donatılmış, gerçek ve sanal dünyaları bir masanın üstünde tek bir mekânsal kayıtlı deneyime birleştirmek üzere tasarlanmış Miragetable isimli etkileşimli bir masaüstü AG uygulaması geliştirilmiştir. Derinlik kamerası kullanıcının gözlerini takip etmekte ve kamera önünde (kullanıcının vücudu ve elleri dâhil) yerleştirilen herhangi bir nesnenin şeklini ve görünümünü gerçek zamanlı olarak yakalamaktadır. Bu gerçek zamanlı yakalama, masadaki fiziksel nesnelerin neden olduğu deformasyonları hesaba katan perspektifli stereoskopik 3B görselleştirmeleri tek bir kullanıcıya sağlamaktadır. Buna ek olarak, kullanıcı herhangi bir eldiven, işaretleyici veya araç gereç olmaksızın fiziksel olarak gerçekçi serbest hareketlerle sanal nesnelerle etkileşime girebilmektedir. Bu yetenekler üç uygulama örneği ile açıklanmıştır: sanal 3B model oluşturma, gerçek ve sanal nesnelerle etkileşimli oyun ve 3B telekonferans deneyimi. Ayrıca, kullanıcının farklı arka plan renkleri ve geometrileri üzerinde (ör. Boşluklar, damla) yansıtıldığında bile bu nesneleri doğru algıladıklarını doğrulayan öngörülen 3B nesnelerin algılanışını sistem üzerinde değerlendirmişlerdir.



Şekil 2.29. Miragetable etkileşimli masaüstü AG uygulaması [42].

[43]'de mobil cihazlar için AG uygulamalarında oluşturulan 3B sanal nesneleri kontrol etmek için Microsoft Kinect derinlik kamerası kullanan bir çalışma yapılmıştır. Geliştirilen uygulamada Windows işletim sistemi tabanlı bir tablet ve OpenCV Kütüphanesi kullanılmıştır. Mobil cihaz üzerine monte edilen Microsoft Kinect ile parmak uçları tespit edilmekte ve koordinatları çıkarılmaktadır. Bu koordinatlar AG uygulamasında sanal nesneyle etkileşim sağlamak için kullanılmıştır.



Şekil 2.30. Parmakucu tanıma ve sistem prototipi [43].

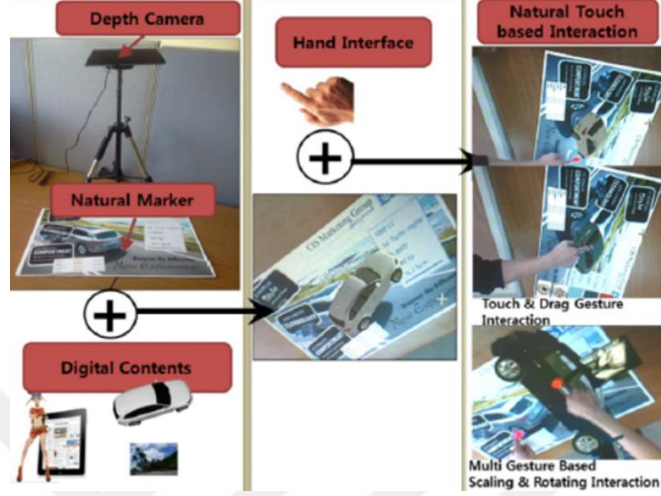
[44]'de kullanıcıların ellerini ve parmaklarını cep telefonu ekranında görünen sanal nesnelerle etkileşimde bulunmak için kullanan işaretleyici tabanlı bir AG uygulaması geliştirilmiştir. Sistem ARToolKitPlus Kütüphanesi ile tasarlanmıştır ve Nokia N900 ile Maemo 5 telefonlarında çalıştırılmıştır. Birden çok el algılama yönteminin birleştirilmesi genel doğruluğunu artırmış, ancak özellikle cep telefonu uygulamalarındaki genel performansı da yavaşlatmıştır. Sahneden eli tespit etmek için renk bölütlemesi uygulayarak çok basit bir yaklaşım izlenmiştir. Cilt rengi aralığı için farklı renk uzayları önerilmiştir; renk alanı dönüşümünün maliyetinden kaçınmak için, bu proje için RGB (Red-Green-Blue) renk uzayı seçilmiştir.



Şekil 2.31. AG sistemi çalışması [44].

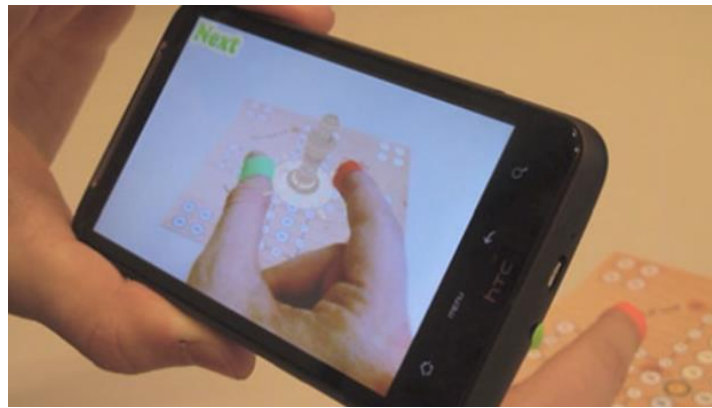
[45]'de AG tabanlı ortamlarda, sürükleyici his ve doğal etkileşimi iyileştirmeyi amaçlayan sanal nesneleri manipüle etmek ve etkileşim kurmak için kullanıcı arabirimleriyle uğraşan İOAG uygulamasını bir derinlik kamerasıyla birleştirilmiştir. Ayrıca elle dokunmatik tanımayı basitleştirmek için Kinect tarafından üretilen nokta bulutu analiz edilmekte ve filtrelenmektedir. Önerilen yaklaşım, AG etkileşimlerini kolayca tetikleyebilmekte ve kullanıcıların daha sezgisel ve doğal duyumlara sahip olmalarını sağlamakta ve çeşitli AG ortamlarında çok fazla kontrol verimliliği sağlamaktadır. Dahası, elde edilen bulut bulutu analizi ile geleneksel AG yaklaşımlarına özgü el ve kol bölgesinin oklüzyon problemi kolayca çözülebilmektedir. Çalışma

aşağıdaki adımları izlemektedir: (1) İOAG için önışleme, (2) derinlik kamerası kullanılarak derinlik bilgisinin renkli görüntüsü ve nokta bulutunun yakalanması, (3) el dokunmalarının tanınması, (4) el ve kol tikanıklığının giderilmesi ve AG bilgilerinin fiziki alanda arttırılması ve (5) doğal ve elle dokunmatik AG etkileşimleri



Şekil 2.32. Etkileşimli AG uygulaması şeması [45].

[46] ve [47]'de cep telefonlarında kullanılan AG uygulamalarında, parmak izlemenin hareket tabanlı etkileşim potansiyelini araştırılmıştır. Geliştirilen uygulamada, işaretleyici üzerinde çalışan sanal nesneler el ile kontrol edilebilmektedir. El tanıma işlemi için iki parmak ucuna renkli işaretleyiciler yerleştirilmiştir. Bu sayede mobil AG uygulamasında sanal nesnelerin çevrilmesi, ölçeklendirilmesi ve döndürülmesi gibi işlemler gerçekleştirilmiştir.



Şekil 2.33. AG kart oyunu ve nesne etkileşimi [46].

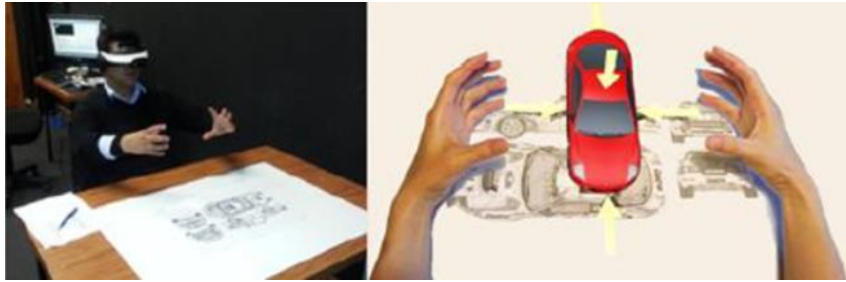
[48]'de üzerine iki adet kamera yerleştirilmiş stereo görüntüleme özelliği olan bir video gözlüğü kullanılan AG sistemleri için çoklu el algılama ve izleme yöntemi ile el

hareketi tanıyan bir sistem geliştirilmiştir. Çalışmada farklı ipuçlarının birleşimi algılama oranını arttırmış, arka plan bölgelerini ortadan kaldırmış ve bu nedenle zorlu koşullar altında izleme performansını arttırmıştır. Tespit edilen el pozisyonları ve yörüngeler, tıklama, seçme vb. işlemleri gerçekleştirmek için kullanılmıştır.



Şekil 2.34. Çoklu el izleme sistemi [48].

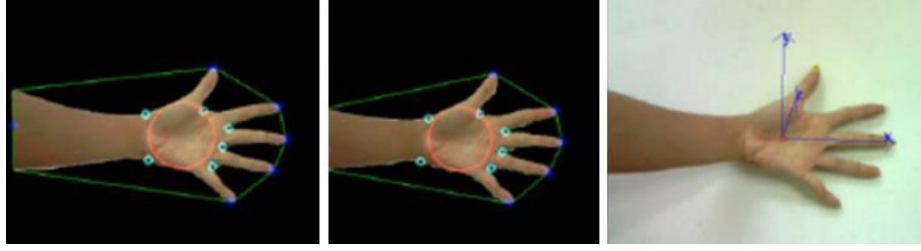
[49]'da AG uygulamalarında en çok kullanılabilecek el hareketleri çizelgesi oluşturmak için bir çalışma yapılmıştır. 3B sanal nesne kasklı bir ekran ve derinlik kamerası kullanan işaretleyici tabanlı bir AG uygulaması ile kullanıcılara gösterilmiştir. Daha önceden belirlenen 40 farklı görevden, çalışmaya katılan 20 katılımcının istediği görevi yaparak, bu görevleri değerlendirmesi istenilmiş ve sonuçta AG uygulamalarında kullanımı kolay olan ve en çok kullanılabilecek el hareketleri için bir çizelge oluşturulmuştur.



Şekil 2.35. AR araba boyutlandırma animasyonu [49].

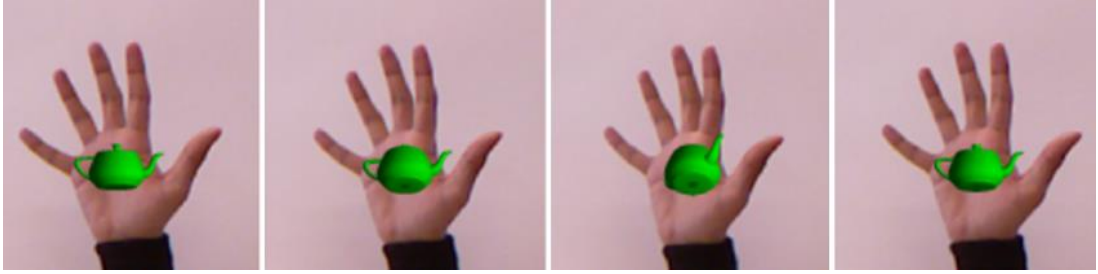
[50]'de gerçek ortamda sanal nesnelerin kaydedilmesi için uzatılmış eli kullanan, sanal nesneleri el üzerinde gösteren ve elin yönlendirilmesine göre oynatabilen bir AG uygulaması üzerine çalışılmıştır. Uygulamada görüntüleri yakalamak için stereo kamera kullanılmıştır, böylece elin derinlik bilgisi oluşturulmuştur. Deri rengi bölümlenmesi, el bölgelerini görüntülerden kesmek için kullanılmıştır. Eşlik işaretleri yerine avuç içi ve parmak uçlarının merkezi gerçek zamanlı olarak izlenmiştir. Hesaplanan derinlik bilgisini bir araya getiren bu el özelliklerinin 3B konumları daha sonra elle ilgili altı serbestlik düzeyi (six degrees of freedom -6DOF) kamera pozunu tahmin etmek için

kullanılmış ve bu da sanal nesnelerin avuç içine doğru bir şekilde takviye edilmesini sağlamıştır. Kullanıcılar AG nesnelerini çıplak elle sezgisel bir şekilde inceleyebilmekte ve manipüle edebilmektedir.



Şekil 2.36. Avuçiçi tespiti [50].

[51]'de bir derinlik kamerası kullanarak AG uygulamalarında el tanıma ve tanınan el üzerinde sanal bir nesne gösterilmesi üzerine bir çalışma yapılmıştır. El üzerinde gösterilen sanal nesne elin hareket ettirilmesi ile hareket yönünde dönebilmektedir. Derinlik kamerası sistemi eğitmek, elin renkli görüntülerinin ve yönlendirme açılarının oluşturduğu geniş bir veri kümesini yakalamak için kullanılmıştır.



Şekil 2.37. El tanıma ve sanal nesne gösterimi [51].

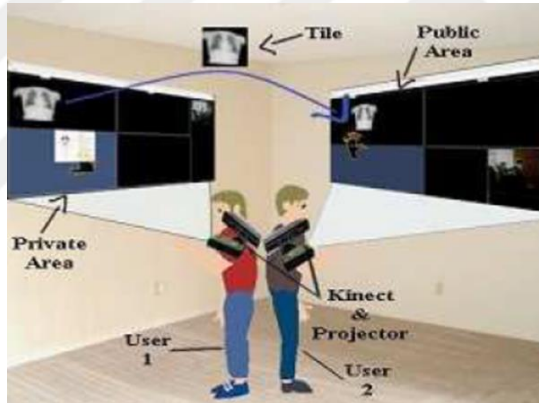
[52]'de NyARToolkit1 kütüphanesi ve Kinect sensörü kullanılarak geliştirdikleri çalışmalarında AG ortamlarında gösterilen sanal 3B içerik ile etkileşim sağlayan üç farklı teknik incelenmiş ve sonuçları karşılaştırılmıştır. Bu teknikler şunlardır:

- 1) El ile tutulan işaretleyici: elle tutulan işaretleyicinin hareket ettirilmesi ile kontrol edilebilen bir AG uygulaması
- 2) Serbest el etkileşimi: Bir el ile işaretleyici tutuluyor ve sanal nesne gösteriliyor, diğer ile yapılan çeşitli hareketler ile sanal nesneyi kontrol edilebilen bir AG uygulaması
- 3) Tuş takımı denetleyicisi: Bir el ile işaretleyici tutuluyor ve sanal nesne gösteriliyor, diğer ile ekranda çıkan bir menüye dokunularak sanal nesneyi kontrol edilebilen bir AG uygulaması



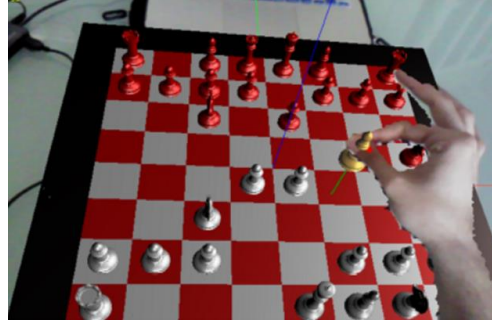
Şekil 2.38. El ile tutulan işaretleyici, Serbest el etkileşimi, Tuş takımı denetleyicisi [52].

[53]'de RemoAct adı verilen İOAG tabanlı, Microsoft Kinect ve projektör kullanan çok kullanıcılı EmguCv kütüphanesi ile oluşturulmuş bir AG sistemi geliştirilmiştir. Bu sistem, düz yüzeylere yansıtılan görüntü ile etkileşimde bulunmak için hareketli ve sezgisel bir çözüm sunmaktadır. Her kullanıcının kamusal ve özel alanları vardır; burada kullanıcılar anında nesne oluşturabilmekte ve başkalarıyla paylaşabilmekte ve bu genel nesne AG yoluyla diğer kullanıcılara gösterebilmektedir. Etkileşim, el hareketleri, parmak izleme ve el takibi yoluyla yapılmaktadır.



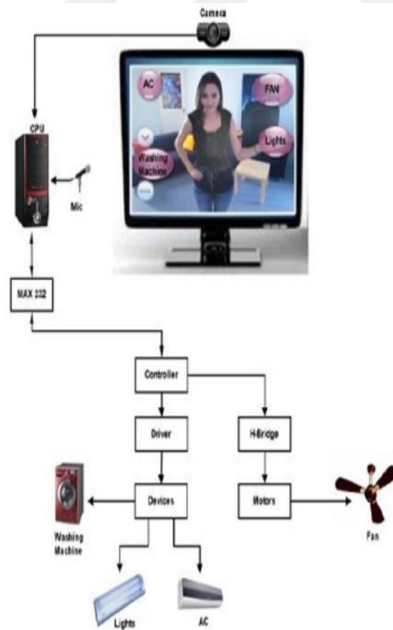
Şekil 2.39. RemoAct AG sistemi [53].

[54]'de kullanıcılarla sanal içerik arasındaki etkileşimi anlayabilen, kullanıcının Microsoft Kinect kullanarak sanal içerikle doğal bir şekilde etkileşim kurmak için başparmağını ve işaret parmağını kullanan, ArUco kütüphanesi kullanılarak oluşturulmuş bir AG satranç oyunu geliştirilmiştir. AG satranç oyunu için işaret parmağı ve başparmak ile tutma üzerine bir algoritma geliştirmişlerdir.



Şekil 2.40. AG Satranç oyunu [54].

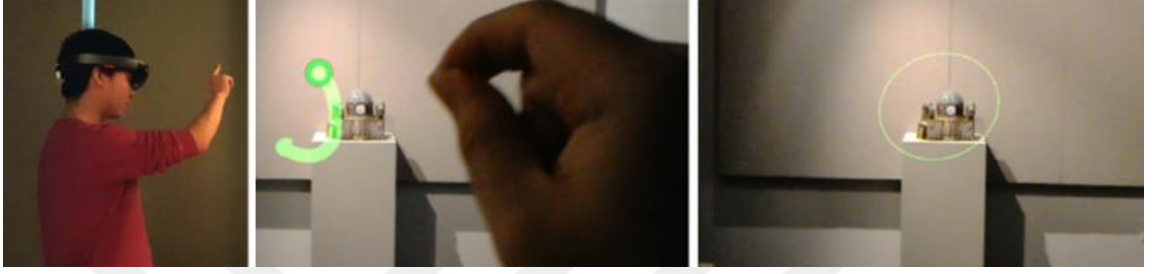
[55]'de AG teknolojisini kullanarak lamba, makine, klima vb. elektrikli aygıtları el hareketleriyle kontrol edebilen bir uygulama üzerine çalışılmıştır. Sistem bir kamera ile görüntü almaktadır. Geliştirilen sistem ile 8 adede kadar elektrikli cihaz kontrol edilebilmektedir. Kullanıcı ve ekran arasında sanal bir düzlem bulunmakta ve kullanıcı bu sanal düzlemle etkileşime girmektedir. Sanal düzlemde kullanıcının elleri ile parmakları izlenerek, hareketleri tanımlanmaktadır. Bu hareketler daha sonra belirli bir cihazı seçmek ve kullanıcının isteğine göre açıp kapatmak için kullanılan ekrandaki düğmeyi seçmek için kullanılmaktadır.



Şekil 2.41. AG ev aletleri kontrolü blok diyagramı [55].

[56]'da HoloLens destekli AG platformunu kullanarak yüzey çizimi ve hava çizimi yöntemleri üzerine bir çalışma yapılmıştır. Yüzey çizimi yöntemi için, hareket girişinin gerçek dünyadaki nesnelerin yüzeylerinde doğrudan gerçekleşmesine izin verilerek,

dünya modeline yansıtılan bir imleç kontrol edilmektedir. Hava çizimi yöntemi için, hareket çizimi kullanıcının parmak ucunda gerçekleşmekte ve serbest bırakıldığında dünya modeline yansıtılmaktadır. Yöntemler, gerekli geçiş anahtarı ve imleç kontrolü ile ilgili olarak farklı özelliklere sahiptir. Kullanıcıların, farklı yöntemleri kullanarak farklı mesafelerde iki farklı gerçek dünya nesne üzerine çektiği bir deney gerçekleştirilmiştir ve yüzey çiziminin, hava çiziminden daha doğru sonuç ürettiği görülmüştür.



Şekil 2.42. Yüzey çizimi ve hava çizimi yöntemleri [56].

3. DERİN ÖĞRENME

Derin öğrenme, yapay zekâ alanında son yıllarda yaygın olarak kullanılmaya başlayan bir makine öğrenme tekniğidir. Büyük miktarda veriler üzerinde çalışılarak, özellik çıkarma, sınıflandırma, desen analizi ve dönüştürme işlemleri birden fazla doğrusal olmayan gizli katman ve nöronlardan oluşan yapay sinir ağlarında yapılmaktadır [57]. Derin öğrenme modellerinde bir katmanın çıkış verisi bir sonraki katmanın giriş verisi olarak kullanılmaktadır. Derin öğrenmenin avantajı, geleneksel makine öğrenmesinde olduğu gibi işleme giren verilere ait öznitelikleri çıkarmak için işlem yapmaya gerek olmamasıdır. Derin öğrenmede, etiketlenmiş olan verilerden oluşan bir veri seti giriş verisi olarak kullanılır ve yapay sinir ağlarından oluşan bir ya da daha fazla gizli katmanlarda çeşitli algoritmalar kullanılarak verilere ait özellikler denetimli/denetimsiz özellik öğrenmesi ve hiyerarşik özellik çıkarımı ile otomatik olarak çıkartılır ve öğrenilir [58-59].

Derin öğrenme yöntemleri, Lecun ve arkadaşları tarafından 1998 yılında el ile yazılan rakamları tanımak için önerilen ilk evrimsel sinir ağı modeli olan LENET ile popüler olmaya başlamıştır [60]. İlerleyen yıllarda, nesne tanıma ve görüntü işleme alanında CNN kullanımı yaygınlaşmış ve farklı yapılar, farklı katman sayısına ve nöron sayısına sahip olan AlexNet [61], ZFNet [62], ResNet [63], VGG [64], GoogLeNet [65], DenseNet [66], MobileNet [67], InceptionV2 [68], InceptionV3 [68] vb. modeller tasarlanmıştır. Bu tasarımların temelindeki ilke, daha derin bir CNN modelinin çok karmaşık ve büyük ölçekli verileri işleme konusunda daha güçlü bir kabiliyete sahip olmasıdır [69].

Son yıllarda makinesi öğrenmesi dalında, ses, video ve görüntü dosyalarından nesne tanıma ve sınıflandırma üzerine yapılan çalışmalarda derin öğrenme tekniklerinin kullanılması ile birçok başarı elde edilmektedir [70-72].

CNN yapıların nesne tanıma çalışmalarındaki başarılarından dolayı birçok araştırmacı derin öğrenme mimarileri üzerine yoğunlaşmıştır. İnsan bilgisayar etkileşimi için önemli bir çalışma alanı olan el tanıma üzerine, görüntü işleme alanındaki başarısından dolayı derin öğrenme mimarileri birçok araştırmacı için çalışma konusu olmuştur.

Çizelge 3.1’de el tanıma üzerine derin öğrenme ile çalışma yapan araştırmacılar, çalışma konuları ve kullanılan derin öğrenme mimarileri görülmektedir.

Çizelge 3.1. El tanıma üzerine derin öğrenme modelleri ile yapılan çalışmalar.

Yazar/ Kaynak	Kullanılan Yöntem/ Derin Öğrenme Mimarisi	Çalışma Konusu
[73]	3B-CNN	Araç sürücülerinin el hareketlerini tanımak
[74]	Derin İnanç Ağları & CNN	Kinect sensörü kullanarak işaret dili tanıma
[75]	Derin Dinamik Sinir Ağları	Görüntü ve derinlik verilerini ve iskelet özelliklerini içeren çoklu tip veriler üzerinde sürekli hareket tanıma
[76]	Tekrarlayan sinir ağı (Recurrent Neural Network-RNN)	Çok tipli verilerden dinamik el hareketlerinin eşzamanlı olarak tespitini ve sınıflandırılması
[77]	Uzun Süreli RNN	Akıllı araçlar için görüş tabanlı bir el hareketi tanıma sistemi
[78]	Beklenti maksimizasyonu & CNN	Çok tipli işaret dili tanıma
[79]	Bölgeye Dayalı CNN	Açık ve kapalı konumda iki tip el hareketi tanıma
[80]	Derin CNN	Bütün bir görüntüden sadece el hareketi tanıma
[81]	3B RNN	Kask ve gözlük gibi giyilebilir cihazlarla etkileşim
[82]	CNN	Renkli kamera ile elde edilen görüntülerde el duruşlarının tanınması
[83]	CNN	El bölgesi konumu ve cilt tespiti
[84]	İki aşamalı CNN	Statik El Hareketi Tanıma
[85]	CNN	El-iskelet verisi kullanan ve derinlik görüntüsü olmayan 3B el tanıma
[86]	Ayrık Fourier Dönüşümü & CNN	Etkinlik görüntülerinin oluşturulmasına dayanan kol hareketi tanıma
[87]	CNN	El pozisyonu tahmini ve el işareti tanıma
[88]	Sürekli dalgacık dönüşümü - CNN	Elektromiyografi tabanlı hareket tanıma
[89]	Derin CNN	Minyatür bir radar sensörü ile 14 farklı el hareketinin Doppler imzalarını alarak tanıma
[90]	Hızlı R-CNN Nesne Tanıma	İşbirlikçi Robotlar için ayarlanmış akıllı el hareketi tanıma
[91]	CNN+LSTM (Long short-term memory)	Hareket kontrolörü ile oluşturulan çok çeşitli hareketlerden oluşan Karma El Hareketi Tanıma
[92]	CNN	El Hareketi Tanıma
[93]	CNN	Bistatik radar sistemi kullanarak el hareketi tanıma

Son yıllarda el hareketi tanıma ile ilgili literatürde yapılan çalışmaları incelediğimizde, çalışmaların [73-93] genel olarak Görme tabanlı ve Sensör tabanlı olmak üzere iki türe ayrıldığını gördük. Sensör temelli çalışmalarda [92, 93] hareketler özel sistemler (radar,

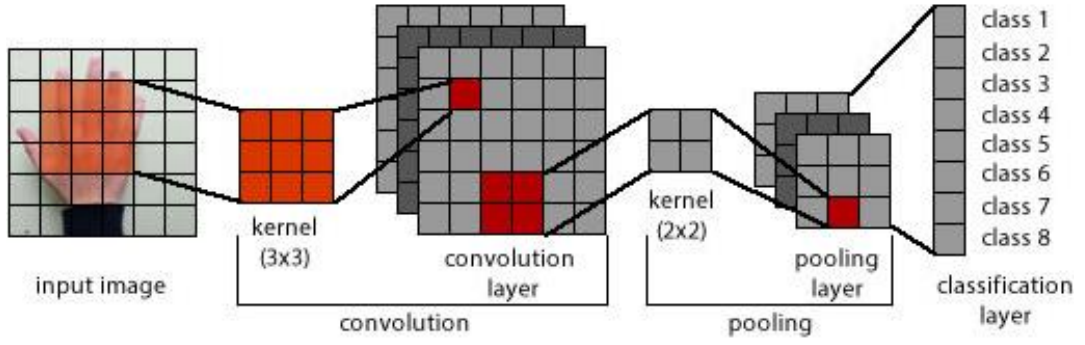
EMG vb.) İle tanımlanmış ve başarılı sonuçlar alınmıştır. Görme temelli çalışmalar; hareket detektörleri, kızılötesi, derinlik vb. Kullanılarak yapılan hareket tanıma çalışmaları olarak ikiye ayrılır. Kameralar ve normal kameralarla doğal çalışmalar. El ve hareket tespiti için hareket tanıma sistemlerinde [73-77, 82, 88-90] kullanılan kızılötesi ışınlar, derinlik bilgileri vb. Bu sayede yüksek başarı oranlarına sahip çalışmalar yapılmaktadır. Sadece kamera kullanılarak yapılan doğal el tanıma çalışmalarında [78-81,83-87,91], geliştirilen derin öğrenme modeli yapısına göre yüksek performans elde edildiği görülmektedir. Çalışmalar boyunca daha önce geliştirilen el veri setlerinin kullanıldığı görülmüştür.

CNN'in derin öğrenmede yaygın olarak kullanılması ve birçok araştırmada üstün başarılar elde etmesine rağmen, yetersiz kaldığı alanlar da olmaktadır. Bu alanlardan biri, bir görüntünün eğitildiği açıdan farklı açılardaki ya da farklı yapıdaki görüntülerini tanımakta düşük başarılar elde edilmesidir. Ayrıca evrişim işleminde uygulanan havuzlama katmanlarında görüntüye ait verilerin boyut küçültme amaçlı olarak atılmasıdır. Bu sorunların üzerinden gelmek için, 2017 yılında kapsül ağlar ve dinamik yönlendirme yaklaşımı önerilmiştir [94].

3.1. EVRİŞİMSEL SİNİR AĞLARI

Evrişimsel Sinir Ağı, özellikle bilgisayarlı görme ve nesne tanımlama problemlerinde yaygın olarak kullanılan derin ileri beslemeli yapay sinir ağları mimarisidir. 1990'larda LeCun ve ark. [60] tarafından CNN'lere gradyan tabanlı bir öğrenme algoritması uygulanması ve el yazısı rakam sınıflandırma probleminde başarılı sonuçlar elde edilmesinden sonra, araştırmacılar için yeni bir çalışma alanı olan CNN, her geçen gün geliştirilmiş ve son teknoloji ürünler ile kullanılmaya başlanmıştır. CNN'lerin, insan görsel işleme sistemine daha çok benzemesi, 2B ve 3B görüntülerin işlenmesi için yüksek düzeyde optimize edilmesi ve 2B özelliklerin soyutlamalarının öğrenilmesi ve çıkarılmasında etkili olması gibi çeşitli avantajları yaygın olarak kullanılmasını sağlamaktadır [95].

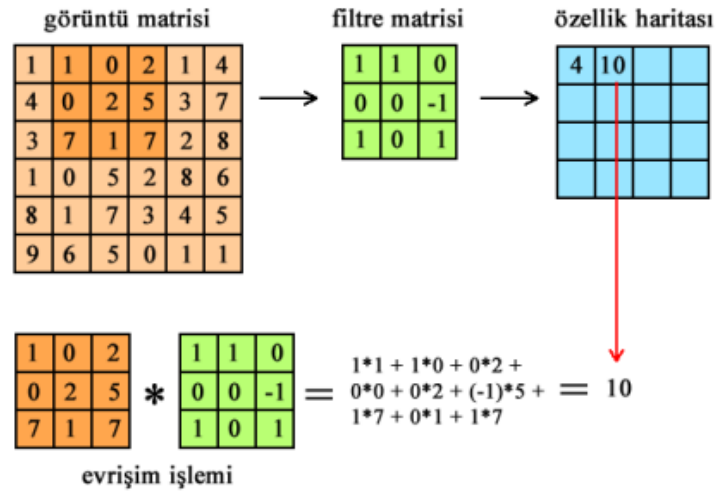
Görüntü tanıma ve sınıflandırma çalışmalarında yaygın olarak kullanılan çok katmanlı sinir ağı olan CNN mimarisi temel olarak Evrişim (Convolution), Havuzlama (Pooling), Aktivasyon, Dropout, Tambağı ve Sınıflandırma (Classification) katmanlarından oluşur [96-97]. Şekil 3.1'de evrişimsel sinir ağına ait örnek bir model verilmiştir.



Şekil 3.1. Evrişimsel sinir ağı örnek model.

3.1.1. Evrişim Katmanı

Görüntü sayısı, görüntü boyutu ve filtre boyutu parametrelerini içeren evrişim katmanında, matris olarak ifade edilen bir önceki katmandan gelen giriş verisi üzerinde filtre matrisi gezdirilerek evrişim işlemi uygulanır ve yeni bir özellik haritası elde edilir [69,96]. Şekil 3.2’de giriş verisi üzerine filtre uygulanması gösterilmiştir.

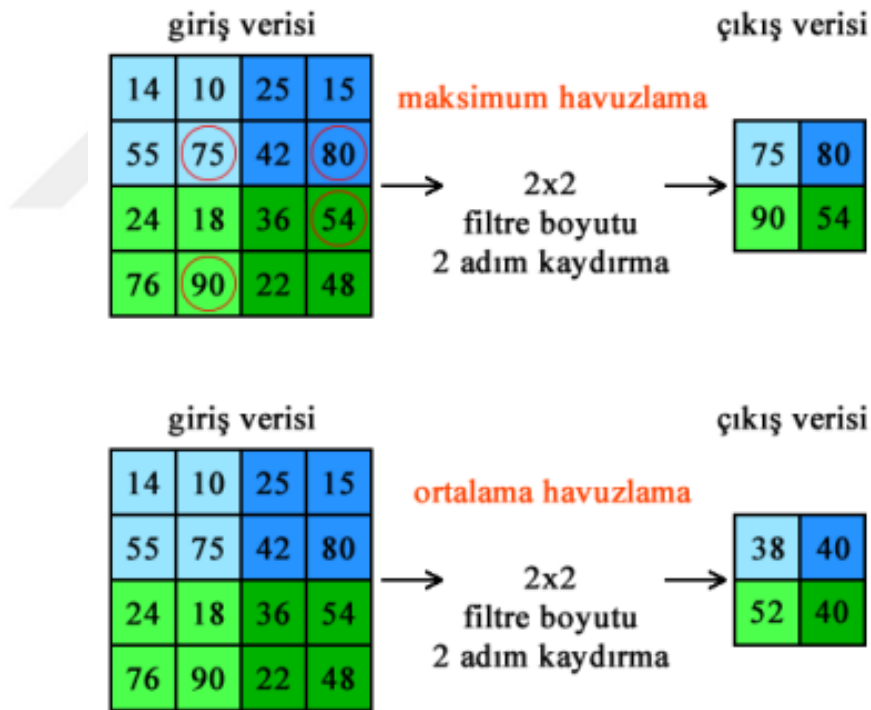


Şekil 3.2. Evrişim işlemi.

Şekil 3.2’de 6×6 boyutlarında bir giriş verisi matrisi üzerinde 3×3’lük bir filtre matrisi ile evrişim işlemi uygulanmıştır. Filtre matrisi, giriş matrisi üzerinde adım kaydırma işlemi ile belirlenen adım aralığında ilk olarak yatayda sağa kaydırılarak uygulanır. Son sütuna gelindiğinde ise alt satıra geçerek tekrar adım kaydırma işlemi ile ikinci satır elde edilir. Bu işlem son satır son sütuna kadar devam eder. İşlem sonucunda özellik haritası çıkarılmış olur.

3.1.2. Havuzlama Katmanı

Havuzlama katmanında, evrişim katmanında özellikleri çıkartılan görüntüye ait özellik haritası boyutu maksimum, ortalama havuzlama yöntemleri kullanılarak bir sonraki evrişim katmanı için azaltılır [69,98]. Bu işlem aşağı örnekleme olarak da adlandırılır. Havuzlama katmanında kullanıcı tarafından belirlenen $n \times n$ boyutlarında bir boyutlandırma matrisi uygulanır. Bu boyutlandırma matrisi görüntü üzerinde adım kaydırma değerine göre gezdirilir. Matris içerisinde yer alan maksimum değerler alınır (maksimum havuzlama) veya değerlerin ortalaması alınır (ortalama havuzlama) yöntemi uygulanmış olur [94,99]. Daha iyi performans gösterdiği için maksimum havuzlama yöntemi daha çok kullanılır. Şekil 3.3'te maksimum havuzlama ve ortalama havuzlama işlemleri ile ilgili örnek verilmiştir. Şekil 3.3'te giriş verisi boyutu 4×4 ve filtre boyutu 2×2 ve adım kaydırma 2 olarak alınmıştır. Bu şekilde oluşan yeni görüntü boyutu 2×2 olmuştur.

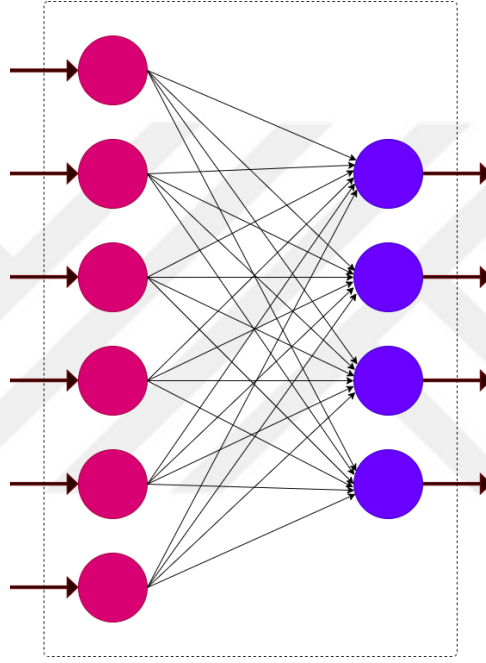


Şekil 3.3. Havuzlama işlemi ile boyut azaltma.

Bu katman sonucunda boyutta azalma olduğu için bilgi kaybı oluşur. Oluşan bilgi kaybı, sinir ağı için bir sonraki ağı katmanlarında daha az hesaplama yükü oluşturması ve sistemin ezberlemesini önlediği için faydalıdır [99].

3.1.3. Tam Bağlı Katman

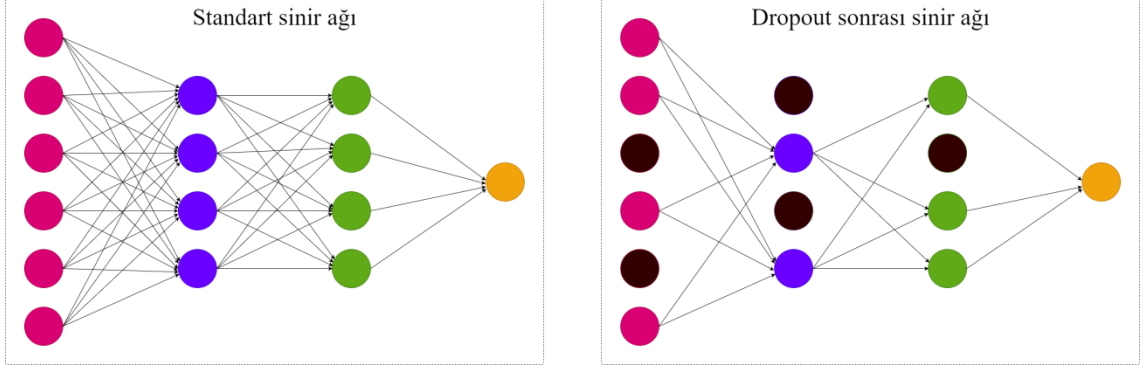
Tam bağlı katmanlar (FullyConnected - FC) sınıflandırma amacıyla sinir ağının sonunda kullanılır. Tam bağlı katman kendinden önce gelen katmanın tüm alanlarına bağlıdır ve bu katmandaki verileri tek boyutlu bir matris şeklinde düzenler. Verilerin sınıflandırılması için kullanılan seçili özelliklerin doğrusal olmayan bir kombinasyonunu oluşturur [100]. Bu katmanda nöronlar kendinden sonra gelen her nörona birebir bağlanır ve bu nedenle bu katman tam bağlı katman olarak adlandırılır [101].



Şekil 3.4. Tam bağlı katman yapısı.

3.1.4. DropOut

Evrişimsel sinir ağlarında modelin aşırı öğrenmesi yani veriyi ezberlemesi istenmeyen bir durumdur. Aşırı öğrenmeyi önlemek amacıyla dropout katmanı uygulanır. Dropout işleminde, eğitim sırasında gizli katmanlar arasındaki bağlantılar, belirlenen oranda silinir ve kalan ağırlıklar yeniden eğitilir [102].



Şekil 3.5. Dropout katmanı.

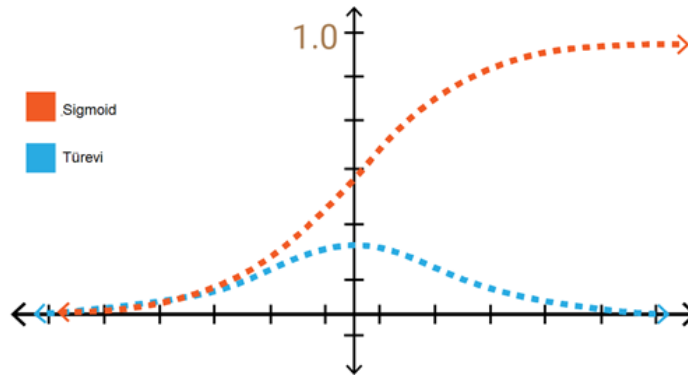
3.1.5. Aktivasyon fonksiyonu

Evrişimsel sinir ağlarında nöronun giriş değerine göre doğrusal olmayan çıkış değeri aktivasyon fonksiyonları ile hesaplanır. Hesaplanan çıkış değeri bir sonraki katmanın girişi olarak kullanılır. Aktivasyon fonksiyonları, çıkış değerinin, bir sonraki katmana iletilip ileilmeyeceğine ve hangi nöronun aktif hangi nöronun pasif olması gerektiğine karar verebilmek için kullanılır [103]. Evrişimsel sinir ağının başarılı olması için kullanılacak aktivasyon fonksiyonunun önemi büyüktür. Derin öğrenme çalışmalarında sıklıkla kullanılan aktivasyon fonksiyonları, matematiksel ifadeleri ve grafiksel gösterimleri aşağıda belirtilmiştir [104,105].

3.1.5.1. Sigmoid Fonksiyonu

(0, 1) arasında çıktı üreten doğrusal olmayan bir fonksiyondur. Fonksiyona ait formül ve grafik aşağıda verilmiştir.

$$f(x) = \sigma(x) = \frac{1}{1+e^{-x}} \quad (3.1)$$

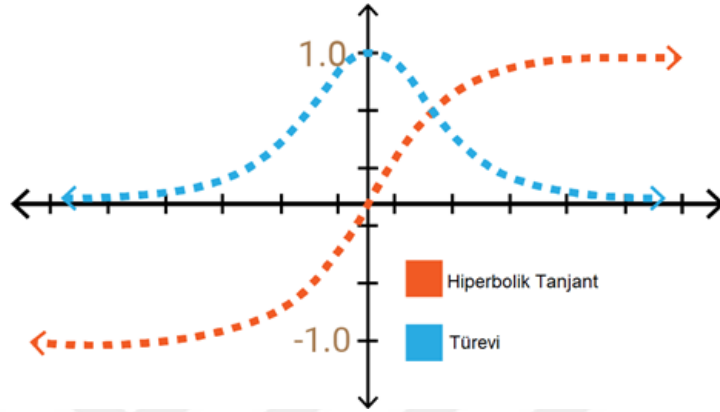


Şekil 3.6. Sigmoid Fonksiyonu [104,105].

3.1.5.2. Hiperbolik Tanjant (tanh) Fonksiyonu

(-1,1) aralığında çıktı üreten doğrusal olmayan bir fonksiyondur. Fonksiyona ait formül ve grafik aşağıda verilmiştir.

$$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})} \quad (3.2)$$

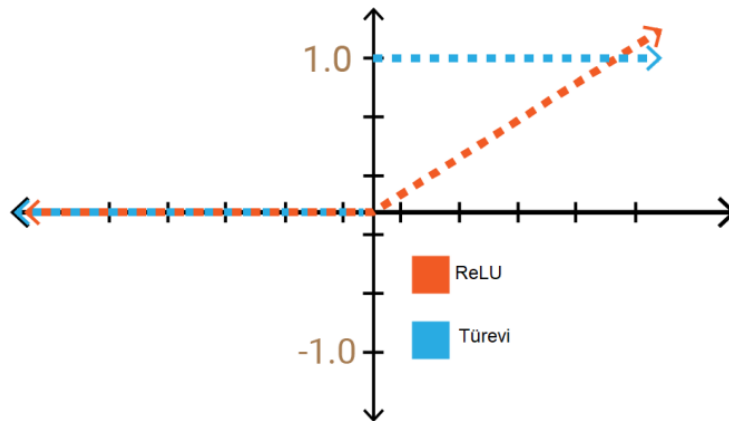


Şekil 3.7. Hiperbolik Tanjant Fonksiyonu [104,105].

3.1.5.3. Doğrultulmuş Lineer Birim (Rectified Lineer Unit - ReLU) Fonksiyonu

$[0, +\infty)$ aralığında çıktı üretir. $x < 0$ olduğunda 0, $x \geq 0$ olduğunda x çıktısı üreten doğrusal olmayan bir fonksiyondur. Fonksiyona ait formül ve grafik aşağıda verilmiştir.

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3.3)$$



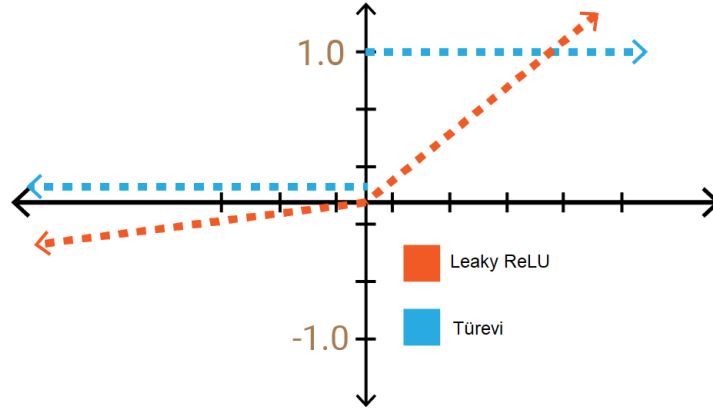
Şekil 3.8. ReLU Fonksiyonu [104,105].

3.1.5.4. Sızıntı (Leaky) ReLU Fonksiyonu

$(-\infty, +\infty)$ aralığında çıktı üreten, negatif değerlerin sıfıra çok yakın ama sıfır olmadığı fonksiyondur. Böylece türevinin sıfır olması engellenmiş olur. Bu sayede öğrenmenin

negatif tarafta da gerçekleşmesi sağlanmış olur. Fonksiyona ait formül ve grafik aşağıda verilmiştir.

$$f(x) = \begin{cases} 0.01, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3.4)$$

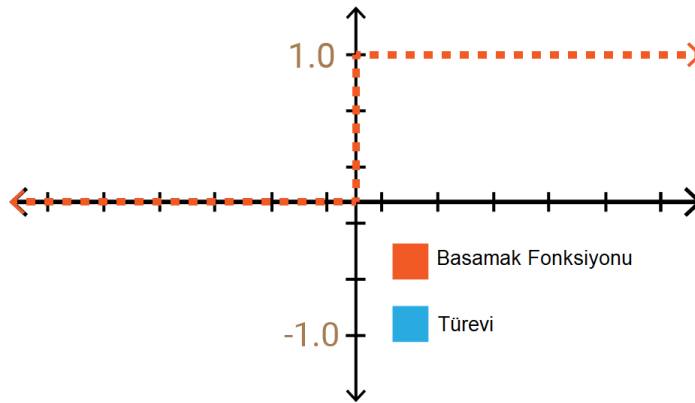


Şekil 3.9. Sızıntı ReLU Fonksiyonu [104,105].

3.1.5.5. Basamak Fonksiyonu

0 yada 1 değeri üreten, ikili değer alan bir fonksiyondur ve ikili sınıflandırma durumunda kullanılır. Fonksiyona ait formül ve grafik aşağıda verilmiştir.

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (3.5)$$

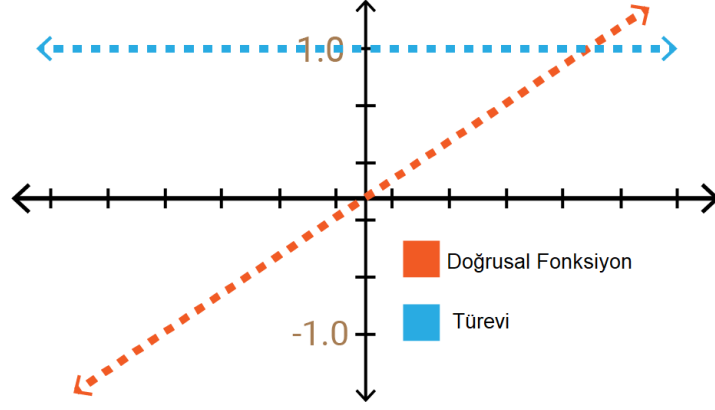


Şekil 3.10. Basamak Fonksiyonu [104,105].

3.1.5.6. Doğrusal Fonksiyon

$(-\infty, +\infty)$ arasında çıktı üreten doğrusal bir fonksiyondur. Fonksiyona ait formül ve grafik aşağıda verilmiştir.

$$f(x) = x \quad (3.6)$$

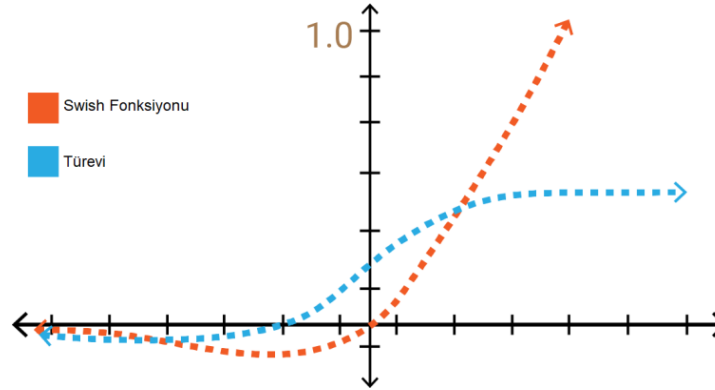


Şekil 3.11. Doğrusal Fonksiyonu [104,105].

3.1.5.7. Swish Fonksiyon

$(-\infty, +\infty)$ arasında çıktı üreten, negatif bölgede doğrusal olmayan bir fonksiyondur. Fonksiyona ait formül ve grafik aşağıda verilmiştir.

$$f(x) = 2x\sigma(\beta x) = \begin{cases} \beta = 0 \text{ için } f(x) = x \\ \beta \rightarrow \infty \text{ için } f(x) = 2\max(0, x) \end{cases} \quad (3.7)$$



Şekil 3.12. Swish Fonksiyonu [104,105].

3.1.6. Sınıflandırma katmanı

Sınıflandırma katmanı, tam bağlı katmanlardan sonra gelir ve sınıflandırması yapılacak nesne sayısı kadar çıkış üretilir. Sınıflandırma katmanında en çok softmax sınıflandırıcı kullanılır. 10 farklı nesneye ait bir sınıflandırma yapılacaksa, çıkış değeri 10 olmalıdır. Sınıflandırma katmanında her bir nesne için 0-1 aralığında bir ağırlığa sahip çıkış üretilir ve değeri 1'e en yakın olan çıkışın tahmin edilen nesne olduğu anlaşılır.

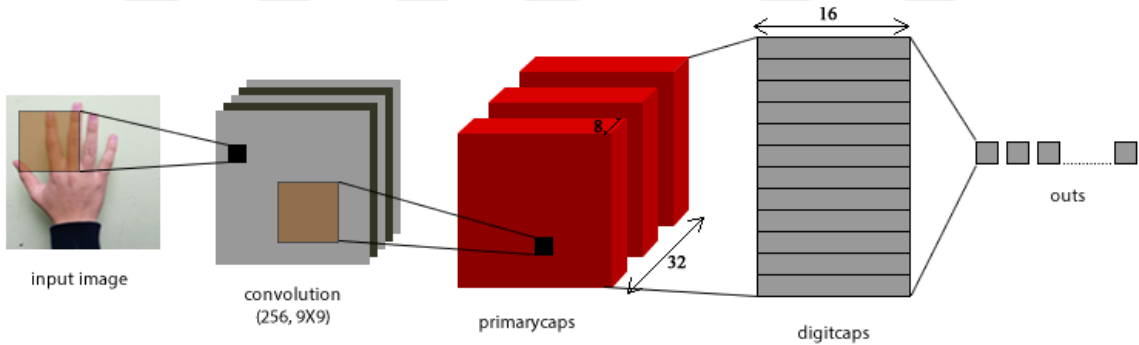
3.2. KAPSÜL AĞLAR VE DİNAMİK YÖNLENDİRME

Kapsül ağlarda, CNN'in başarısız olduğu görüntüye ait farklı açısız değerdeki

görüntülerin de başarıyla tanınabilmesi için bir grup nörondan oluşan kapsüller aracılığıyla nesneyi temsil eden boyut, kaydırma, konum, ölçekleme vb. özelliklerin vektörel olarak anlaşılmalı yönlendirme ile öğrenilmesi önerilmiştir [94].

Kapsülün çıkışları bir aktivasyon vektörü oluşturur. Aktivasyon vektörünün yönü, nesnenin konum ve yön gibi pozlama bilgilerini belirlerken, aktivasyon vektörünün uzunluğu (diğer bir deyişle norm veya büyüklük), ilgili nesnenin tahmini olasılığını belirtir [106]. Örneğin, bir görüntüyü döndürürsek, aktivasyon vektörleri de buna göre değişir, ancak uzunlukları aynı kalır. Alt seviye kapsüllerin ($u_1, u_2, \dots, U_I$) çıktılarının uzunlukları, karşılık gelen varlıklarının (örneğin gözler, burun ve ağız) var olma olasılığını kodlar. Vektörlerin yönleri aynı zamanda belirli bir varlığın büyüklüğü, yönü, konumu vb. çeşitli özelliklerini kodlar [106].

Alt seviye kapsüllerde vektörler hesapladıktan sonra elde edilen tahmin değerleri, tahminleriyle en çok uyuşan üst seviye kapsüllere yönlendirilir, böylece daha temiz bir giriş verisi ile nesnelerin pozu daha doğru bir şekilde belirlenir. Bu işleme Dinamik Yönlendirme adı verilir [96,106]. Şekil 3.13’de kapsül ağına ait örnek bir model verilmiştir.



Şekil 3.13. Kapsül ağı örnek model.

Kapsül ağlarda, ilk katman evrişim katmanıdır. Evrişim katmandan sonra elde edilen çıkış değeri kapsül ağlarda öncül kapsülün giriş değeri olarak uygulanır. Çıkış değerlerini vektör değerlere dönüştürmek için yeniden boyutlandırma işlemi uygulanır. Bu vektörlerin değerlerini sınırlandırmak için CNN yapıda kullanılan havuzlama yöntemi yerine ezme yöntemi kullanılır. Yönlendirme algoritması aşağıda belirtilmiştir:

Algoritma 1 Yönlendirme Algoritması [94].

“procedure ROUTING($\hat{u}_{ji}, r, l$)

for all capsule i in layer l and capsule j in layer $(l + 1)$: $b_{ij} \leftarrow 0$.

```

for  $r$  iterations do
for all capsule  $i$  in layer  $l$ :  $c_i \leftarrow \text{softmax}(b_i)$  . softmax computes Eq. 3
for all capsule  $j$  in layer  $(l + 1)$ :  $s_j = \sum_i c_{ij} \hat{u}_{j|i}$ 
for all capsule  $j$  in layer  $(l + 1)$ :  $v_j \leftarrow \text{squash}(s_j)$  . squash computes Eq. 1
for all capsule  $i$  in layer  $l$  and capsule  $j$  in layer  $(l + 1)$ :  $b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \cdot v_j$ 
return  $v_j$ ”

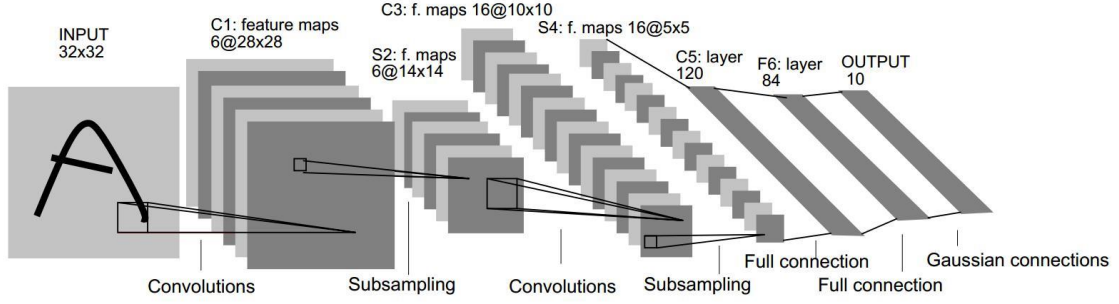
```

3.3. DERİN ÖĞRENME MODELLERİ

Derin öğrenme 2012 yılında ImageNet tarafından yapılan ve aynı isimle anılan Büyük Ölçekli Görsel Tanıma Yarışması (ImageNet Large Scale Visual Recognition Challenge- ILSVRC) yarışması ile popüler olmaya başlamış, geniş kitlelere ulaşmıştır. Daha öncesinde derin öğrenmenin temelini oluşturan ilk çalışma 1998 yılında Lecun ve arkadaşları tarafından 0’den 9’a kadar el ile yazılan rakamları sınıflandırmak amacıyla kullanılan gradyan temelli evrimsel sinir ağı olan LeNet adlı modeldir [107]. 2012 yılında başlatılan ILSVRC, araştırmacılara nesne tanıma modellerini eğitmek için görüntü verileri sağlamak amacıyla devam eden bir araştırma projesidir. ILSVRC tarafından yapılan yarışmada görüntü tanıma için katılımcıların çoğu modellerini derin öğrenme mimarisine göre tasarlamış ve yeni derin öğrenme modelleri önermişlerdir. Bu önerilen modellerin başarıları derin öğrenme mimarilerinin ön plana çıkmasını ve yaygınlaşmasını sağlamıştır. ILSVRC yarışmasını kazanan ve derin öğrenme çalışmaları için önemli bir yere sahip olan bu modeller, araştırmacılar tarafından nesne tanıma ve görüntü işleme alanında yaygın olarak kullanılmıştır. Bu bölümde yaygın olarak kullanılan LeNet, AlexNet, ZfNet, ResNet, VGG, GoogLeNet, DenseNet, MobileNet, InceptionV3 ve CapsNet modelleri hakkında bilgiler verilmiştir.

3.3.1. LeNet

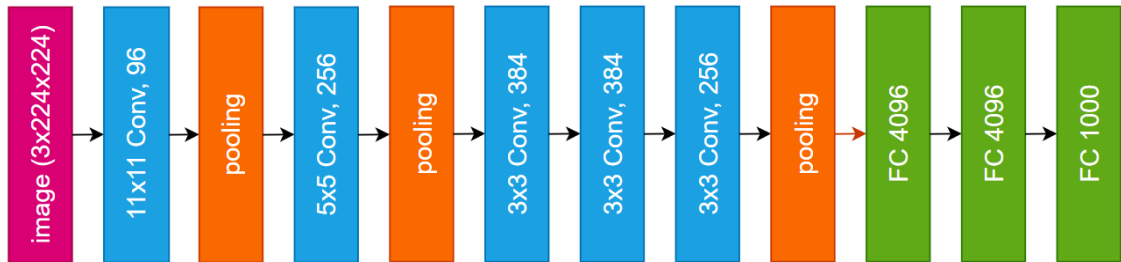
[60]’da önerilen ve 1998 yılında yayınlanan, 0’den 9’a kadar olan el ile yazılan rakamlarının sınıflandırılmasını sağlayan gradyan temelli evrimsel sinir ağları kullanılarak tasarlanan ilk temel derin öğrenme ağıdır. LeNet-5 modeli 2 evrimsel katmandan, 2 havuzlama katmanından ve 2 tamamen bağlı katmandan oluşur. Boyut azaltma için ortalama havuzlama yöntemi, aktivasyon fonksiyonu olarak sigmoid ve hiperbolik tanjant fonksiyonları kullanılmaktadır. LeNet mimarisi Şekil 3.14’de gösterilmiştir.



Şekil 3.14. LeNet mimarisi [60].

3.3.2. AlexNet

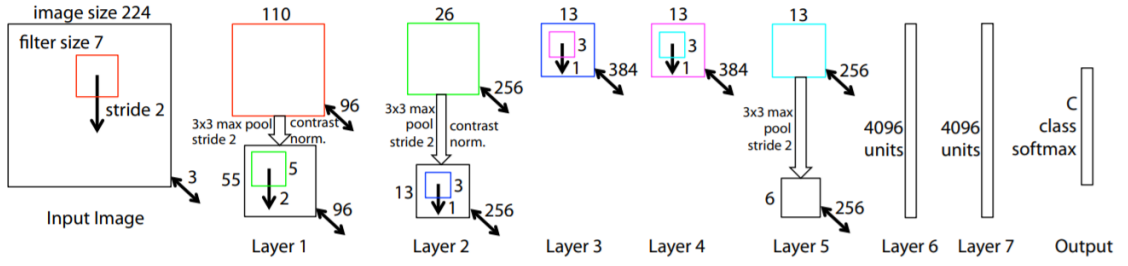
[61]'de önerilen, 2012 yılında düzenlenen ILSVRC yarışmasını kazanan derin öğrenme modelidir. ILSVRC yarışmasında elde ettiği başarı ile derin öğrenme ve evrişimsel sinir ağlarının yaygınlaşmasını sağlamıştır. Yapısında sıra ile gelen evrişim ve havuzlama katmanları bulunmasından dolayı LeNet modeline benzemektedir. AlexNet modeli 5 evrişimli katmandan, 3 havuzlama katmanından ve 2 tamamen bağlı katmandan oluşur. Havuzlama katmanında maksimum havuzlama yöntemi, aktivasyon fonksiyonu olarak ReLU ve sınıflandırma katmanında softmax aktivasyon fonksiyonu kullanılmaktadır. AlexNet modeli Şekil 3.15'de gösterilmiştir.



Şekil 3.15. AlexNet mimarisi.

3.3.3. ZFNet

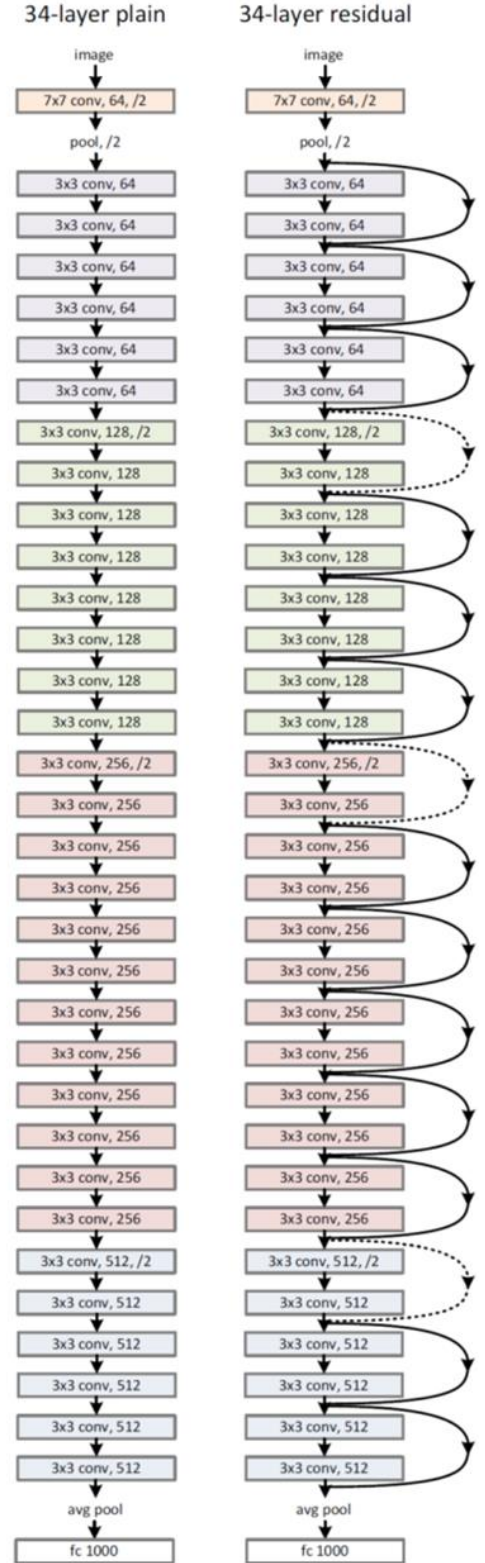
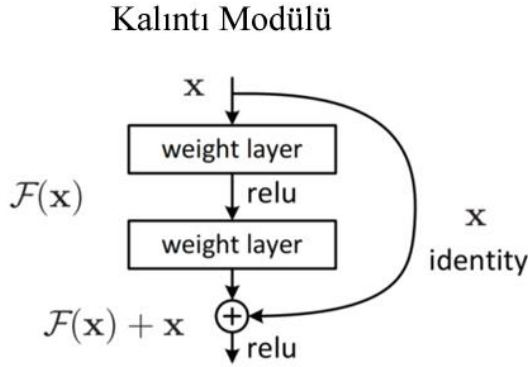
[62]'de önerilen ve 2013 yılında düzenlenen ILSVRC yarışmasını kazanan derin öğrenme modelidir. AlexNet modelinden farkı katmanlarda uygulanan filtre boyutunun azaltılması ve adım kaydırmanın azaltılmasıdır. Aktivasyon fonksiyonu olarak ReLU, havuzlama katmanlarında maksimum havuzlama yöntemi ve hata fonksiyonu için Cross-Entropy Loss fonksiyonu kullanılmıştır. ZFNet modeli 5 evrişimli katmandan, 3 havuzlama katmanından ve 2 tamamen bağlı katmandan oluşur. ZFNet modeli Şekil 3.16'da gösterilmiştir.



Şekil 3.16. ZFNet mimarisi [62].

3.3.4. ResNet

Microsoft araştırma ekibi tarafından geliştirilen ResNet modeli, 2015 yılında düzenlenen ILSVRC yarışmasını %95.51 başarı oranı ile kazanmıştır [63]. ResNet derin ağlar tasarlamak niyetiyle geliştirilmiştir. ResNet modelinin 18, 34, 50, 101, 152 katmana sahip olan türleri vardır. ResNet modelinde bir önceki katmanın ağırlık değerleri bir sonraki katmana direkt olarak aktarılmıştır. Bu mimarinin en önemli özelliği kalıntı bloklarından (Residual Block) oluşmasıdır. Kalıntı bloklarında, x girişine evrişim, ReLU aktivasyon fonksiyonu ve tekrar evrişim işlemi uygulanmasından sonra bir $F(x)$ sonucu hesaplanmaktadır. Elde edilen $F(x)$ sonucuna ağın ilk giriş değeri olan x değeri eklenir ve $H(x) = F(x) + x$ olarak çıkış değeri bulunur. Şekil 3.17’de kalıntı modülü ile 34 katmanlı klasik CNN modeli ve 34 katmanlı ResNet model yapısı görülmektedir.



Şekil 3.17. Kalıntı modülü,34 katmanlı Klasik CNN modeli ile ResNet mimarisi [63].

3.3.5. VGGNet

Oxford Üniversitesi'nde bulunan Görsel Geometri Grubu (Visual Geometry Group - VGG) tarafından önerilen, 2014 yılında düzenlenen ILSVRC yarışmasında ikinci olan

derin öğrenme modelidir [64]. VGGNet, yaygın olarak kullanılan, biri 16 diğeri 19 katmandan oluşan VGG16 ve VGG19 adlı iki farklı modele sahiptir. VGG16 modelinde 13 evrişim katmanı ve 3 tam bağlı katman bulunurken, VGG19 modelinde ise 16 evrişim katmanı ve 3 tam bağlı katman yer almaktadır. VGGNet modelinde havuzlama katmanında maksimum havuzlama yöntemi, aktivasyon fonksiyonu olarak ReLU, ve sınıflandırma katmanında softmax fonksiyonu kullanılmıştır. VGGNet modeli Şekil 3.18’de gösterilmiştir.

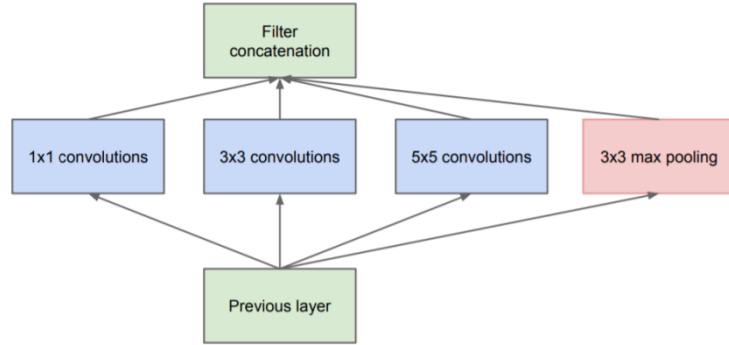
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Şekil 3.18. VGGNet mimarisi [64].

3.3.6. GoogLeNet

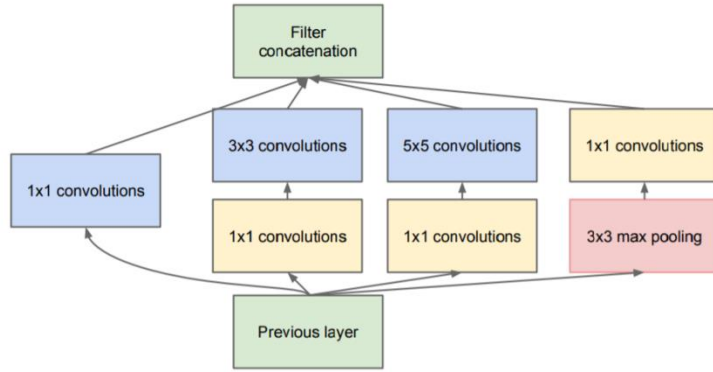
[65]’de geliştirilen ve 2014 yılında düzenlenen ILSVRC yarışmasını %93,33 başarı oranı ile kazanan evrişimsel sinir ağı modelidir. Yapısında inception modülü olarak adlandırılan başlangıç modülü bulunan evrişimsel sinir ağı modeli 22 katman derinliğe sahip olup 144 katmandan oluşmaktadır. Inception modülü ile 1×1, 3×3, 5×5 filtre boyutlarında evrişim işlemi ve 3×3 boyutunda maksimum havuzlama işlemi yapılır.

Inception modülü ile daha önce önerilen diğer derin öğrenme mimarilerinden farklı bir yapı ortaya koymuştur. Şekil 3.19’da inception modülü gösterilmiştir.



Şekil 3.19. Inception modülü [65].

Inception modülünde evrişim katmanında 1×1 , 3×3 , 5×5 filtre boyutlarında evrişim işlemi ve 3×3 boyutunda maksimum havuzlama işlemi uygulanmaktadır. Inception modülünde evrişim işlemi, filtre boyutu ve havuzlama işlemleri paralel olarak gerçekleştirilmektedir. Şekil 3.20’de Boyut düşürme işleminden sonra inception modülü yer almaktadır.

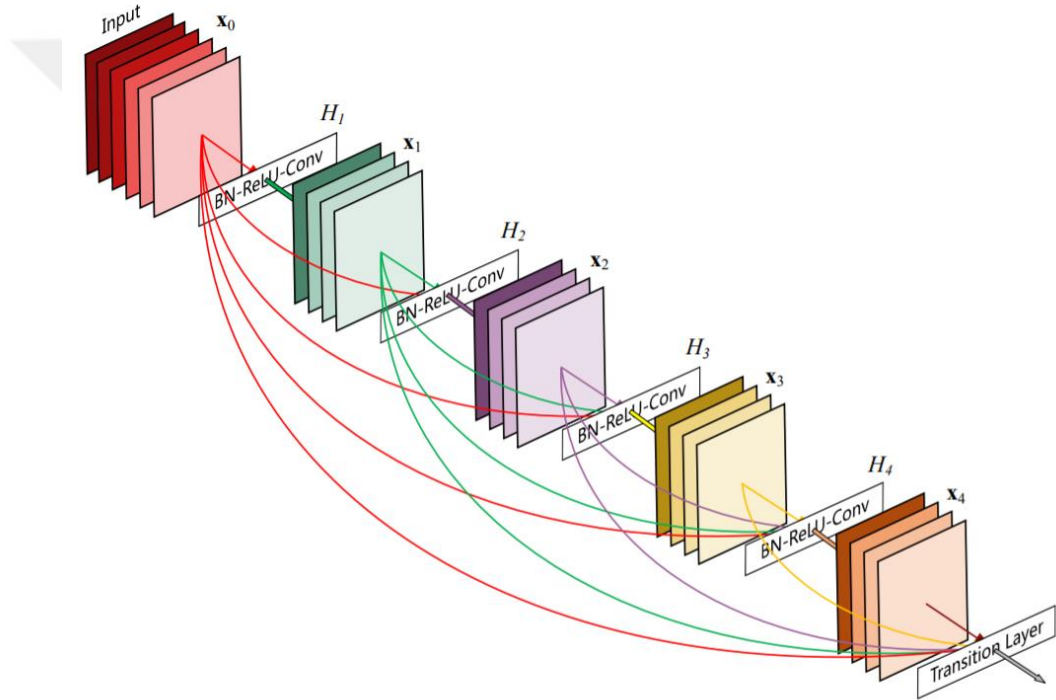


Şekil 3.20. Boyutu düşürme sonrası inception modülü [65].

Şekil 3.20’de gösterildiği gibi Inception modülü içerisine hesaplama maliyetini düşürmek için 1×1 boyutunda evrişim katmanı eklenerek parametre sayısını 10 kat azaltılmaktadır. Şekil 3.21’de GoogLeNet mimari yapısı görülmektedir.

3.3.7. DenseNet

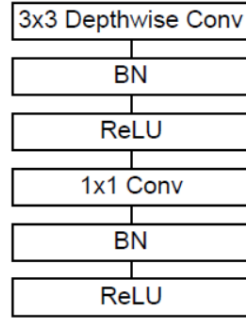
DenseNet mimarisinde, her katman kendinden sonra gelen diğer katmanlara ileri beslemeli bir şekilde bağlanır [66]. DenseNet mimarisinde her katman kendinden önce gelen tüm katmanların özelliklerini giriş değeri olarak kullanmakta ve kendi katmanındaki özelliklerini de sonraki katmanlara giriş değeri olarak göndermektedir. L katmanlı geleneksel evrişimli sinir ağlarında L bağlantı bulunurken, DenseNet ağı $L(L + 1) / 2$ doğrudan bağlantıya sahiptir. Her katman için, önceki tüm katmanların öznitelik haritaları girdi olarak, öznitelik haritaları ise sonraki tüm katmanlarda girdi olarak kullanılır. Şekil 3.22’de 5 katmanlı DenseNet model yapısı görülmektedir.



Şekil 3.22. DenseNet mimarisi [66].

3.3.8. MobileNet

MobileNet, mobil cihazlarda ve gömülü sistemlerde görüntü sınıflandırma uygulamalarında kullanılmak amacıyla geliştirilmiş, hafif derin sinir ağları oluşturmak için derinlemesine ayrılabilir kıvrımlar kullanan modern bir sinir ağı mimarisidir [67]. MobileNet yapısı, tam bir evrişim olan birinci katman ve derinlemesine ayrılabilir evrişim katmanlarından oluşmaktadır. Şekil 3.23’te ayrılabilir ve noktasal evrişim katmanları yapısı görülmektedir.



Şekil 3.23. Ayrılabilir ve noktasal evrişim yapısı [67].

Ayrılabilir evrişim katmanlarında her bir giriş verisi için tek bir filtre uygulanır. Daha sonra noktasal evrişim için 1×1 evrişimler kullanılarak derinlik bazlı katmanların çıktısının doğrusal kombinasyonu oluşturulur. Aşağı örnekleme, birinci katmanda olduğu gibi, derinlikli evrişimlerde de adım adım evrişim ile işlenir. Sınıflandırma için bir softmax katmanına beslenen son tam bağlı katman haricindeki tüm katmanlarda ReLU aktivasyon fonksiyonu kullanılır. Derinlemesine ve noktasal kıvrımları ayrı katmanlar olarak sayan MobileNet, 28 katmana sahiptir [67]. Şekil 3.24'te MobileNet mimari yapısı görülmektedir.

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5x	Conv dw / s1	$3 \times 3 \times 512$ dw
	Conv / s1	$1 \times 1 \times 512 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Şekil 3.24. MobileNet mimarisi [67].

3.3.9. InceptionV2/InceptionV3

[68]'de önerilen InceptionV2 ve InceptionV3 modelleri, GoogLeNet (Inception) modeli üzerinde verimliliği artırmak için yapılan optimizasyon ve daha etkili kullanım yöntemlerini içermektedir. Inception modüllerinin boyutsal indirgeme ve paralel yapılarının daha etkili kullanımı için, evrişim katmanları daha küçük çarpanlara ayrılmış, asimetrik evrişim katmanlarını çarpanlara ayrılmış, bu sayede verimliliğini düşürmeden parametre sayısını azaltılmıştır.

type	patch size/stride or remarks	input size
conv	$3 \times 3 / 2$	$299 \times 299 \times 3$
conv	$3 \times 3 / 1$	$149 \times 149 \times 32$
conv padded	$3 \times 3 / 1$	$147 \times 147 \times 32$
pool	$3 \times 3 / 2$	$147 \times 147 \times 64$
conv	$3 \times 3 / 1$	$73 \times 73 \times 64$
conv	$3 \times 3 / 2$	$71 \times 71 \times 80$
conv	$3 \times 3 / 1$	$35 \times 35 \times 192$
$3 \times$ Inception	As in figure 5	$35 \times 35 \times 288$
$5 \times$ Inception	As in figure 6	$17 \times 17 \times 768$
$2 \times$ Inception	As in figure 7	$8 \times 8 \times 1280$
pool	8×8	$8 \times 8 \times 2048$
linear	logits	$1 \times 1 \times 2048$
softmax	classifier	$1 \times 1 \times 1000$

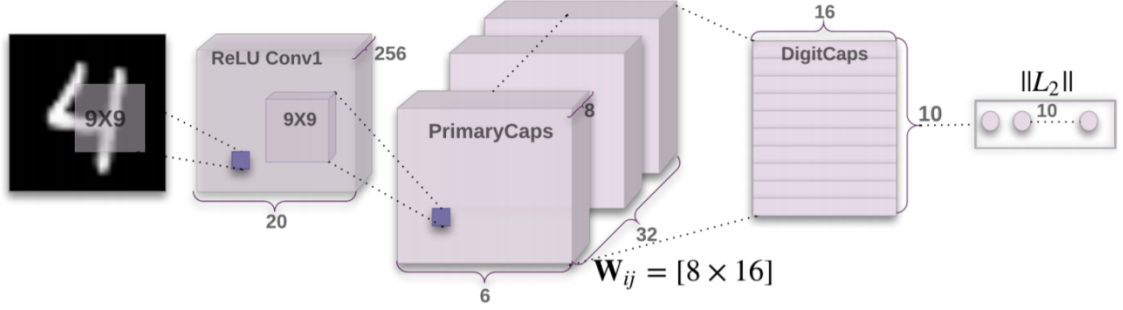
Şekil 3.25. InceptionV2 mimarisi [68].

InceptionV3 mimarisinin, InceptionV2 mimarisinden farkı evrişim katmanları, yardımcı sınıflayıcı olarak küme normalizasyonu uygulanması ve çıkışta FC katmanı kullanılmasıdır.

3.3.10. CapsNet

CapsNet mimarisinde derinlik, kapsüllerin tek bir katmanda birbirine bağlanmasıyla elde edilir. CapsNet sinir ağında, CNN 'de kullanılan nöron tabanlı işlemlerden farklı olarak, “kapsül” adı verilen ve vektör ile ifade edilen bir grup nörondan oluşan yapı kullanılmaktadır [94]. Bu nöron gruplarına kapsül adı verilmesinin sebebi, sinir ağında derinliğin diğer ağlardaki gibi birbirine bağlanan artık katmanlar ile değil iç içe bağlanan kapsüllerle elde edilmesidir. Bu sayede bir katman içerisinde birden çok kapsül katmanı oluşturulabilir.

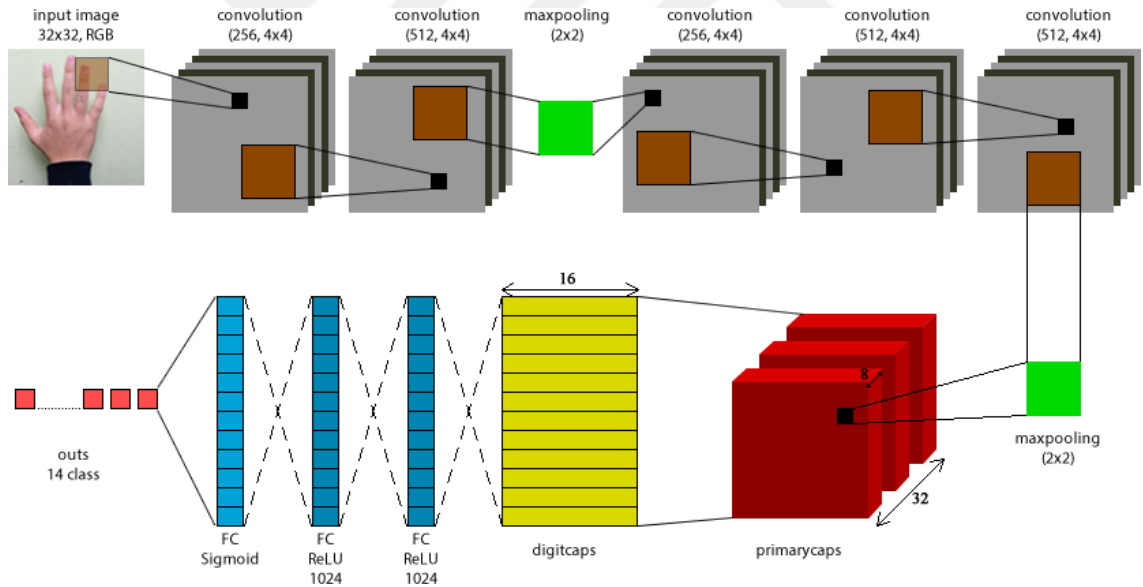
İlk katmanda bir evrişim katmanı vardır. Bu katmanın sonucunda üretilen özellik haritası öncül kapsüle gelir. Bu katmana 2 adım ve 32 katmandan oluşan yeni bir evrişim işlemi uygulanır. Çıkışlara ezme işlevi uygulanır. Daha sonra anlaşmaya göre köklendirme katmanı ile bilgilerin hangi kapsüllere gönderileceğine karar verilir. Şekil 3.25’de 5 katmanlı CapsNet model yapısı görülmektedir.



Şekil 3.26. CapsNet mimarisi [94].

4. ÖNERİLEN MODEL VE VERİSETİ

Derin öğrenme uygulamalarında CNN ve kapsül ağlar ile yapılan çalışmalarda yüksek başarılar elde edilmiştir. CNN yüksek seviyede başarı elde ederken, havuzlama katmanında uygulanan boyut azaltma ve piksel atma işlemi bir dezavantaj olarak görülmektedir. Farklı açılardan elde edilen görüntülerde ise kapsül ağların CNN mimarisine göre daha başarılı olduğu vurgulanmıştır [94]. Bu çalışmada CNN ve kapsül ağlarını birlikte kullanan bir model önerilmiştir. Kapsül ağlarda sadece ilk katman evrişim katmanı olurken, elde edilen sonuçlar kapsül ağın girişi olarak kullanılmaktadır. Önerilen modelde giriş verisi ilk olarak CNN mimaride eğitilerek, elde edilen sonuçlar kapsül ağa giriş olarak kullanılmıştır. Şekil 4.1’de önerilen mimariye ait model verilmiştir.



Şekil 4.1. Önerilen CNN & Kapsül ağı modeli.

Şekil 4.1 incelendiğinde eğitim modelimizin veri girişinin 32×32 RGB görüntüden oluştuğu görülmektedir. Girişte 32×32×3 giriş verisinin kullanılmasının nedeni, çalışmada kullanılan diğer veri seti büyüklükleri ile uyumlu olması ve aynı büyüklükteki verilerin karşılaştırılmasıdır. Önerilen model iki ana bölümden oluşmaktadır: CNN ve kapsül ağ katmanları. İlk bölüm, CNN mimarisi 5 evrişim katmanı ve 2 havuz katmanından oluşur. Evrişim katmanlarında hiperparametre

değerleri filtre boyutu = 4×4 , adım = 1, dolgu = 0 olarak alınmış ve doğrultulmuş doğrusal birim (ReLU) aktivasyon fonksiyonu uygulanmıştır. Birinci evrişim katmanında 256 filtre uygulanmış ve ikinci evrişim katmanında özellik çıkarımı için ReLU aktivasyonu ile 512 filtre uygulanmıştır. Alt örnekleme işlemi olarak özellik haritalarının düzlem içi boyutluluğunu azaltmak için bir Maxpooling katmanı eklenir. Maxpooling katmanından sonra, üçüncü evrişim katmanı 256 filtrede, dördüncü ve beşinci evrişim katmanında, özellik çıkarımı için ReLU aktivasyonu ile 512 filtre uygulanmıştır. Son olarak, bir alt örnekleme işlemi olarak özellik haritalarının düzlem içi boyutluluğunu azaltmak için Maxpooling katmanı tekrar eklenir. Bu işlemlerden sonra CNN mimarisinde skaler çıktılar elde edilir. İkinci bölümde, kapsül ağ mimarisi, CNN'nin skaler çıktıları girdi olarak kullanılmıştır. Bir kapsül ağında, skaler çıktıların yerini, kapsüllerdeki vektör çıktıları alır. Primarycaps katmanında hiperparametre değerleri filtre boyutu = 4×4 , adım = 2, digitcapsule = 8, channel = 32 olarak alınmış ve ezme (squashing) aktivasyon fonksiyonu uygulanmıştır. PrimaryCaps katmanı 32 birincil kapsülden oluşur ve bu katmandaki her kapsül 8 boyutlu vektörler içerir. Bu katmanda, kapsül çıktısını bir sonraki katmandaki tüm yüksek seviyeli kapsüllere göndermek için dinamik bir yönlendirme algoritması kullanılır ve ardından her olası yüksek seviyeli kapsül için alt kapsüller, çıktıları hesaplamak için ağırlık matrisi ile çarpılır. Daha sonra, kısa vektörlerin sifıra yakın bir uzunluğa küçültülmesi için doğrusal olmayan bir "squash" işlevi kullanılır ve çıktı değeri tahmin edilir. Sınıf kapsül katmanında, 8 boyutlu vektörler, bir ağırlık matrisi ve bir kodlama prosedürü kullanılarak kapsül başına 16 boyutlu vektörlere dönüştürülür. CCNN modelimizi, çıkarılan özellikleri nihai çıktıya eşlemek için 1024 kanaldan oluşan üç FC katmanı izler. İlk iki FC katmanında ReLU aktivasyon fonksiyonu, üçüncü FC katmanında Sigmoid aktivasyon fonksiyonu kullanılır. CCNN ağı tarafından üretilen çıkış değeri ile gerçek değer arasındaki hata oranını indirmek için Adam optimizasyon yöntemi kullanılmış ve öğrenme oranı 0.001 olarak alınmıştır.

Çizelge 4.1'de önerilen CCNN modelini ve hiperparametre değerlerini gösterilmektedir. Bu çalışmada önerilen model ve parametreler HG14 veri seti üzerinde yapılan çeşitli testlerden sonra oluşturulmuştur. Yapılan model ve parametre testlerini içeren mimariler Çizelge 4.2, Çizelge 4.3, Çizelge 4.4 ve Çizelge 4.5'te verilmiştir. Yapılan testlere ait eğitim başarımları Çizelge 4.6'da verilmiştir.

Çizelge 4.1. Önerilen CCNN modeli mimarisi.

Katman tipi	Filtre boyutu	Adım Kaydırma	Filtre sayısı Kapsül Derinliği	Dolgulama	Aktivasyon fonksiyonu
Evrişim Katmanı	4×4	1	256	same	ReLU
Evrişim Katmanı	4×4	1	512	same	ReLU
Evrişim Katmanı	2×2	1		same	ReLU
MaxPooling Katmanı	4×4	1	256	same	ReLU
Evrişim Katmanı	4×4	1	512	same	ReLU
Evrişim Katmanı	4×4	1	512	same	ReLU
Evrişim Katmanı	2×2	1		same	ReLU
MaxPooling Katmanı	4×4	2	8	valid	ReLU
Primary Kapsül Katmanı	4×4	2	16	valid	ReLU

Çizelge 4.2. CCNN V1 mimarisi.

Katman tipi	Filtre boyutu	Adım Kaydırma	Filtre sayısı Kapsül Derinliği	Dolgulama	Aktivasyon fonksiyonu
Evrişim Katmanı	3×3	1	512	same	ReLU
Evrişim Katmanı	3×3	1	512	same	ReLU
Evrişim Katmanı	2×2	1		same	ReLU
MaxPooling Katmanı	3×3	1	512	same	ReLU
Evrişim Katmanı	3×3	1	512	same	ReLU
Evrişim Katmanı	3×3	1	512	same	ReLU
Evrişim Katmanı	2×2	1		same	ReLU
MaxPooling Katmanı	3×3	2	8	valid	ReLU
Primary Kapsül Katmanı	3×3	2	16	valid	ReLU

Çizelge 4.3. CCNN V2 mimarisi.

Katman tipi	Filtre boyutu	Adım Kaydırma	Filtre sayısı Kapsül Derinliği	Dolgulama	Aktivasyon fonksiyonu
Evrişim Katmanı	3×3	1	512	same	ReLU
Evrişim Katmanı	3×3	1	512	same	ReLU
Evrişim Katmanı	3×3	1	512	same	ReLU
MaxPooling Katmanı	2×2	1		same	ReLU
Evrişim Katmanı	3×3	1	512	same	ReLU
Evrişim Katmanı	3×3	1	512	same	ReLU
Evrişim Katmanı	3×3	1	512	same	ReLU
MaxPooling Katmanı	2×2	1		same	ReLU
Primary Kapsül Katmanı	3×3	2	4	valid	ReLU
DigitCaps Katmanı	3×3	2	8	valid	ReLU

Çizelge 4.4. CCNN V3 mimarisi.

Katman tipi	Filtre boyutu	Adım Kaydırma	Filtre sayısı Kapsül Derinliği	Dolgulama	Aktivasyon fonksiyonu
Evrişim Katmanı	3×3	1	256	same	ReLU
Evrişim Katmanı	3×3	1	256	same	ReLU
Evrişim Katmanı	3×3	1	256	same	ReLU
MaxPooling Katmanı	2×2	1		same	ReLU
Evrişim Katmanı	3×3	1	512	same	ReLU
Evrişim Katmanı	3×3	1	512	same	ReLU
Evrişim Katmanı	3×3	1	512	same	ReLU
MaxPooling Katmanı	2×2	1		same	ReLU
Primary Kapsül Katmanı	3×3	2	4	valid	ReLU
DigitCaps Katmanı	3×3	2	8	valid	ReLU

Çizelge 4.5. CCNN V4 mimarisi.

Katman tipi	Filtre boyutu	Adım Kaydırma	Filtre sayısı Kapsül Derinliği	Dolgulama	Aktivasyon fonksiyonu
Evrişim Katmanı	4×4	1	256	same	ReLU
Evrişim Katmanı	4×4	1	256	same	ReLU
Evrişim Katmanı	4×4	1	256	same	ReLU
MaxPooling Katmanı	2×2	1		same	ReLU
Evrişim Katmanı	4×4	1	512	same	ReLU
Evrişim Katmanı	4×4	1	512	same	ReLU
Evrişim Katmanı	4×4	1	512	same	ReLU
MaxPooling Katmanı	2×2	1		same	ReLU
Primary Kapsül Katmanı	4×4	2	4	valid	ReLU
DigitCaps Katmanı	4×4	2	4	valid	ReLU

Çizelge 4.6. CCNN mimarileri eğitim doğruluk oranları.

Model Adı	CCNN Modeli	CCNN V1 Modeli	CCNN V2 Modeli	CCNN V3 Modeli	CCNN V4 Modeli
Eğitim Doğruluk Oranı	0.99999	0.92679	0.91893	0.89750	0.91679

En yüksek eğitim doğruluk oranına erişen model, çalışmada kullanılmak üzere seçilmiştir ve CCNN modeli olarak adlandırılmıştır.

Önerilen modelin fonksiyonlarının matematiksel ifadesi aşağıda gösterilmiştir.

Evrişim işleminin matematiksel ifadesi Denklem 4.1’ de gösterilmiştir.

$$a_j = f \left(\sum_i w_i x_i + b \right) \quad (4.1)$$

Denklem 4.1’de a_j (i+1). katmandaki çıktı sonucunu; w_i bir önceki katmandaki çekirdek (kernel) ağırlıklarını; x_i bir önceki katmanda oluşan özellik haritasını; b bir önceki

katmandaki bias değerini; f ise bir aktivasyon fonksiyonunu ifade etmektedir.

Evrişim katmanında uygulanan ReLU aktivasyon fonksiyonu ile giriş değeri sıfırın altında bir değer aldığı zaman çıkış değeri sıfır, giriş değeri sıfırın üzerinde bir değer aldığı zaman ise çıkış değeri giriş değerine eşitlenir. ReLU aktivasyon fonksiyonuna ait formül Denklem 4.2' de gösterilmiştir.

$$f(x)=\max(0,x) \quad (4.2)$$

CNN ağından çıkan sonuçlar 8x32 derinlikte öncül kapsüle alınmıştır. Her bir kapsül 4x4 filtre ve 1 adımlı 8 evrişim birimi içerir ve 32 kanaldan oluşur. CNN ağdan kapsül ağına girişine gelen skaler giriş değerlerine ilk olarak afin dönüşümü yapılmakta, daha sonra elde edilen değerlerin ağırlıkları toplanmakta ve son olarak squash(ezme) fonksiyonu ile vektörel olarak dönüşümü yapılmaktadır. Afin dönüşümüne ait formül Denklem 4.3'de, ağırlıkların toplanmasına ait formül Denklem 4.4'de verilmiştir.

$$\hat{u}_{jli} = W_{ij} u_i \quad (4.3)$$

Denklem 4.3'de $\hat{u}_{jli}$, ağırlık matrisi olan W_{ij} ile kapsül çıkışı olan u_i 'nin çarpılması ile elde edilen tahmin vektörüdür.

$$s_j = \sum_i c_{ij} \hat{u}_{jli} \quad (4.4)$$

Denklem 4.4'de s_j tahmin vektörlerinin toplam ağırlığını, c_{ij} ise yinelemeli dinamik yönlendirme işlemiyle belirlenen bağlantı katsayıları ifade etmektedir. Denklem 4.5'de c_{ij} bağlantı katsayı formülü verilmiştir.

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})} \quad (4.5)$$

Bir kapsülün çıkış vektörünün uzunluğunun, kısa olması durumunda sıfır uzunluğa ve uzun olması durumunda 1'in biraz altında bir uzunluğa küçülmesini sağlamak için doğrusal olmayan bir "ezme" fonksiyonu kullanılır. Ezme fonksiyonuna ait formül Denklem 4.6' da verilmiştir.

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2} \frac{s_j}{\|s_j\|} \quad (4.6)$$

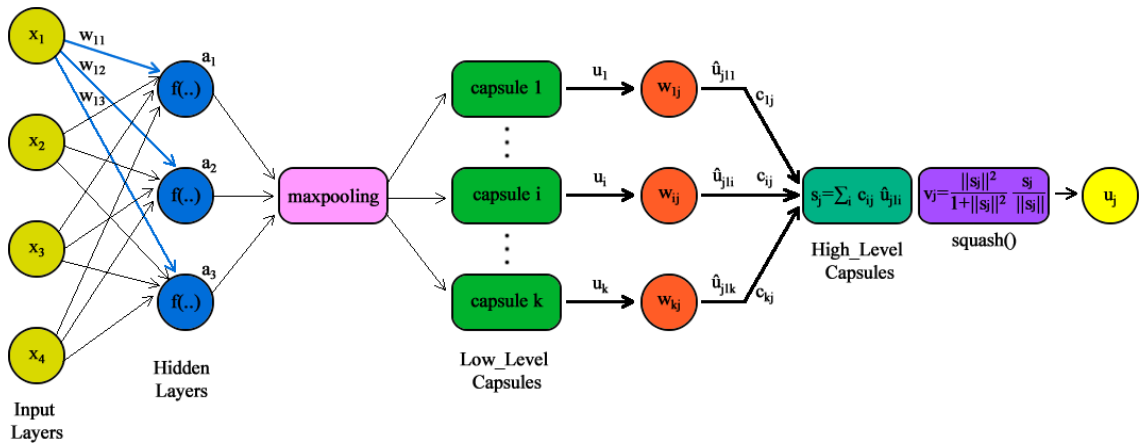
Denklem 4.6'da v_j , kapsül j'nin vektör çıkışıdır.

Çıkış katmanında (DigitsCaps) veri setinin sınıf sayısına bağlı olarak her bir sınıf için 16 kapsül vardır ve bu kapsüller bir önceki katmanlardan giriş verisi alırlar. Kapsül

ağın çıkışında 3 tam bağlı katman bulunmaktadır. Tam bağlı katmanlarda aktivasyon fonksiyonu olarak sırayla ReLU, ReLU ve sigmoid fonksiyonları uygulanmaktadır. Sigmoid fonksiyonu kullanılarak görüntünün tekrar oluşması sağlanmaktadır. Sigmoid aktivasyon fonksiyonu giriş değerlerinin her biri için sıfır ile bir arasında bir çıkış değeri üretir. Sigmoid aktivasyon fonksiyonuna ait formül Denklem 4.7' de gösterilmiştir.

$$f(x) = 1 / (1 + e^{-x}) \quad (4.7)$$

Bu çalışmada önerilen CCNN modelinin detaylı yapay sinir ağı modeli Şekil 4.2'de gösterilmiştir. Genel olarak modelimiz iki ana bölümden oluşmaktadır: CNN ve kapsül ağ katmanı. Modelimizin girdileri görüntü dizileridir. Görüntünün özellik haritasını öğrenmek için bir CNN kullanılır. CNN ile birden fazla evrişim çekirdeği kullanılarak görüntü dizisine ait özellikler oluşturulur ve ağırlıklar hesaplanır. Maksimum havuzlama ile ağırlıklar azaltılır ve çıktıların uygunluğu kontrol edilir. Daha sonra, CNN'nin skaler çıktıları, kapsül ağına girdiler olarak kullanılır ve skaler çıktılar, vektör çıktı kapsülleri ile değiştirilir. Kapsül ağı, kapsülün çıktısının uygun kapsüle ulaşmasını sağlayan güçlü bir dinamik yönlendirme mekanizması kullanır. Kapsül çıkışı, bir sonraki katmandaki tüm yüksek seviyeli kapsüllere gönderilir ve ardından her olası yüksek seviyeli kapsül için alt kapsüller, çıktıları hesaplamak için ağırlık matrisi ile çarpılır. Daha sonra, kısa vektörlerin sıfıra yakın bir uzunluğa küçültülmesi için doğrusal olmayan bir "squash" işlevi kullanılır ve çıktı değeri tahmin edilir.



Şekil 4.2. Önerilen CNN & Kapsül ağı yapay sinir ağı modeli.

4.1. VERİ SETİ

AG uygulamalarında el ile etkileşim ve uygulama kontrolü için 14 farklı el hareketi

belirlenmiştir. Her bir hareket için 17 farklı kişiden farklı renkte arka planlar kullanılarak, farklı kamera açılarından 1000 görüntü alınmıştır. %80'i eğitim görüntüsü ve %20'si test görüntüsü olarak kullanılmak üzere, toplamda 14000 renkli görüntüden oluşan 256×256 piksel boyutlarında bir El Hareketi veri seti oluşturulmuştur. 14 farklı el hareketi içerdiği için veri setimiz HandGesture14 (HG14) veri seti olarak isimlendirilmiştir. HG14 veri seti çevrimiçi bağlantısı eklerde sunulmuştur.

Veri setimiz farklı uygulamalarda kullanılmak amacıyla 32×32, 64×64, 128×128 ve 256×256 piksel boyutlarında renkli ve gri tonlamalı görüntülerden oluşmak üzere yeniden düzenlenmiştir. Veri setimizin farklı renkte arka planlar kullanılarak oluşturulmasının amacı, gerçek dünya görüntüleri ile çalışma yaparken başarısını artırmak, tek düze bir renk ile öğrenilebilirliğinin kolay olmasını engellemektir.

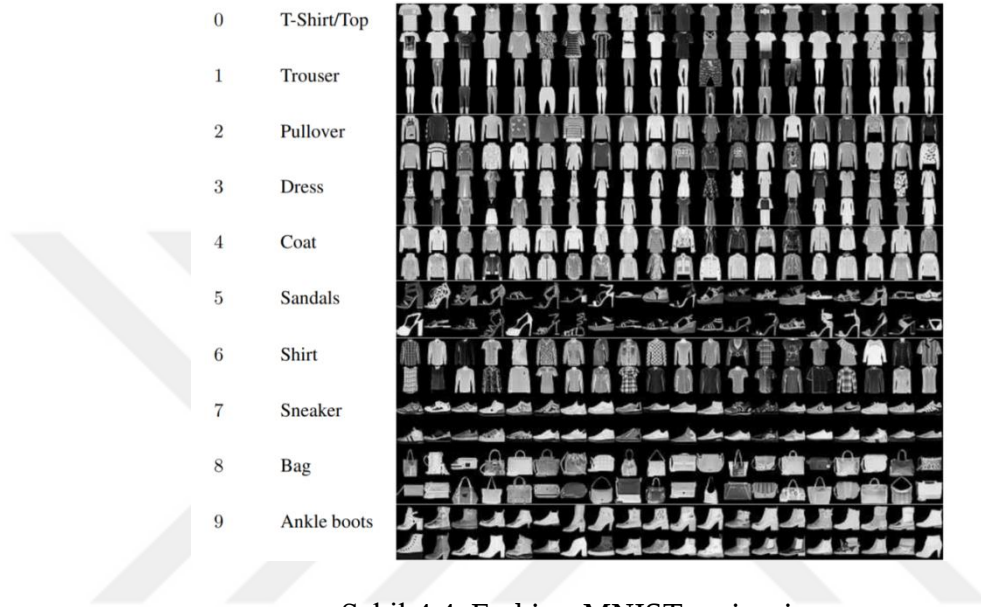


Şekil 4.3. HG14- El Hareketi Veri seti.

CNN ve kapsül ağ yapısında oluşturduğumuz hibrit modelimizin başarısını test etmek

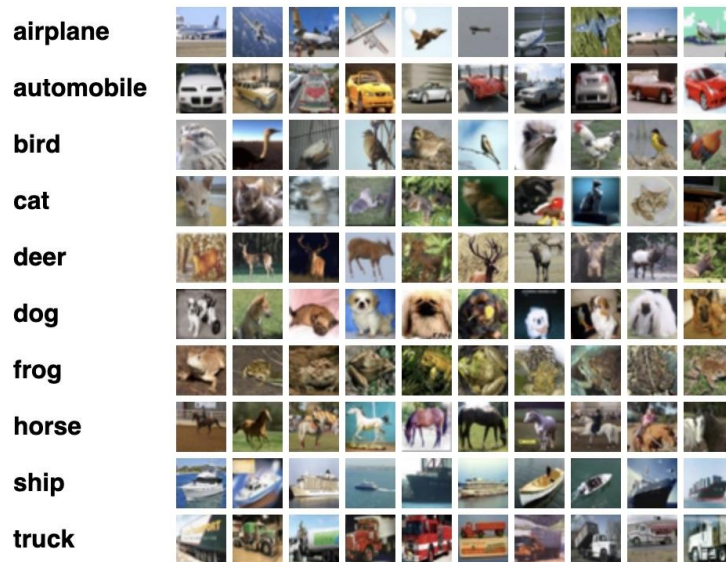
amacıyla HG14 veri seti haricinde, Zalando Arařtırma Ekibi tarafından geliřtirilen ve kıyafetlerden oluřan Fashion-MNIST [108] ve Kanada İleri Arařtırmalar Enstitüsü, Alex Krizhevky, Geoffrey Hinton, Vinod Nair tarafından dzenlenen CIFAR-10 [109] veri setleri üzerinde eęitimler yapılmıřtır.

Fashion-MNIST veri seti, 10 sınıftan, gri tonlamalı ve 28×28 pikselden oluřan 60000 eęitim grnts ve 10000 test grnts olmak zere toplam 70000 grnt zerir.



řekil 4.4. Fashion-MNIST veri seti.

CIFAR-10 veri seti, 10 sınıftan ve 3 kanalda 32×32 pikselden oluřan 50000 eęitim grnts ve 10000 test grnts olmak zere toplam 60000 grnt zerir.



řekil 4.5. CIFAR-10 veri seti.

5. TEST SONUÇLARI

Önerilen CCNN modelinin farklı veri setleri üzerindeki performansını ölçmek için, çalışma kapsamında geliştirilen HG14 veri seti haricinde, Fashion-MNIST ve CIFAR-10 veri setlerinde üzerinde de eğitimler yapılmıştır. Önerilen CCNN modelinin HG14, Fashion-MNIST ve CIFAR-10 veri setlerinde elde ettiği sonuçları karşılaştırmak üzere literatürde yer alan modellerden VGG16, DenseNet, Resnet50, MobileNet, InceptionV3 ve CapsNet modelleri ile eğitimler yapılmıştır. Eğitimlerden elde edilen sonuçlar karşılaştırılmıştır. Yapılan tüm eğitimlerde standart olması amacıyla test parametreleri çevrim (epoch) sayısı 50, batchsize 10 olarak alınmıştır. Rastgele parametreler kullanıldığı için, eğitim sonuçlarının güvenilirliğini artırmak amacıyla, her bir model ve veri seti için aynı test 30 defa tekrar edilmiş, ortalamaları alınmış ve standart sapmaları hesap edilmiştir.

Yapılan eğitimlerde Nvidia GTX 1080 TI 11 GB (Gigabyte) ekran kartı, 32 GB DDR4 (Double Data Rate 4) RAM (Random Access Memory), i5 7400 işlemci ve 240GB M.2 SSD (Solid-State Disk) donanımlarından oluşan bir bilgisayar kullanılmıştır.

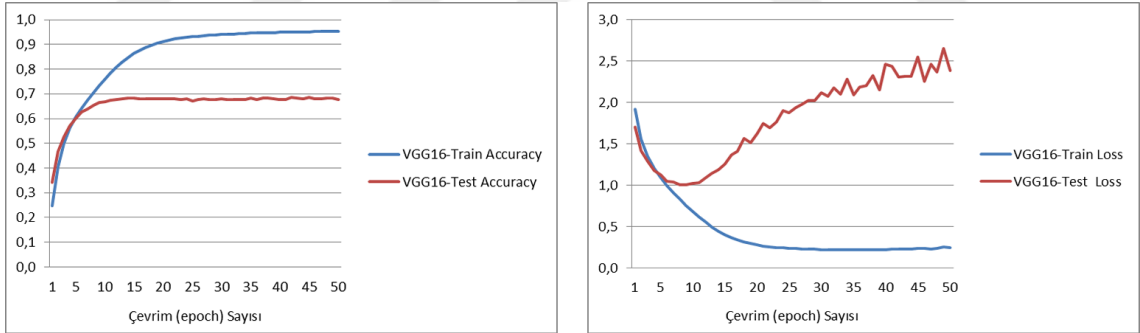
Fashion-MNIST veri seti 28×28 piksel boyutlarında olduğundan dolayı, bazı modeller üzerinde çalışma yaparken veri seti boyutu hatası vermiştir. Bu nedenle Fashion-MNIST veri seti üzerine padding işlemi uygulanarak 32×32 piksel ebatlarında yeniden düzenlenmiştir. InceptionV3 modeli minimum 75×75 piksel boyutlarında giriş verisi le çalıştığı için, InceptionV3 modeli ile yapılan eğitimlerde tüm veri setleri üzerine padding işlemi uygulanmıştır.

CIFAR-10 veri seti üzerine yapılan 30 teste ait sonuçların ortalaması, maksimum test doğruluk oranı, standart sapma oranı ve minimum hata oranları Çizelge 5.1’de verilmiştir. Çizelge 5.1 incelendiği zaman CIFAR-10 veri seti üzerinde yapılan testlerde önerilen CCNN modelin %81.42 doğruluk oranı ve %20 minimum hata oranı ile en iyi oranı elde ettiği görülmektedir.

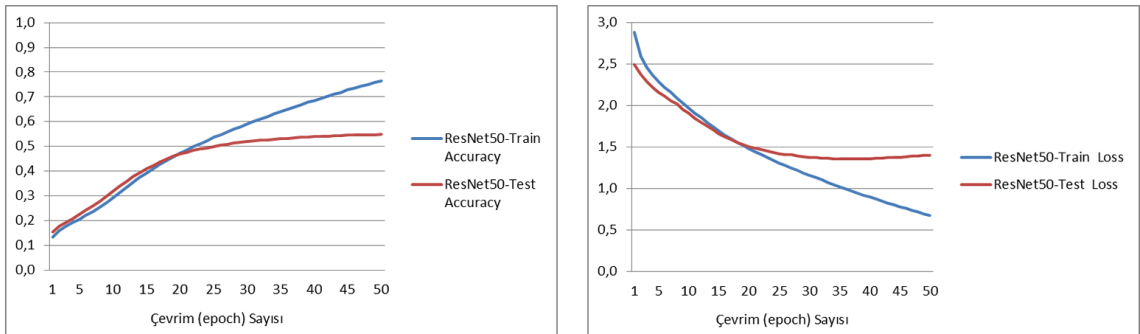
Çizelge 5.1. CIFAR-10 veri seti test sonuçları.

Model İsmi	Eğitim Doğruluk Oranı Ort.	Maksimum Geçerlilik Doğruluk Oranı	Maksimum Geçerlilik Doğruluk Oranı Ort.	Doğruluk Oranı Standart Sapma	Eğitim Hata Oranı Ort.	Geçerlilik Hata Oranı Ort.
VGG16	0.95454	0.71240	0.69941	0.00728	0.20471	0.96607
ResNet50	0.76452	0.56460	0.55138	0.00530	0.67460	1.33530
DenseNet121	0.88317	0.65600	0.64803	0.00420	0.33746	1.05596
MobileNet	0.86434	0.65090	0.63904	0.00812	0.47591	1.13976
InceptionV3	0.96983	0.75000	0.73116	0,01648	0.11558	1.07140
CapsNet	0.92538	0.66480	0.53985	0.06136	0.10036	0.34814
CCNN	0.99997	0.81420	0.75979	0.01980	0.00676	0.20740

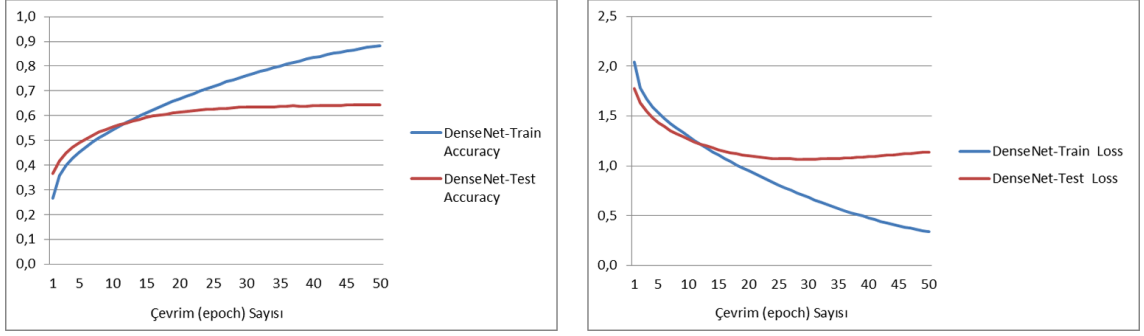
Şekil 5.1’de VGG16 modeli, Şekil 5.2’de ResNet50 modeli, Şekil 5.3’de DenseNet121 modeli, Şekil 5.4’de MobileNet modeli, Şekil 5.5’de InceptionV3 modeli, Şekil 5.6’da CapsNet modeli, Şekil 5.7’de CCNN modeli ile CIFAR-10 veri seti üzerinde yapılan testlere ait ortalama sonuçların grafikleri görülmektedir. Şekiller incelendiği zaman, testlerde önerilen CCNN modelin başarılı sonuçlar elde ettiği görülmektedir.



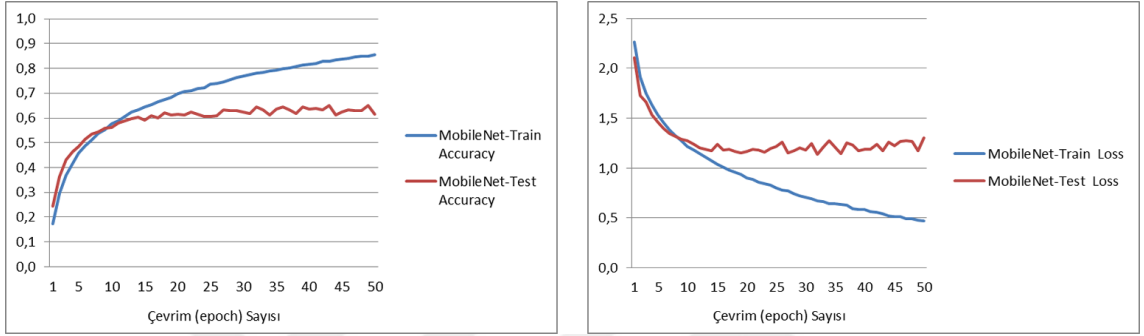
Şekil 5.1. VGG16 Modeli CIFAR-10 veri seti eğitim sonuçları.



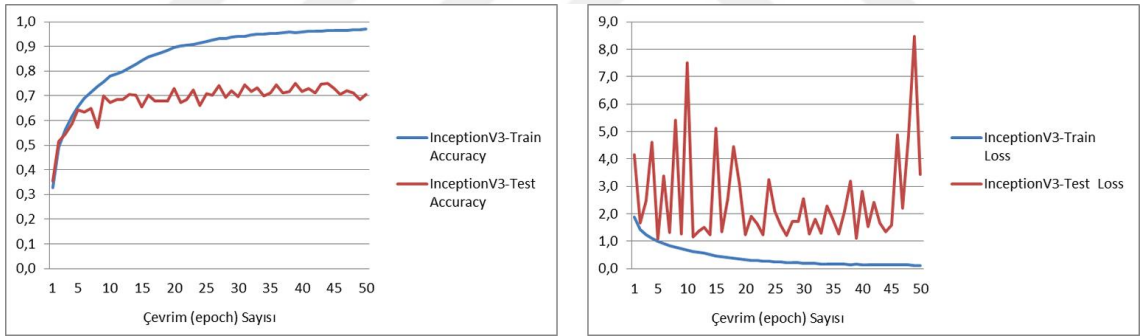
Şekil 5.2. ResNet50 Modeli CIFAR-10 veri seti eğitim sonuçları.



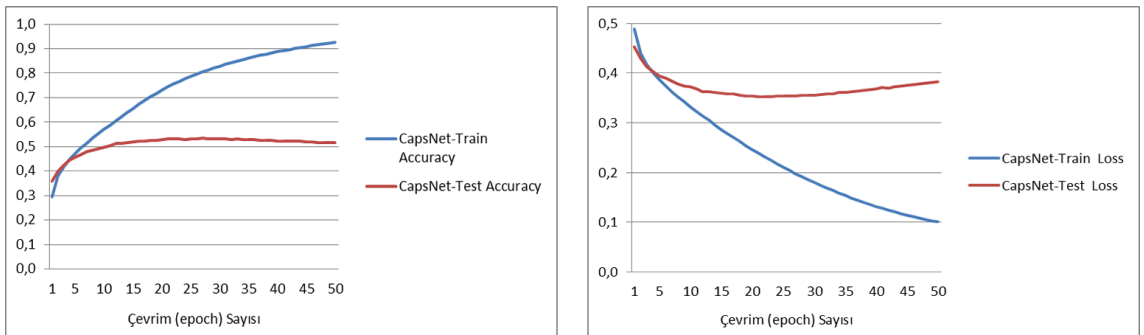
Şekil 5.3. DenseNet121 Modeli CIFAR-10 veri seti eğitim sonuçları.



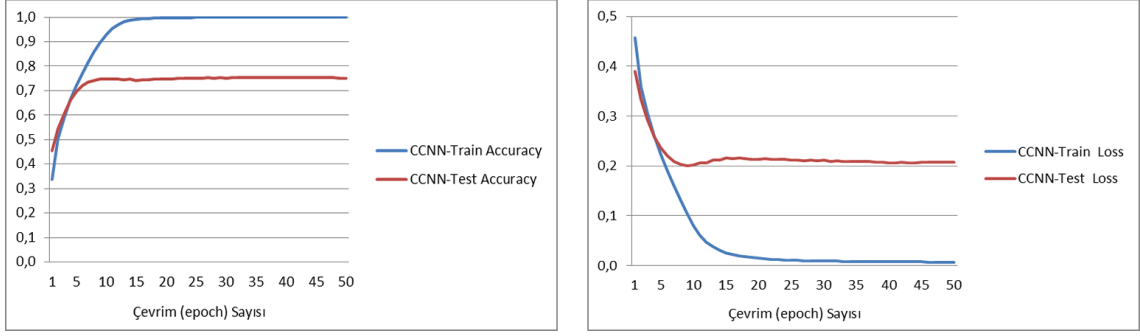
Şekil 5.4. MobilNet Modeli CIFAR-10 veri seti eğitim sonuçları.



Şekil 5.5. InceptionV3 Modeli CIFAR-10 veri seti eğitim sonuçları.



Şekil 5.6. CapsNet Modeli CIFAR-10 veri seti eğitim sonuçları.



Şekil 5.7. CCNN Modeli CIFAR-10 veri seti eğitim sonuçları.

CIFAR-10 veri seti üzerine literatürde yapılan bazı çalışmalar, metodlar ve doğruluk sonuçları Çizelge 5.2’de verilmiştir. Çizelge 5.2 incelendiği zaman önerilen hibrit CCNN modelimizin yapılan çalışmaların birçoğuna oranla geçerlilik doğruluk oranı daha yüksek sonuçlar elde ettiği görülmektedir.

Çizelge 5.2. CIFAR-10 veri seti üzerine yapılan çalışmalar.

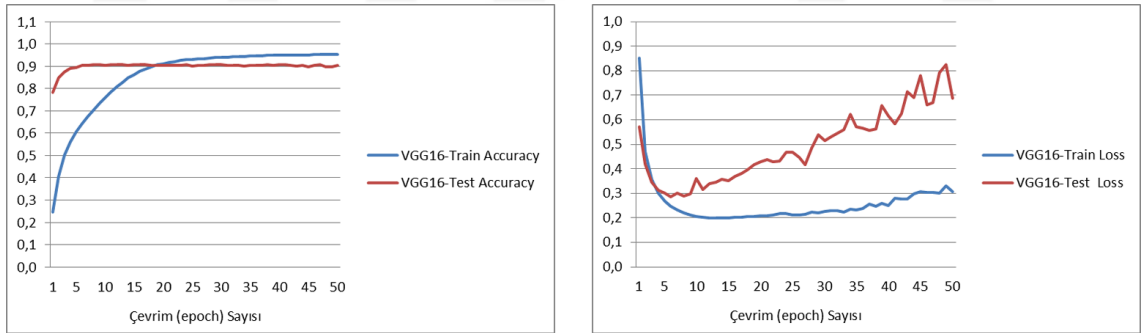
Yazar	Yöntem-Model	Doğruluk Oranı %
[110]	CNN	80.17
[111]	CNN + Gradyan Güçlendirme Makineleri	75.57
[112]	Çok Şeritli Kapsül Ağları	75.18
[113]	Kapsül Ağlar	81.20
[114]	PositNN-Posit Sayısal Format	80.40
[115]	Evrişimsel Diskriminatif Analiz	62.50
[116]	Mixed-Signal Binary CNN	88.60
[117]	Elastically Augmented Convolutions	91.11
[118]	K-Nearest Neighbors ve CNN	94.03
[119]	3B-Gömülü CNN	75.00

Fashion-MNIST veri seti üzerine yapılan 30 teste ait sonuçların ortalaması, maksimum test doğruluk oranı, standart sapma oranı ve minimum hata oranları Çizelge 5.3’ de verilmiştir. Çizelge 5.3 incelendiği zaman Fashion-MNIST veri seti üzerinde yapılan testlerde önerilen CCNN modelin %93.88 doğruluk oranı ve % 0.05 minimum hata oranı ile en iyi oranı elde ettiği görülmektedir.

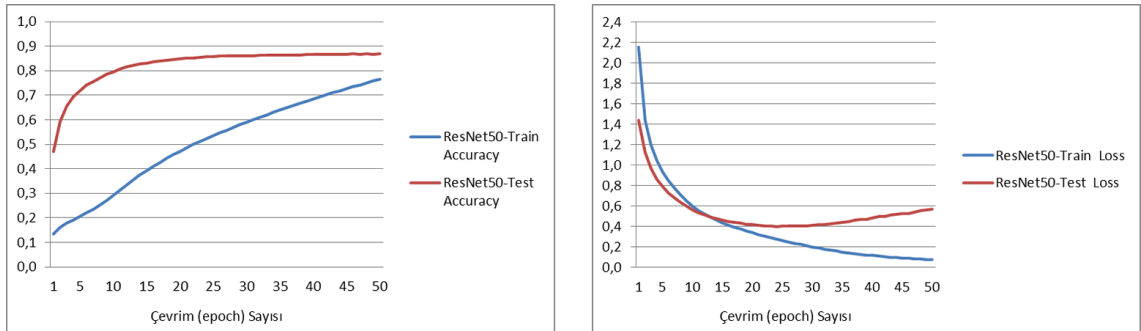
Çizelge 5.3. Fashion-MNIST veri seti test sonuçları.

Model İsmi	Eğitim Doğruluk Oranı Ort.	Maksimum Geçerlilik Doğruluk Oranı	Maksimum Geçerlilik Doğruluk Oranı Ort.	Doğruluk Oranı Standart Sapma	Eğitim Hata Oranı Ort.	Geçerlilik Hata Oranı Ort.
VGG16	0.95937	0.92140	0.91797	0.00219	0.19230	0.25976
ResNet50	0.97658	0.87510	0.87160	0.00195	0.07100	0.38906
DenseNet121	0.98752	0.89780	0.89423	0.00163	0.03645	0.31815
MobileNet	0.97263	0.90020	0.89583	0.00302	0.12181	0.44088
InceptionV3	0.98949	0.91109	0.89510	0.01850	0.04432	1.15461
CapsNet	0.99762	0.90590	0.90201	0.00202	0.00735	0.07954
CCNN	0.99998	0.93880	0.93413	0.00268	0.00280	0.05750

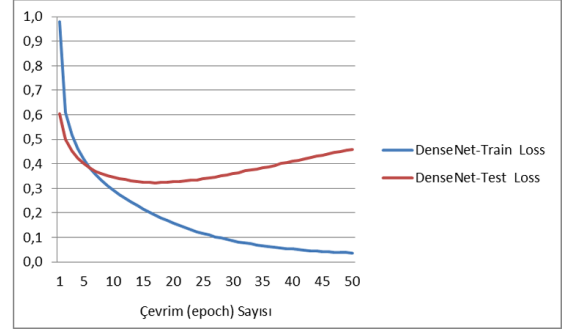
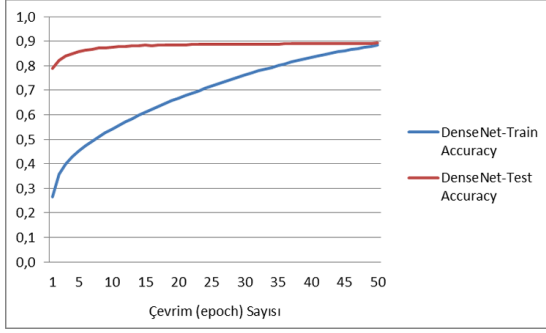
Şekil 5.8’de VGG16 modeli, Şekil 5.9’da ResNet50 modeli, Şekil 5.10’da DenseNet121 modeli, Şekil 5.11’de MobileNet modeli, Şekil 5.12’de InceptionV3 modeli, Şekil 5.13’de CapsNet modeli, Şekil 5.14’de CCNN modeli ile Fashion-MNIST veri seti üzerinde yapılan testlere ait ortalama sonuçların grafikleri görülmektedir. Şekiller incelendiği zaman, testlerde önerilen CCNN modelin başarılı sonuçlar elde ettiği görülmektedir.



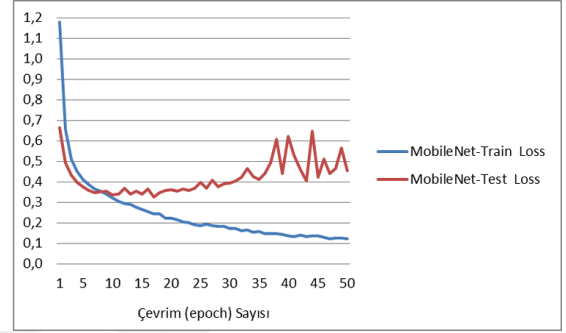
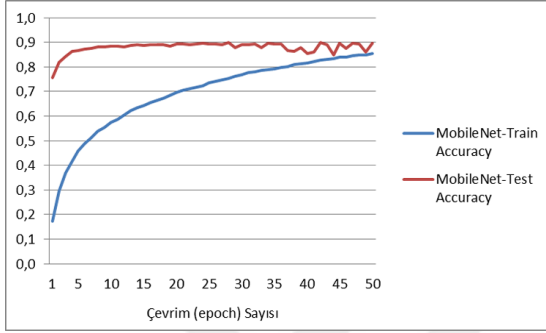
Şekil 5.8. VGG16 Modeli Fashion-MNIST veri seti eğitim sonuçları.



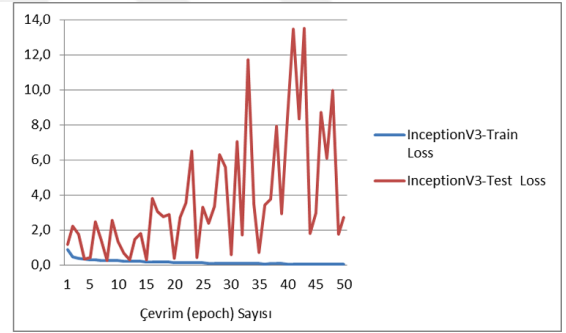
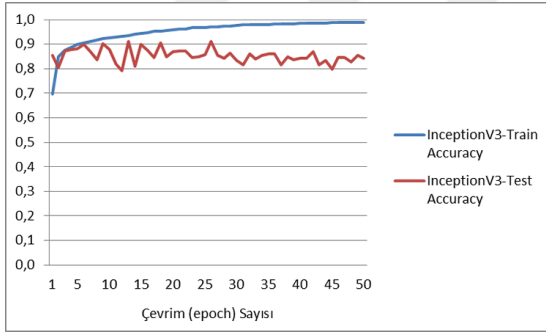
Şekil 5.9. ResNet50 Modeli Fashion-MNIST veri seti eğitim sonuçları.



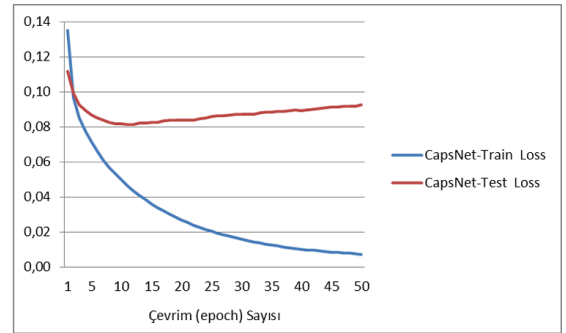
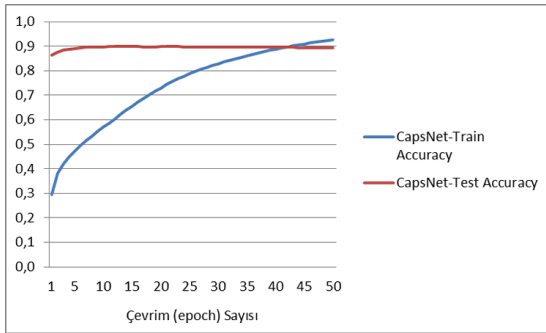
Şekil 5.10. DenseNet121 Modeli Fashion-MNIST veri seti eğitim sonuçları.



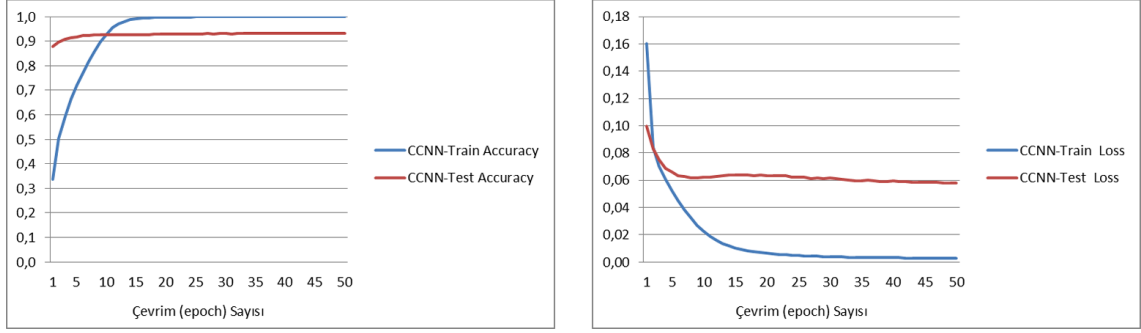
Şekil 5.11. MobileNet Modeli Fashion-MNIST veri seti eğitim sonuçları.



Şekil 5.12. InceptionV3 Modeli Fashion-MNIST veri seti eğitim sonuçları.



Şekil 5.13. CapsNet Modeli Fashion-MNIST veri seti eğitim sonuçları.



Şekil 5.14. CCNN Modeli Fashion-MNIST veri seti eğitim sonuçları.

Fashion-MNIST veri seti üzerine literatürde yapılan bazı çalışmalar, metotlar ve doğruluk sonuçları Çizelge 5.4’de verilmiştir. Çizelge 5.4 incelendiği zaman önerilen hibrit modelimizin yapılan çalışmalardan daha başarılı olduğu görülmektedir. Gresshma ve arkadaşları tarafından yapılan çalışmada %93.99 doğruluk oranı, Sun ve arkadaşları tarafından yapılan çalışmada %94.93 doğruluk oranı, Fooladgar ve arkadaşları tarafından yapılan çalışmada %99.30 doğruluk oranı elde edilmiştir. Önerilen CCNN modelimizden daha yüksek doğruluk oranları doğruluk oranı yakalayan çalışmalar olsa da önerilen modelimizin hata oranı da incelendiği zaman, modelimizin literatürde yapılan çalışmalara göre başarılı olduğu söylenebilir.

Çizelge 5.4. Fashion-MNIST veri seti üzerine yapılan çalışmalar.

Yazar	Yöntem-Model	Doğruluk Oranı %
[108]	Destek Vektör Sınıflandırıcı	89.70
[111]	CNN + Random Forest	90.84
[112]	Çok Şeritli Kapsül Ağları	92.63
[113]	Kapsül Ağları	93.89
[114]	PositNN-Posit Sayısal Format	92.70
[115]	Evrişimsel Diskriminatif Analiz	86.80
[119]	3B-Gömülü CNN	92.00
[120]	CNN	92.54
[121]	CNN	90.00
[122]	CNN+ Hiperparametre optimizasyonu+Regularizasyon	93.99
[123]	CNN	91.86
[124]	Yoğun Kapsül Ağ	94.93
[125]	Hafif Artık Yoğun Bağlı Bloklar	99.30

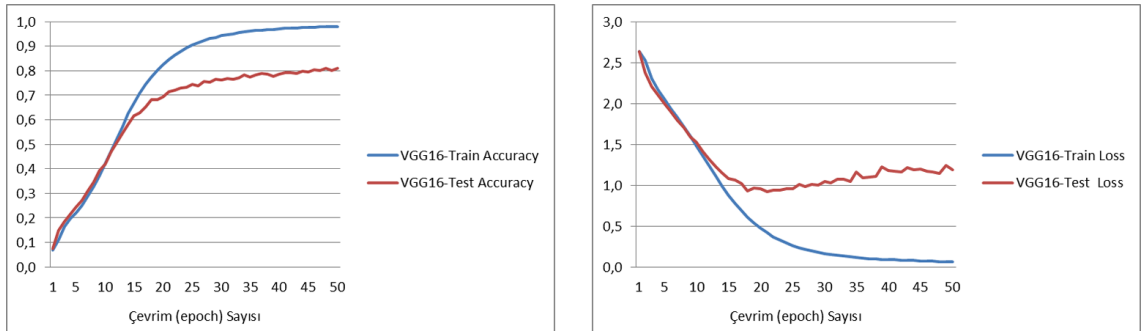
HG14 veri seti üzerine yapılan 30 teste ait sonuçların ortalaması, maksimum test

doğruluk oranı, standart sapma oranı ve minimum hata oranları Çizelge 5.5’ de verilmiştir. Çizelge 5.5 incelendiği zaman HG14 veri seti üzerinde yapılan testlerde InceptionV3 modelinin %91.46 ve %31.49 minimum hata oranı yakaladığı, önerilen CCNN modelin % 90.07 doğruluk oranı ve % 12.26 minimum hata oranı yakaladığı görülmektedir. InceptionV3 modeli, önerilen CCNN modelinden yüksek bir doğruluk oranı yakalamasına rağmen, hata ve kayıp oranları incelendiği zaman önerilen CCNN modelinin başarılı olduğu söylenebilir.

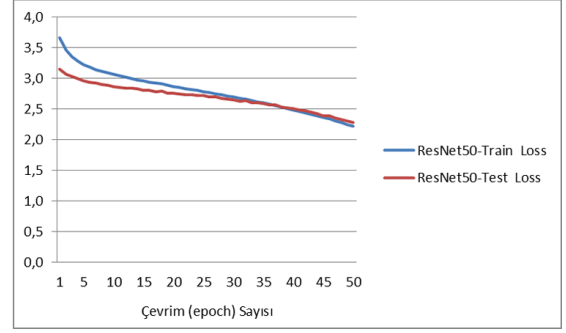
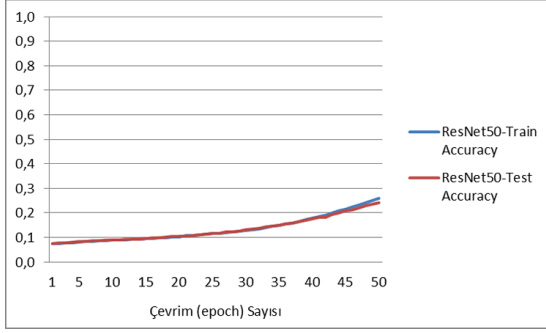
Çizelge 5.5. HG14 veri seti test sonuçları.

Model İsmi	Eğitim Doğruluk Oranı Ort.	Maksimum Geçerlilik Doğruluk Oranı	Maksimum Geçerlilik Doğruluk Oranı Ort.	Doğruluk Oranı Standart Sapma	Eğitim Hata Oranı Ort.	Geçerlilik Hata Oranı Ort.
VGG16	0.98123	0.83929	0.82276	0.01437	0.06181	0.81200
ResNet50	0.26033	0.30321	0.24313	0.03252	2.21301	2.27512
DenseNet121	0.83158	0.72857	0.70670	0.00933	0.50440	0.85346
MobileNet	0.88169	0.73071	0.65453	0.02453	0.34327	0.93719
InceptionV3	0.98696	0.91464	0.89482	0.01153	0.04776	0.31498
CapsNet	0.88847	0.64000	0.48619	0.06357	0.15966	0.41767
CCNN	0.99999	0.90071	0.87387	0.01501	0.00457	0.12269

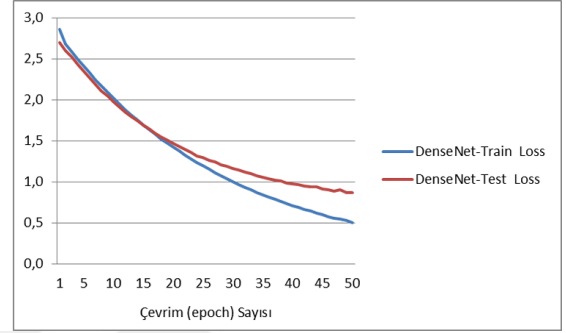
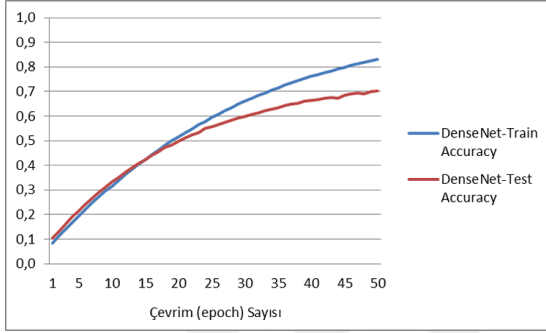
Şekil 5.15’de VGG16 modeli, Şekil 5.16’da ResNet50 modeli, Şekil 5.17’de DenseNet121 modeli, Şekil 5.18’de MobileNet modeli, Şekil 5.19’da InceptionV3 modeli, Şekil 5.20’de CapsNet modeli, Şekil 5.21’de CCNN modeli ile HG14 veri seti üzerinde yapılan testlere ait ortalama sonuçların grafikleri görülmektedir. Şekiller incelendiği zaman, testlerde önerilen CCNN modelin başarılı sonuçlar elde ettiği görülmektedir.



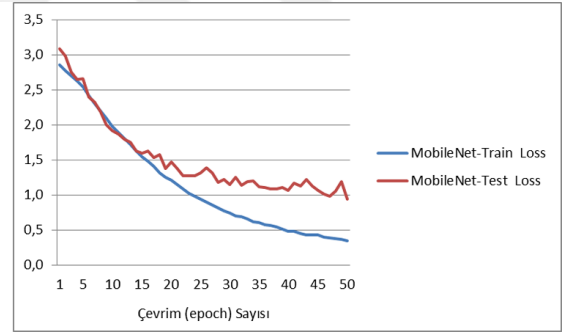
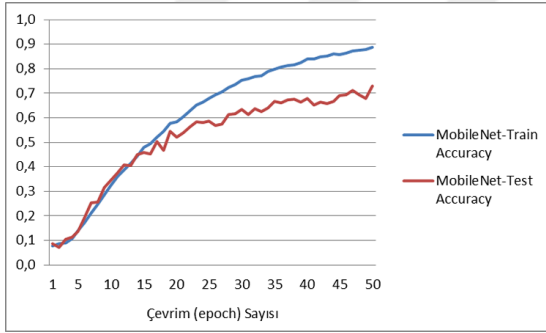
Şekil 5.15. VGG16 Modeli HG14 veri seti eğitim sonuçları.



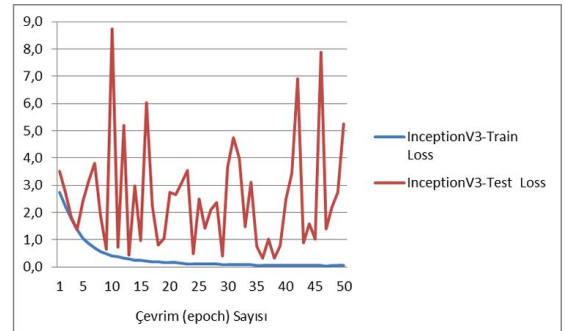
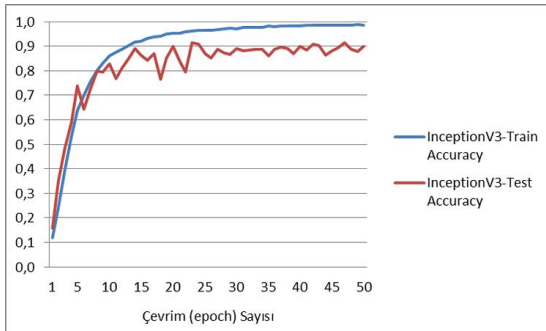
Şekil 5.16. ResNet50 Modeli HG14 veri seti eğitim sonuçları.



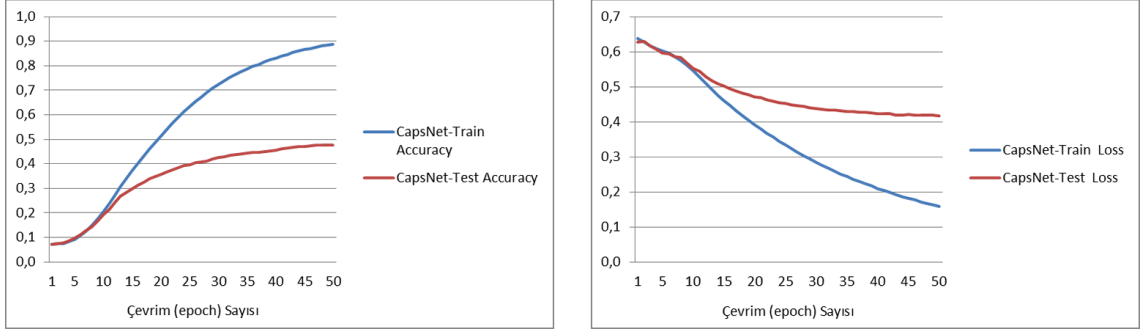
Şekil 5.17. DenseNet121 Modeli HG14 veri seti eğitim sonuçları.



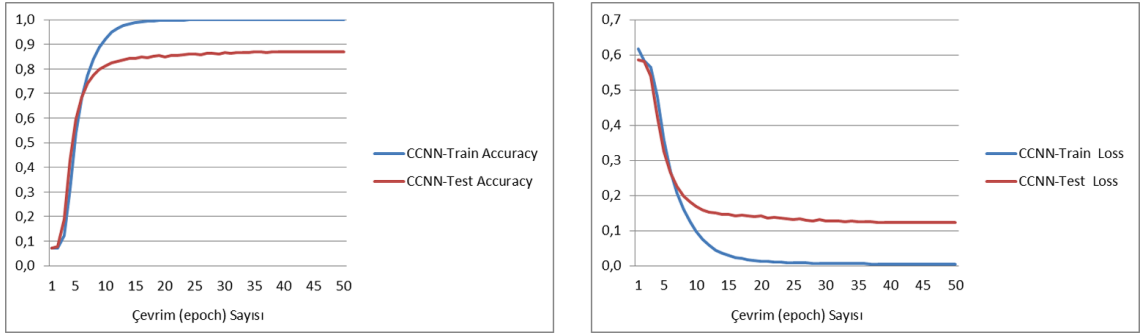
Şekil 5.18. MobileNet Modeli HG14 veri seti eğitim sonuçları.



Şekil 5.19. InceptionV3 Modeli HG14 veri seti eğitim sonuçları.



Şekil 5.20. CapsNet Modeli HG14 veri seti eğitim sonuçları.



Şekil 5.21. CCNN Modeli HG14 veri seti eğitim sonuçları.

Çizelge 5.6' da çalışmada kullandığımız VGG16, ResNet50, DenseNet121, MobileNet, , InceptionV3, CapsNet ve CCNN derin öğrenme modellerinin CIFAR-10, Fashion-MNIST ve HG14 veri setlerine yönelik eğitim sırasında 1 çevrim için geçen süreler saniye(sn) ve milisaniye(ms) olarak verilmiştir. Çizelge 5.6' da verilen eğitim süreleri incelendiğinde MobileNet modelinin 1 testi en kısa sürede gerçekleştiren model olduğu görülmektedir. MobileNet modelinden sonra VGG16 modelinin hız açısından ikinci sırada, önerilen CCNN modelimizden önce geldiği görülmektedir. Model yapıları incelendiği zaman, MobileNet modeli özellikle mobil cihazlarda çalışmak amacıyla üretildiği için, hızlı bir yapıya sahip bir modeldir. Bu nedenle en hızlı eğitim süresine erişmiştir. VGG16 modeli ise ileri beslemeli CNN yapıda olduğu için hız ve performans açısından diğer modellere kıyasla daha başarılıdır. Çalışmada önerilen CCNN modeli, hem CNN hem de kapsül yapıları içerdiği için VGG16 modelinden sonra gelmektedir. Resnet50 modeli yapısındaki kalıntı blokları, DenseNet121 modeli bir katmandan kendinden önce gelen diğer tüm katmanların çıkışlarını giriş olarak kullanmasından dolayı eğitim süresi daha uzun sürmektedir. CapsNet modeli ile önerilen CCNN modelin hız farkı ise önerilen modelde kullanılan CNN yapı ve havuzlama katmanlarında yapılan boyut azaltma işlemlerinden kaynaklanmaktadır.

Çizelge 5.6. CIFAR-10, Fashion-MNIST ve HG14 veri seti eğitim zamanı sonuçları.

Model İsmi	CIFAR-10 veri seti 50000 eğitim görüntüsü	Fashion-MNIST veri seti 60000 eğitim görüntüsü	HG14 veri seti 11200 eğitim görüntüsü
VGG16	152sn 30ms	182sn 30ms	35sn 31ms
ResNet50	330sn 66ms	393sn 66ms	75sn 67ms
DenseNet121	502sn 100ms	597sn 100ms	114sn 102ms
MobileNet	117sn 23ms	141sn 23ms	27sn 24ms
InceptionV3	347sn 69ms	418sn 68ms	79sn 70 ms
CapsNet	429sn 9ms	501sn 8ms	126sn 11ms
CCNN	327sn 7ms	384sn 6ms	87sn 8ms

Çizelge 5.7'de çalışmada kullandığımız VGG16, ResNet50, DenseNet121, MobileNet, CapsNet ve CCNN derin öğrenme modellerinin HG14 veri setine yönelik tek görüntü tahmini için geçen süreler verilmiştir.

Çizelge 5.7. Tek görüntü tahmin süreleri.

Model İsmi	VGG16	ResNet50	DenseNet121	MobileNet	InceptionV3	CapsNet	CCNN
Tahmin Süresi	0sn 57ms	1sn 30ms	2sn 11ms	0sn 85ms	1sn 67ms	1sn 75ms	1sn 23 ms

Çizelge 5.7 incelendiği tek görüntü için zaman en iyi tahmin süresi VGG16 modeli ile yakalanmıştır. Eğitim süresi MobileNet modelinde daha hızlı olmasına rağmen, tahmin süresinde ikinci sırada yer almaktadır. Önerilen CCNN modelimiz ise ortalama bir değerde tahmin süresi yakalamıştır.

HG14 veri setinin 256×256 piksel boyutlarında renkli görüntülerden oluşan versiyonu ile eğitim ve test çalışmaları yapılmıştır. Yapılan eğitimlerde 14000 görüntünün %80'i eğitim, %20'si test için ayrıştırılmıştır. Elde edilen ağırlıklar kullanılarak, HG14 veri seti görüntüleri haricinde oluşturulan, her hareket için 160 görüntü içeren toplam 2240 adet eğitime girmeyen görüntüler (unseen data) üzerinde test çalışmaları yapılmıştır. Çizelge 5.8'de çalışmada kullandığımız VGG16, ResNet50, DenseNet121, MobileNet, CapsNet ve CCNN derin öğrenme modellerinin HG14 veri setine üzerinde, eğitime girmeyen veriler (unseen data) üzerinde yapılan testlere ait doğruluk oranları verilmiştir.

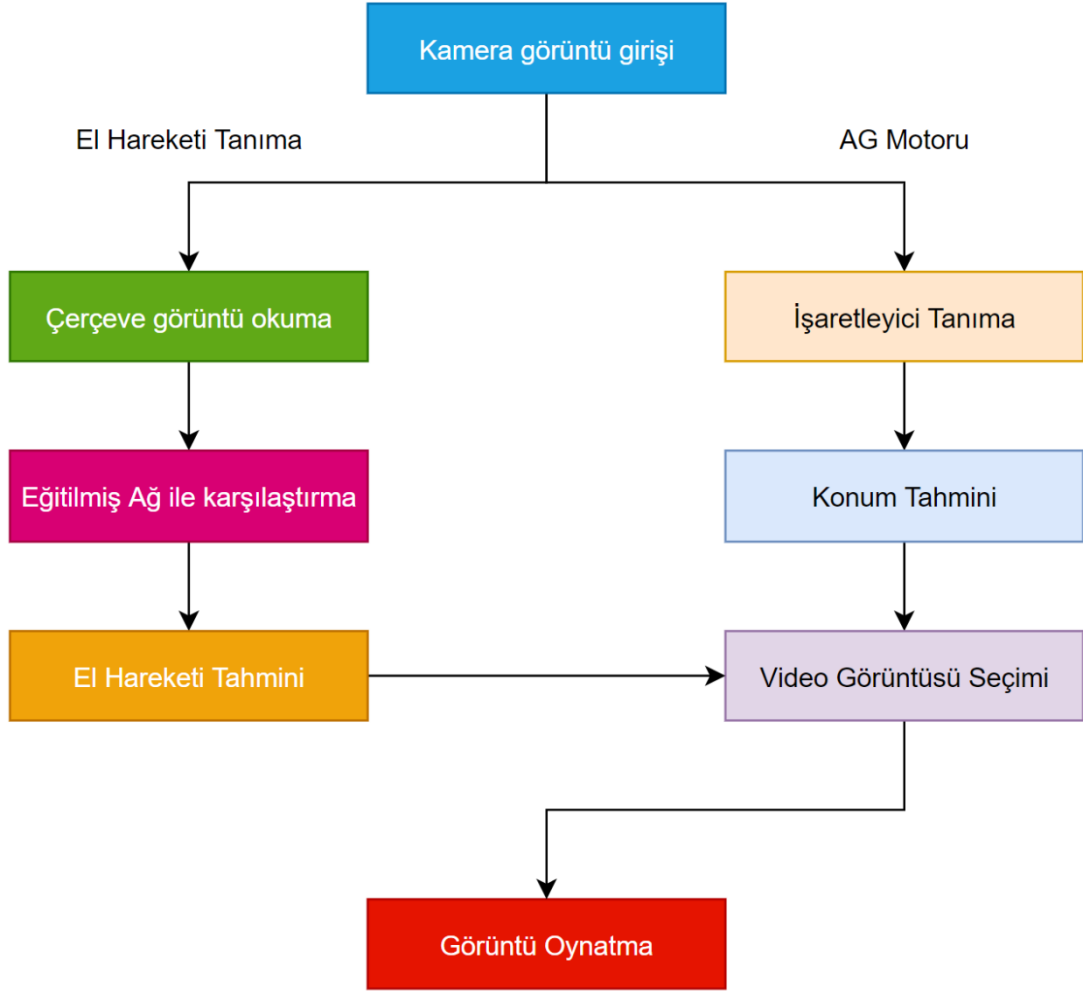
Çizelge 5.8. HG14 veri seti eğitime girmeyen veri test doğruluk oranı.

Model İsmi	Eğitime Girmeyen Veri Test Doğruluk Oranı
VGG16	0.7544
ResNet50	0.7495
DenseNet121	0.3977
MobileNet	0.7281
InceptionV3	0.2357
CapsNet	0.4681
CCNN	0.8378

6. ARTIRILMIŞ GERÇEKLİK UYGULAMASI

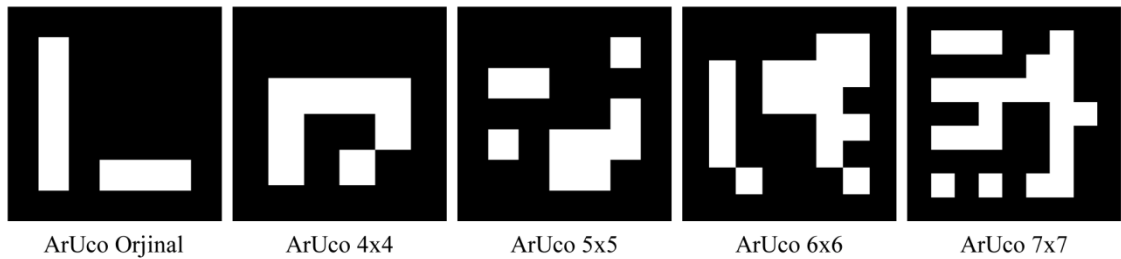
Tez çalışması kapsamında geliştirilecek olan el hareketleri ile kontrol edilebilen bir AG materyali hazırlık sürecinde öncelikle AG teknolojisini destekleyen uygun yazılım ve donanımlar araştırılmıştır. İTAG uygulamalarının kullanılabilirliğinin ve erişilebilirliğinin yüksek olması, donanım desteğinin daha çok ve maliyetinin az olmasından dolayı bu çalışmada İTAG uygulaması geliştirilmiştir. Bu tez çalışmasında, AG uygulamalarında sağlamlık, güvenilirlik ve çok yönlülük açısından tüm gereksinimlerimizi karşılayan, kamera poz tahmini için özel olarak geliştirilmiş referans işaretçi sistemi olan ArUco işaretçi sistemi kullanılmıştır [126,127]. Geliştirilen AG uygulamasının, el hareketleri ile kontrol edilebilmesi amaçlanmıştır. El ile etkileşim ve kontrol için kullanılan sistemler incelendiği zaman özel olarak tasarlanmış eldivenler, el üzerinde yerleştirilen işaretleyiciler, sensörler ve derinlik kameraları ile özel olarak tasarlanmış gözlüklerin kullanıldığı, doğal el tanıma yöntemi kullanılan AG uygulamalarında ise, elin nesne olarak tanındığı ve sanal nesnenin el üzerinde gösterildiği görülmüştür [16-56]. Bu çalışmada herhangi bir ek donanım içermeyen doğal el tanıma ile etkileşim sağlanması amaçlanmıştır. Bu kapsamda HG14 isimli bir el hareketi veri seti geliştirilmiş ve derin öğrenme modelleri ile eğitim çalışmaları yapılarak el tanıma tekniği geliştirilmiştir. Derin öğrenme çalışmalarında python programlama dili kullanıldığı için, AG uygulaması da python programlama dili ve OpenCV desteği kullanılarak geliştirilmiştir.

Şekil 6.1’de çalışma kapsamında geliştirilen AG uygulamasının çalışma akış diyagramı verilmiştir.



Şekil 6.1. AG Uygulaması Akış Diyagramı.

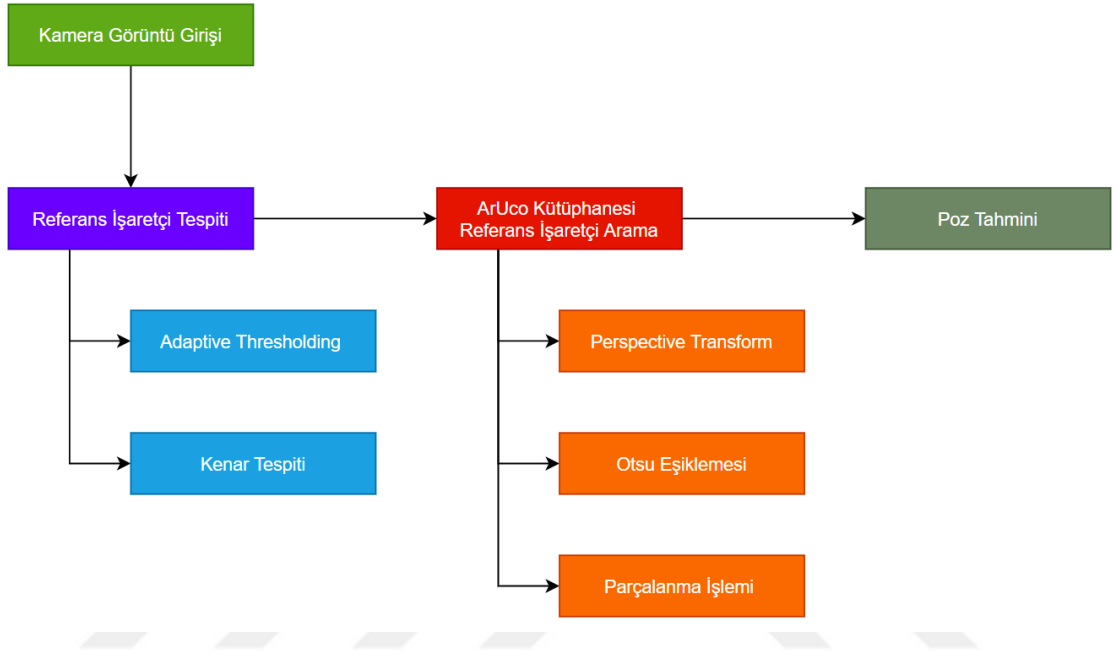
ArUco işaretleyici sistem kare koda benzer şekilde kare şeklinde ve siyah-beyaz görüntülerden oluşur. Referans işaretçiler ArUco kütüphanesinde binary(ikili) olarak 4×4 , 5×5 , 6×6 veya 7×7 boyutlarında 1 ile 1000 arasında numara verilerek saklanır. ArUco kütüphanesinde çeşitli boyutlarda yer alan 1 numaralı referans işaretleyicilere ait görüntü Şekil 6,2’de verilmiştir.



Şekil 6.2. ArUco 1 numaralı Referans İşaretçiler.

İlk olarak görüntü üzerinde referans işaretçi tespit edilir, daha sonra ArUco kütüphanesinde bulunan verilerle karşılaştırılarak referans işaretçi numarası bulunur ve işaretleyici üzerine atanan fonksiyonlar çalıştırılır.

Şekil 6.3’de ArUco referans işaretçilerinin bulunma aşamaları akış diyagramı verilmiştir.



Şekil 6.3. ArUco Referans İşaretçi Tespiti Akış Diyagramı.

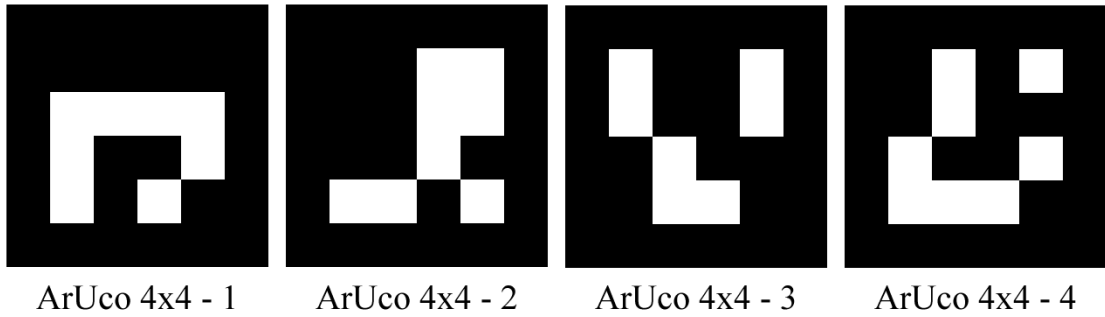
Şekil 6.3 incelendiği zaman ArUco referans işaretçilerinin bulunmalarının 3 aşamada gerçekleştiği görülmektedir.

1. Referans İşaretçi Tespiti
2. ArUco Kütüphanesi Referans İşaretçi Arama
3. Poz Tahmini

Referans işaretçi tespitinde kameradan alınan görüntüye ilk olarak adaptive thresholding eşiklemesi uygulanır ve görüntü siyah beyaz olarak dönüştürülür. ArUco referans işaretçisi kare olduğu için, siyah beyaz görüntüden kenar tespiti ile kare olan görüntüler tespit edilir. ArUco kütüphanesi referans işaretçi arama aşamasında kenar tespiti ile kare olarak kabul edilen görüntüler ilk olarak perspektif dönüşümden (perspective transform) geçer ve bu sayede kare bir görüntü olarak değerlendirilir. Daha sonra otsu eşiklemesi yapılır ve görüntü siyah ve beyaz dağılımını minimuma indirger. Görüntüyü ArUco kütüphanesinde referans işaretçilerin saklama boyutu olan 4×4, 5×5, 6×6 veya 7×7

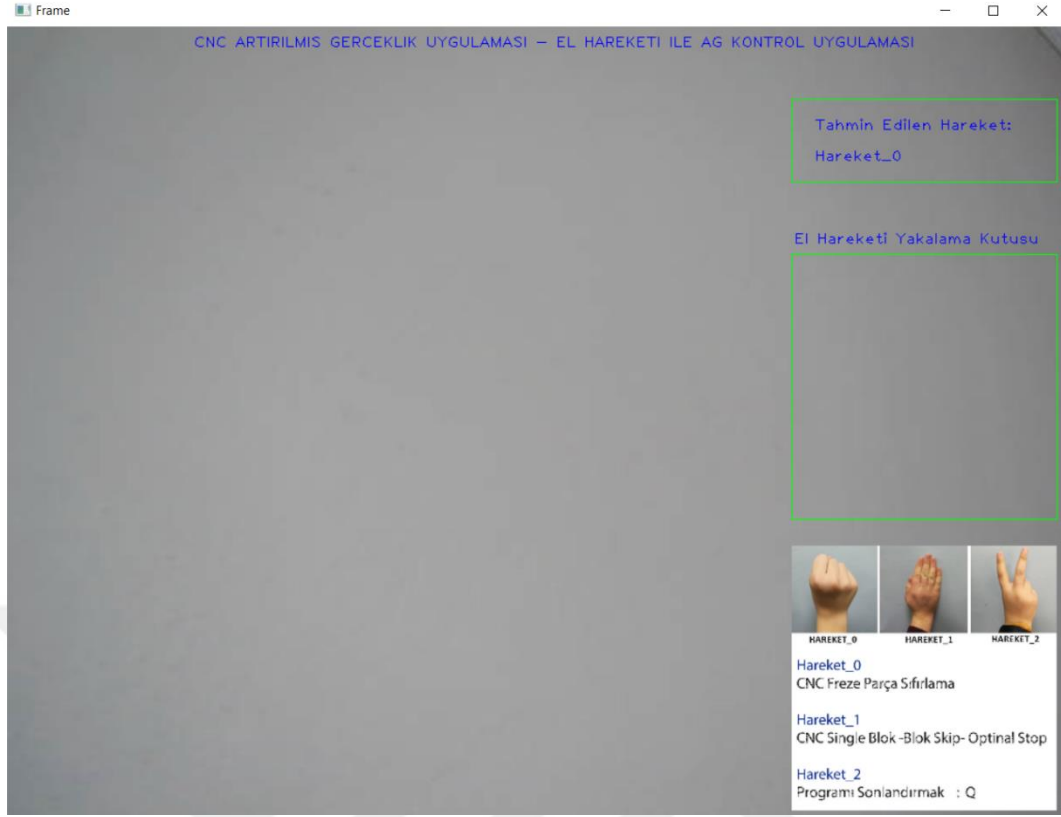
piksel boyutuna indirmek için görüntüye parçalanma işlemi uygulanır ve son olarak görüntü siyah 0, beyaz 1 olmak üzere ikili bir matris haline getirilerek ArUco kütüphanesinde aranır. ArUco kütüphanesinde eşleşen referans işaretçi değerleri fonksiyon tarafından döndürülür. Son olarak poz tahmini aşaması uygulanır. ArUco fonksiyonları, referans işaretçileri tespit ettiği zaman, kameranın pozisyonunu ve rotasyonunu gönderir. Sanal nesne bu pozisyon ve rotasyon üzerinde gösterilir.

Tez çalışması kapsamında ArUco 4x4 kütüphanesi ile 1, 2, 3 ve 4 numaralı işaretçiler oluşturulmuştur. Çalışmada 4 işaretleyici kullanılmasının nedeni, ekranda oynatılacak olan videonun sınırlarını belirlemek içindir. ArUco işaretleyicileri oluşturmak için online sistem kullanılabilir ya da OpenCV ve python programlama dili kullanılarak oluşturulabilir. Çalışmada kullanılan ArUco işaretleyici oluşturma kodu eklerde sunulmuştur. Aşağıdaki şekilde oluşturulan ArUco referans işaretleyiciler görülmektedir.



Şekil 6.4. ArUco 4x4 1-2-3-4 numaralı Referans İşaretçiler.

Tez çalışması kapsamında CNC (Computer Numerical Control) makinelerinde bakım ve ayarlama işlemlerine yönelik olarak CNC Freze Parça Sıfırlama işlemi [128] ve CNC Single Blok- Blok Skip- Optimal Stop [129] işlemlerinin videolu olarak anlatıldığı bir AG uygulaması geliştirilmiştir. Referans işaretleyici olarak Şekil 6.5’de verilen ArUco işaretleyiciler kullanılmıştır. Uygulamada 4 adet referans işaretçi kullanılarak ekranda oynatılacak olan videonun sınır köşeleri belirlenmiştir. AG uygulamasına ait ekran görüntüsü Şekil 5.8’de verilmiştir. Çalışmada geliştirilen AG uygulamasına ait kodlar eklerde sunulmuştur.

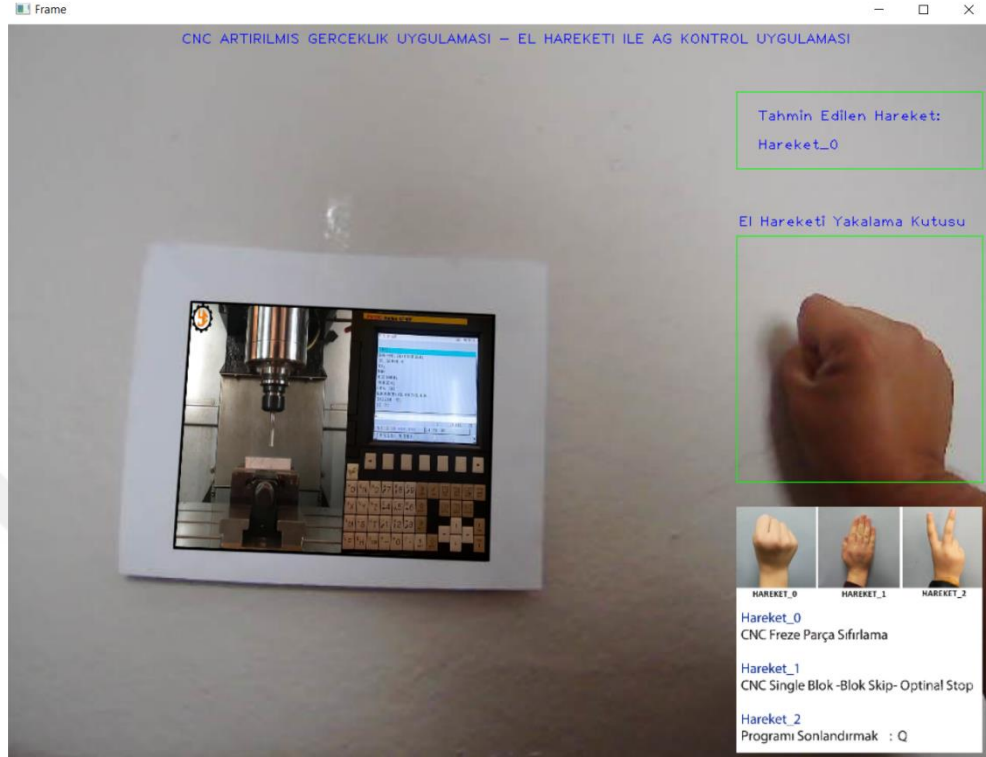


Şekil 6.5. AG uygulaması ekran görüntüsü.

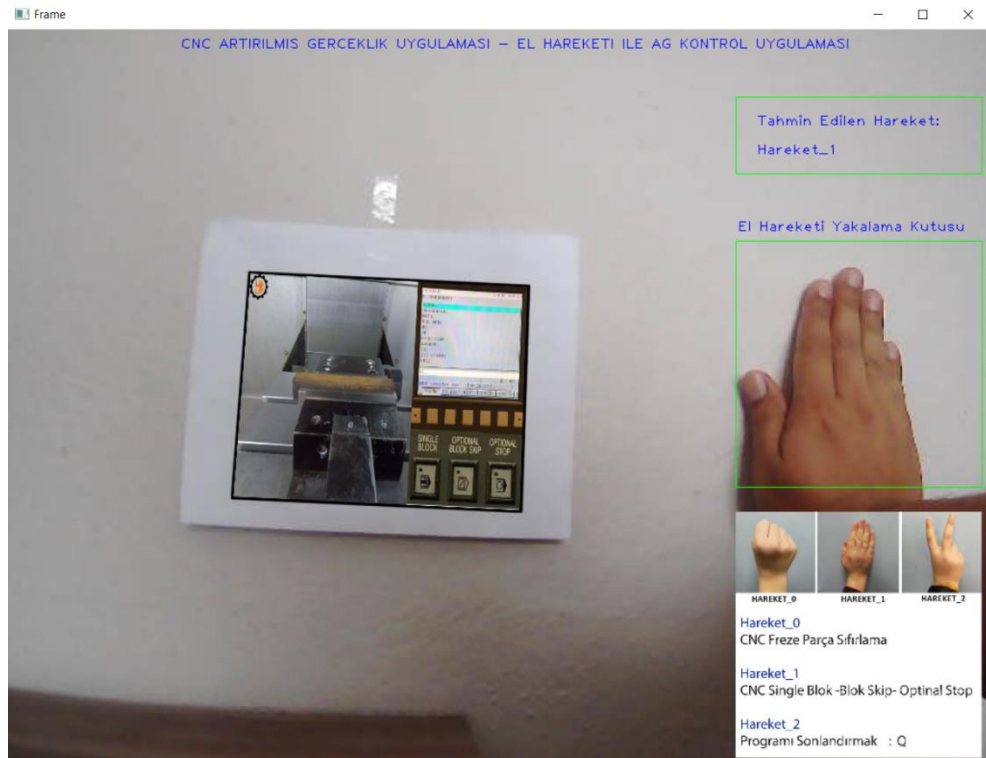
Uygulamanın sağ üst köşesinde tahmin edilen el hareketi yazdırılmaktadır. Sağ alt köşede ise uygulamada kullanılacak el hareketleri ve bu harekete bağlı olarak yapılacak işlemler görüntü formatında gösterilmiştir. Sağ orta köşede çizdirilen bir El Hareketi Yakalama Kutusu ile kutu içerisinde yapılan el hareketi tahmin edilmektedir. El Hareketi Yakalama Kutusu içerisindeki görüntü belirlenen çerçeve sınırları içerisinde yakalanarak derin öğrenme ile elde edilen ağırlıklarla karşılaştırılmakta ve tahmin edilen hareket ekrana yazdırılmaktadır. Tahmin edilen harekete göre ise ekranda video oynatılmaktadır.

Geliştirilen uygulama içerisinde, daha önceden derin öğrenme modeli eğitilen el hareketlerine ait ağırlıkların yüklenmesi ile ekranda yapılan el hareketine göre uygulama istenilen videoyu oynatılmakta, videolar arası geçiş yapılabilmektedir. AG uygulamasında 3 adet el hareketi tanımlanmış ve 2 video eklenmiştir. Birinci el hareketi ile 1 nolu video olan CNC Freze Parça Sıfırlama işlemi videosu, ikinci el hareketi ile 2 nolu video olan CNC Single Blok-Blok Skip-Optinal Stop işlemleri videosu referans işaretleyici üzerinde gösterilmektedir. Üçüncü el hareketi ile uygulama sonlandırılmaktadır. Şekil 6.6'da AG uygulaması 1. Video gösterimi, Şekil 6.7'de AG

uygulaması 2. Video gösterimi ve Şekil 6.8’de AG uygulaması program sonlandırma görüntüleri verilmiştir.



Şekil 6.6. AG uygulaması 1. Video gösterimi.



Şekil 6.7. AG uygulaması 2. Video gösterimi.



Şekil 6.8. AG uygulaması program sonlandırma.

7. SONUÇ

Gerçek dünya görüntüleri ile 3B sanal nesnelerinin birleştirilerek kullanıcıya aktarılması, AG'yi özellikle bakım, tamir, montaj vb. işlemlerde sanal nesneler ile etkileşime girilerek öğrenilmesi gereken eğitim alanlarında, ideal yöntem haline getirmektedir. Bakım, tamir, montaj vb. uygulamalı eğitimler ile kullanıcıya kazandırılmak istenen yetenek ve beceriler, bireysel öğrenim veya eğitmen kontrolü olmadan öğrenilmesi zor konulardır. Eğitimlerde parça başında bakım, tamir, montaj vb. uygulamalar yüksek maliyet ve malzeme eksikliği gibi nedenlerden dolayı her zaman uygulanmaz. Daha az maliyeti olduğu için görsel olarak hazırlanan diyagram, şema ve yönerge gibi basılı materyaller ise anlaşılması zor ve zaman alıcı oldukları için kullanışlı değildirler. Daha etkili bir eğitim yöntemi olan videolar ise kullanıcıların yapılması gereken her işlem için video ile izlemek zorunda olmaları nedeniyle fazla zaman almaktadır. Bu tarz eğitim materyalleri uygulama işlemlerinde hem süre artışına hem de hata oranı artışına neden olmakta ve fiziksel olarak zorlanmalara yol açmaktadır. AG teknolojisi gerçek görüntü ve sanal nesneleri eşzamanlı olarak dijital ortamda birleştirdiği için bu tarz problemlerin ve engellerin oranını en aza indirmektedir. AG teknolojisi kullanılarak geliştirilen eğitim materyallerinde, kullanıcılar uygulama sırasında ihtiyaç duydukları bilgiye, başını ya da vücudunu başka bir yere çevirmeden doğrudan uygulama yaptıkları alan üzerinde bir dijital ekran vasıtasıyla erişebilmektedir.

AG uygulamalarında kullanıcı ile etkileşim sağlamak için el hareketi tanıma teknolojisini kullanan çalışmalar incelendiği zaman, AG uygulamalarında özel olarak tasarlanmış eldivenler, el üzerinde yerleştirilen işaretleyiciler, sensörler ve derinlik kameraları ile özel olarak tasarlanmış gözlüklerin kullanıldığı görülmüştür. Doğal el tanıma yöntemi kullanılan AG uygulamalarında ise, elin nesne olarak tanındığı ve sanal nesnenin el üzerinde gösterildiği görülmüştür.

El hareketi tanıma, insan-bilgisayar etkileşimi alanındaki birçok farklı uygulama ile iletişimin en etkili yoludur. Derin öğrenme ile el hareketi tanıma üzerine yapılan çalışmalar incelendiğinde genellikle CNN, RNN, LSTM, 3D CNN modellerinin kullanıldığı görülmüştür [73-93]. Bu araştırmalar akıllı araç kontrolü, işaret dili tanıma

sistemleri, giyilebilir teknolojiler, robotik sistemler ve sanal gerçeklik uygulamalarına dayanmaktadır. Genel olarak incelendiğinde, el hareketi tanıma için kullanılan derin öğrenme modellerinin tüm çalışmalarda başarılı olduğu görülmüştür. Bu çalışmalardan farklı olarak çalışmamızda CNN modeli ile Capsnet modelini birleştirerek yeni bir model önerilmiş ve tasarladığımız veri seti üzerinde testler yapılmıştır. Sonuçlara göre, önerdiğimiz modelin hem farklı veri setlerinde hem de veri setimizde başarılı olduğu görülmüştür.

Bu çalışmada özellikle insan-bilgisayar etkileşimi için AG uygulamalarında kullanılmak üzere 14 farklı el hareketinden oluşan HG14 isimli bir veri seti oluşturulmuştur. Geliştirdiğimiz veri setinin diğer veri setlerinden (NVgesture, EgoGesture) [76,81] farkı, RGB, derinlik, kızılötesi gibi özel kameralarla değil, doğrudan kamera ile alınan görüntülerden oluşmasıdır. Birinci şahıs görünümünden alınan el hareketi görüntüleri, özellikle AG uygulamaları ve giyilebilir teknolojiler ile kullanılmak üzere kullanıldı. Görüntüler yalnızca el hareketlerinden oluşur ve farklı bir hareket tanımı yoktur. Birinci şahıs görüntüleme görüntüleri EgoGesture veri setine benzemekle birlikte RGB bilgisi içermez, sadece görüntü bilgisi içerir ve normal kameralı cihazlarla kullanılabilir veri setimizin farkını ortaya koymaktadır. Bu şekilde HG14 veri seti, sensör, tablet ve mobil cihazlar olmadan akıllı gözlüklerle kullanılabilir. El hareketlerini tanımak için derin öğrenme mimarileri olan CNN ve kapsül ağları kullanılarak hibrit bir model oluşturuldu. Görüntü işlemedeki yüksek başarısı nedeniyle görüntüler önce CNN ağında eğitilmekte ve elde edilen yeni değerler farklı açısal görüntülerde daha başarılı olan kapsül ağına girdi değerleri olarak verilmekte ve eğitim tamamlanmaktadır. Eğitim, önerilen CCNN modeli, VGG16, ResNet50, DenseNet121, MobileNet, InceptionV3 ve CapsNet modelleri ile HG14, Fashion-MNIST ve CIFAR-10 veri setleri üzerinde gerçekleştirildi. Eğitimin sonuçları karşılaştırıldığında, önerilen hibrit modelin HG14 veri setinde %90, Fashion-MNIST veri setinde %93.88 ve CIFAR-10 veri setinde %81.42 doğruluk oranlarına ulaştığı görülmektedir.

CCNN modelinin görüntü tanımadaki başarısını test etmek ve karşılaştırmak için HG14, Fashion-MNIST ve CIFAR-10 veri setlerinde VGG16, ResNet50, DenseNet121, MobileNet, InceptionV3 ve CapsNet modelleri ile 30 eğitim gerçekleştirildi ve ortalamaları hesaplandı. Eğitimlerin ortalama sonuçlarına göre önerilen CCNN modelimiz, CIFAR-10 veri setinde %81.42, Fashion-MNIST veri setinde %93.88 ve HG14 veri setinde %90 ile en yüksek doğruluk oranlarına ulaşmıştır. VGG16 modelinin

elde ettiđi dođruluk oranları CIFAR-10 veri setinde %71.24, Fashion-MNIST veri setinde %92.14 ve HG14 veri setinde %83.93'dür. ResNet50 modelinin elde ettiđi dođruluk oranları CIFAR-10 veri setinde %56.46, Fashion-MNIST veri setinde %87.51 ve HG14 veri setinde %30.32'dir. DenseNet121 modelinin elde ettiđi dođruluk oranları CIFAR-10 veri setinde %65.60, Fashion-MNIST veri setinde %89.78 ve HG14 veri setinde % 72.86'dır. MobileNet modelinin elde ettiđi dođruluk oranları CIFAR-10 veri setinde %65.09, Fashion-MNIST veri setinde %90.02 ve HG14 veri setinde %73.07'dir. InceptionV3 modelinin elde ettiđi dođruluk oranları CIFAR-10 veri setinde %75.00, Fashion-MNIST veri setinde %91.10 ve HG14 veri setinde %91.46'dır. CapsNet modelinin elde ettiđi dođruluk oranları CIFAR-10 veri setinde %66.48, Fashion-MNIST veri setinde %90.59 ve HG14 veri setinde %64.00'dür. Önerilen CCNN modelinin görüntü tanıma ve sınıflandırmada başarılı olduđunu söyleyebiliriz. Önerilen CCNN modeli, CNN ve Kapsül Ađı yapısı ile diđer modellerden farklılık göstermektedir. Önerilen CCNN model yapısının, CNN modelinin nesne özellikleri çıkarma ve sınıflandırma süreçlerindeki başarısı ile bir nesnenin farklı açılardan ve ışık ortamlarından alınan görüntüleri öğrenmedeki başarısını kapsül ađı ile birleřtirmesi nedeniyle başarılı olduđu söylenebilir.

Tez çalışması kapsamında python programlama dili ve OpenCV desteđi kullanılarak, CNC makinelerinde bakım ve ayarlama işlemlerine yönelik olarak videolu anlatım olan bir İTAG uygulaması geliştirilmiştir. Referans işaretleyici olarak ArUco işaretleyiciler kullanılmıştır. AG uygulamasında derin öğrenme eğitimleri sonucunda elde edilen ađrılıklar kullanılarak el hareketleri ile videolar arası geçiř sağlanmıştır.

Derin öğrenme ile yapılan çalışmalarda veri setinin boyutu, veri seti yapısı, kullanılan derin mimari yapı, optimizasyon ve hiperparametre deđerlerinin deđiřimi gibi işlemler modelin performansını etkilemektedir. Önerilen model üzerinde gerçekteřtirilecek hiperparametre düzenleme ve optimizasyon işlemleri ile modelin performansının artırılması planlanmaktadır.

Bu çalışma kapsamında geliştirilen HG14 veri seti, insan-bilgisayar etkileřimi, sanal gerçeklik ve AG uygulamaları üzerinde çalışmayı düşünen arařtırmacılar için bir kaynak teřkil edecektir.

Geliřen teknoloji ile birlikte atılan dijitalleşme adımları ile birlikte, sanayi, savunma, eğitim vb. alanlarda faaliyet gösteren kurumların ve karar verme ve iş prosedürlerini

geliştirmek, çalışan eğitimlerini vermek, pazarlama stratejilerini geliştirmek için AG teknolojisini yoğun bir şekilde kullanacakları görülmektedir. Sanal dünyada, kullanıcılar bir butona tıklayarak bilgisayar, makine vb. cihazlar ile etkileşime girecek, cihazları kontrol edebilecek, parametrelerini değiştirebilecek, veri işleyebilecek ve bakım talimatlarını gerçekleştirebileceklerdir. Tez çalışması kapsamında geliştirilen el hareketleri ile kontrol edilebilen AG sisteminin, teknoloji kullanımı olarak hayatımızın her alanında önemli bir yer bulacağı söylenebilir. Özellikle kullanıcı, çalışan ve öğrenci eğitimlerinde kullanılarak, öğrenmede kolaylık, kalıcı öğrenme, anlık eğitim ile zamandan tasarruf ve daha kalifiye çalışanların yetiştirilmesini sağlayacaktır. Önerilen AG sistemi mühendislik eğitimi, mesleki ve teknik eğitim veren kurum ve kuruluşlarda çeşitli eğitim materyalleri hazırlamak, derslerde verimliliği artırmak üzerine yapılacak olan çalışmalar için örnek teşkil edecektir. Kullanımı henüz yaygınlaşmamış olan ve içerik geliştirme açısından yeterli bilginin ve geliştiricilerin olmadığı Akıllı gözlükler için uygulama geliştirme açısından faydalı olacaktır.

8. KAYNAKLAR

- [1] T. H. H. Maung, “Real-time hand tracking and gesture recognition system using neural networks”, *Proceedings of World Academy of Science, Engineering and Technology*, c. 38, ss. 466-470, 2009.
- [2] P. Trigueiros, F. Ribeiro ve L. P. Reis, “Hand gesture recognition system based in computer vision and machine learning”, *Lecture Notes in Computational Vision and Biomechanics*, c. 19, ss. 355-377, 2015.
- [3] K. Gül ve S. Şahin, “Bilgisayar Donanım Öğretimi için Artırılmış Gerçeklik Materyalinin Geliştirilmesi ve Etkililiğinin İncelenmesi”, *Bilişim Teknojileri Dergisi*, c.10, sayı 4, ss.353-362, 2017.
- [4] C. Güngör ve M. Kurt, “Mobil Cihazlarda Görsel Arttırılmış Gerçeklik Algısının 3 Boyutlu Kırmızı-Camgöbeği Gözlükler İle Arttırılması”, *22nd IEEE Signal Processing and Communications Applications Conference (SUI 2014)*, Trabzon, Türkiye, 2014, ss. 1706-1709.
- [5] T. İçten ve G. Bal, “Artırılmış Gerçeklik Üzerine Son Gelişmelerin ve Uygulamaların İncelenmesi”, *Gazi Üniversitesi Fen Bilimleri Dergisi*, c. 5, sayı 2, ss.111-136, 2017.
- [6] D. Kaleci, T. Demirel ve İ. Akkuş, “Örnek Bir Artırılmış Gerçeklik Uygulaması Tasarımı”, *XVIII. Akademik Bilişim Konferansı*, Aydın, Türkiye, 2016, ss. 861-872.
- [7] H. Altınpulluk, “Artırılmış gerçekliği anlamak: kavramlar ve uygulamalar”, *Açıköğretim Uygulamaları ve Araştırmaları Dergisi*, c. 1, sayı 4, ss. 123-131, 2015.
- [8] K. Dorfmüller-Ulhaas ve D. Schmalstieg, “Finger tracking for interaction in augmented environments”, *Proceedings IEEE and ACM International Symposium on Augmented Reality*, New York, Amerika, 2001, ss. 55-64.
- [9] H. Koike, Y. Kobayashi ve Y. Sato, “Integrating Paper and Digital Information on EnhancedDesk: A Method for Realtime Finger Tracking on an Augmented Desk System”, *ACM Transactions on Computer-Human Interaction*, c. 8, sayı 4, ss. 307-322, 2001.
- [10] K. Oka, Y. Sato ve H. Koike, “Real-time fingertip tracking and gesture recognition”, *IEEE Computer Graphics and Applications*, c. 22, sayı 6, ss. 64-71, 2002.
- [11] B. H. Thomas ve W. Piekarski, “Glove based user interaction techniques for augmented reality in an outdoor environment”, *Virtual Reality*, c. 6, sayı 3, ss. 167-180, 2002.
- [12] C. McDonald ve G. Roth. (2003, 24 Nisan). “Replacing a mouse with hand gesture in a plane-based augmented reality system”, National Research Council of Canada, [Online]. Erişim: <http://nparc.cisti-icist.nrc-cnrc.gc.ca/npsi/ctrl?>

action=shwart&index=an&req=8913375&lang=en.

- [13] B. Limonchik ve G. Amdur. (2017, 25 Mayıs) “3D Model-Based Data Augmentation for Hand Gesture Recognition”, Stanford Universty, [Online]. Erişim: <http://cs231n.stanford.edu/reports/2017/pdfs/218.pdf>.
- [14] M. Störring, T. B. Moeslund, Y. Liu ve E. Granum, “Computer vision-based gesture recognition for an Augmented Reality interface”, *4th IASTED International Conference on Visualization, Imaging and Image Processing*, Marbella, İspanya, 2004, ss. 771.
- [15] H. Benko, E. W. Ishak ve S. Feiner, “Collaborative mixed reality visualization of an archaeological excavation”, *3rd IEEE/ACM International Symposium on Mixed and Augmented Reality*, Arlington, Amerika, 2004, ss. 132-140.
- [16] H. Benko, E. W. Ishak ve S. Feiner, “Cross-dimensional gestural interaction techniques for hybrid immersive environments”, *IEEE Proceedings, Virtual Reality*, Bonn, Almanya, 2005, ss. 209-216.
- [17] G. A. Lee, M. Billinghurst ve G. J. Kim, “Occlusion based interaction methods for tangible augmented reality environments”, *Proceedings of the 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry*, Singapur, 2004, ss. 419-426.
- [18] V. Buchmann, S. Violich, M. Billinghurst ve A. Cockburn, “FingARtips - Gesture based direct manipulation in augmented reality”, *Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*, Singapur, 2004, ss. 212-221.
- [19] W. Lee ve J. Park, “Augmented foam: A tangible augmented reality for product design”, *Proceedings of Fourth IEEE and ACM International Symposium on Symposium on Mixed and Augmented Reality*, Viyana, Avusturya, 2005, ss. 106-109.
- [20] J. H. Yoon, J. S. Park ve M. Y. Sung, “Vision-based bare-hand gesture interface for interactive augmented reality applications”, *International Conference on Entertainment Computing*, Berlin, Almanya, 2006, ss. 386-389.
- [21] T. Lee ve T. Höllerer, “Handy AR: Markerless inspection of augmented reality objects using fingertip tracking”, *11th IEEE International Symposium on Wearable Computers*, Boston, Amerika, 2007, ss. 83-90.
- [22] T. Lee ve T. Höllerer, “Initializing markerless tracking using a simple hand gesture”, *2007 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, Nara, Japonya, 2007, ss. 259-260.
- [23] T. Lee ve T. Höllerer, “Hybrid feature tracking and user interaction for markerless augmented reality”, *Virtual Reality Conference*, Nevada, Amerika, 2008, ss. 145-152.
- [24] A. Sarkar, A. Gepperth, U. Handmann ve T. Kopinski, “Dynamic hand gesture recognition for mobile systems using deep LSTM”, *International Conference on Intelligent Human Computer Interaction*, Evry, Fransa, 2017, ss. 19-31.
- [25] S. Reifinger, F. Wallhoff, M. Ablassmeier, T. Poitschke ve G. Rigoll, “Static and dynamic hand-gesture recognition for augmented reality applications”, *International Conference on Human Computer Interaction*, Beejing, Çin, 2007, ss. 728-737.

- [26] M. Lee, R. Green ve M. Billinghurst, "3D natural hand interaction for AR applications", *2008 23rd International Conference Image and Vision Computing*, Yeni Zelenda, 2008, ss. 1-6.
- [27] P. Song, H. Yu ve S. Winkler, "Vision-based 3D Finger Interactions for Mixed Reality Games with Physics Simulation", *The International Journal of Virtual Reality*, c. 8, sayı 2, ss. 1-6, 2009.
- [28] B. K. Seo, J. Choi, J. H. Han, H. Park ve J. Il Park, "One-handed interaction with augmented virtual objects on mobile devices", *The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry (VRCAI '08)*, Association for Computing Machinery, New York, Amerika, 2008, ss. 1-6.
- [29] M. Billinghurst, H. Kato ve S. Myojin, "Advanced interaction techniques for augmented reality applications", *International Conference on Virtual and Mixed Reality*, San Diego, Amerika, 2009, ss. 13-22.
- [30] J. Chun ve B. Lee, "Dynamic manipulation of a virtual object in marker-less AR system based on both human hands", *KSII Transactions on Internet & Information Systems*, c. 4, sayı 4, ss. 618-632, 2010.
- [31] B. Lee ve J. Chun, "Interactive manipulation of augmented objects in marker-less AR using vision-based hand interaction", *Seventh International Conference on Information Technology: New Generations*, Las Vegas, Amerika, 2010, ss. 398-403.
- [32] B. C. A. Fernandes ve J. F. Sánchez, "Using Haar-like feature classifiers for hand tracking in tabletop augmented reality", *Symposium on Virtual and Augmented Reality*, Natal, Brezilya, 2010, ss. 6-13.
- [33] R. Y. Wang, S. Paris ve J. Popovic, "6D hands: Markerless hand tracking for computer aided design", *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, Kaliforniya, Amerika, 2011, ss. 549-558.
- [34] J. Chun ve S. Lee, "A vision-based 3D hand interaction for marker-based AR", *International Journal of Multimedia and Ubiquitous Engineering*, c. 7, sayı 3, ss. 51-58, 2012.
- [35] C. Harshith, K.R. Shastry, M. Ravindran, M.V.V.N.S. Srikanth ve N. Lakshmikanth, "Survey on Various Gesture Recognition Techniques for Interfacing Machines Based on Ambient Intelligence", *International Journal of Computer Science & Engineering Survey*, c. 1, sayı 2, ss. 31-42, 2010.
- [36] R. Radkowski ve C. Stritzke, "Interactive hand gesture-based assembly for augmented reality applications", *Proceedings of the 2012 International Conference on Advances in Computer-Human Interactions*, Valensiya, İspanya, 2012, ss. 303-308.
- [37] H. Benko, R. Jota ve A. D. Wilson, "MirageTable: Freehand interaction on a projected augmented reality tabletop", *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Teksas, Amerika, 2012, ss.199-208.
- [38] H. Bai, L. Gao, J. El-Sana ve M. Billinghurst, "Markerless 3D gesture-based interaction for handheld Augmented Reality interfaces", *2013 IEEE International*

Symposium on Mixed and Augmented Reality (ISMAR), Adelaide, Avustralya, 2013, ss. 1-6.

- [39] W. H. Chun ve T. Höllerer, “Real-time hand interaction for augmented reality on mobile phones”, *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, New York, Amerika, 2013, ss. 307-314.
- [40] D. W. Seo ve J. Y. Lee, “Direct hand touchable interactions in augmented reality environments for natural and intuitive user experiences”, *Expert Systems with Applications*, c. 40, sayı 9, ss. 3784-3793, 2013.
- [41] W. Hürst ve C. Van Wezel, “Gesture-based interaction via finger tracking for mobile augmented reality”, *Multimedia Tools and Applications*, c. 62, sayı 1, ss. 233-258, 2013.
- [42] W. Hürst ve J. Dekker, “Tracking-based interaction for object creation in mobile augmented reality”, *Proceedings of the 21st ACM International Conference on Multimedia*, New York, Amerika, 2013, ss.93-102.
- [43] O. Akman, R. Poelman, W. Caarls ve P. Jonker, “Multi-cue hand detection and tracking for a head-mounted augmented reality system”, *Machine Vision and Applications*, c. 24, sayı 5, ss. 931-946, 2013.
- [44] T. Piumsomboon, A. Clark, M. Billinghamhurst ve A. Cockburn, “User-Defined Gestures for Augmented Reality”, *Conference on Human Factors in Computing Systems*, Berlin, Almanya, 2013, ss. 282-299.
- [45] K. Pin Ng, G. Yew Tan ve Y. Ping Wong, “Vision-Based Hand Detection for Registration of Virtual Objects in Augmented Reality”, *International Journal of Future Computer and Communication*, c. 2, sayı 5, ss. 423, 2013.
- [46] F. Kishino ve P. Milgram, “A taxonomy of mixed reality visual displays”, *IEICE Transactions on Information and Systems*, c. 77, sayı 12, ss. 1321-1329, 1994.
- [47] M. Asad ve G. Slabaugh, “Hand orientation regression using random forest for augmented reality”, *International Conference on Augmented and Virtual Reality*, Lecce, İtalya, 2014, ss. 159-174.
- [48] I. AlAgha ve R. Rasheed, “An Exploratory Study of 3D Interaction Techniques in Augmented Reality Environments”, *Journal of Software*, c. 10, sayı 4, ss. 427-440, 2015.
- [49] H. S. Adeen, A. Atia, A. Amin, A. Victor, A. Essam, E.Gharib ve M. Hussien, “RemoAct: Portable projected interface with hand gesture interaction”, *Journal of Computer Science*, c. 11, sayı 5, ss. 738, 2015.
- [50] M. Bikos, Y. Itoh, G. Klinker ve K. Moustakas, “An Interactive Augmented Reality Chess Game Using Bare-Hand Pinch Gestures”, *International Conference on Cyberworlds*, Gotland, İsveç, 2015, pp. 355-358.
- [51] R.K. Moje, C. K. Shukla, R. D. Kolhe ve S. Bhat, “Smart Device Control Using Augmented Reality”, *International Journal Of Innovative Research In Electrical, Electronics, Instrumentation And Control Engineering*, c. 3, sayı 4, 2015.
- [52] Y. S. Chang, B. Nuernberger, B. Luan ve T. Höllerer, “Evaluating gesture-based augmented reality annotation”, *2017 IEEE Symposium on 3D User Interfaces (3DUI)*, Los Angeles, Amerika, 2017, ss. 182-185.
- [53] L. Deng ve D. Yu, “Deep learning: Methods and applications”, *Foundations and*

Trends in Signal Processing, c. 7, sayı 3–4, ss. 197- 387, 2013.

- [54] O. Güler ve İ. Yücedağ, “Derin Öğrenme İle El Hareketi Tanıma Üzerine Yapılan Çalışmaların İncelenmesi”, *21. Akademik Bilişim Konferansı*, Ordu, Türkiye, 2018.
- [55] H. A. Song ve S. Y. Lee, “Hierarchical representation using NMF”, *International Conference on Neural Information Processing*, Berlin, Almanya, 2013, sayı 1, ss. 466-473.
- [56] Y. LeCun, L. Bottou, Y. Bengio ve P. Haffner, “Gradient-based learning applied to document recognition”, *Proceedings of the IEEE*, c. 86, sayı 11, ss. 2278-2324, 1998.
- [57] M. F. Akbaş ve C. Güngör, “Arttırılmış Gerçeklikte İşaretçi-Tabanlı Takip Sistemleri Üzerine Bir Literatür Çalışması Ve Tasarlanan Çok-Katmanlı İşaretçi Modeli”, *Dokuz Eylül Üniversitesi Mühendislik Fakültesi Fen ve Mühendislik Dergisi*, c. 19, sayı 56, ss. 599-619, 2017.
- [58] A. Krizhevsky, I. Sutskever ve G. E. Hinton, “ImageNet classification with deep convolutional neural networks”, *Communications of the ACM*, c. 60, sayı 6, ss. 84-90, 2017.
- [59] M. D. Zeiler ve R. Fergus, “Visualizing and understanding convolutional networks”, *European Conference on Computer Vision*, Zürich, İsviçre, 2014, ss. 818-833.
- [60] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke ve A. Rabinovich, “Going deeper with convolutions”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Boston, Amerika, 2015, ss. 1-9.
- [61] K. He, X. Zhang, S. Ren ve J. Sun, “Deep residual learning for image recognition”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Las Vegas, Amerika, 2016, ss. 770-778.
- [62] K. Simonyan ve A. Zisserman. (2015, 13 Temmuz). “Very deep convolutional networks for large-scale image recognition”, [Online]. Erişim: <https://arxiv.org/abs/1409.1556>.
- [63] G. Huang, Z. Liu, L. Van Der Maaten ve K. Q. Weinberger, “Densely connected convolutional networks”, *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, Hawaii, 2017, ss. 4700-4708.
- [64] R. T. Azuma, “A survey of augmented reality”, *Presence: Teleoperators and Virtual Environments*, c. 6, sayı 4, ss.355-385, 1997.
- [65] O. Köroğlu, “En yaygın iletişim ortamında artırılmış gerçeklik uygulamaları”, *XVII. Türkiye’de İnternet Konferansı*, Eskişehir, Türkiye, 2012.
- [66] M. Sirakaya, “Arttırılmış Gerçekliğin Uygulamalı Eğitimde Kullanımı: Anakart Montajı”, *Kırşehir Eğitim Fakültesi Dergisi*, c. 17, ss. 301–316, 2016.
- [67] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto ve H. Adam. (2017, 23 Haziran). “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”, [Online]. Erişim: <https://arxiv.org/abs/1704.04861>.

- [68] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens ve Z. Wojna, “Rethinking the Inception Architecture for Computer Vision”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, Amerika, 2016, ss. 2818-2826.
- [69] Y. Sun, B. Xue, M. Zhang, G. G. Yen ve J. Lv, “Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification”, *IEEE Transactions on Cybernetics*, c. 50, sayı 9, ss. 3840-3854, 2020.
- [70] T. Pala, I. Yücedağ, H. T. Kahraman, U. Güvenç ve Y. Sönmez, “Haar Wavelet Neural Network Model”, *2018 International Conference on Artificial Intelligence and Data Processing*, Malatya, Türkiye, 2018, ss. 1-8.
- [71] T. Pala, U. Güvenç, H. T. Kahraman, I. Yücedağ ve Y. Sonmez, “Comparison of Pooling Methods for Handwritten Digit Recognition Problem”, *2018 International Conference on Artificial Intelligence and Data Processing*, Malatya, Türkiye, 2018, ss. 1-5.
- [72] M. Korkmaz ve İ. Yücedağ, “Visual Object Detection With Deep Learning”, *1. International Technological Sciences and Design Symposium*, Giresun, Türkiye, 2018.
- [73] P. Molchanov, S. Gupta, K. Kim ve J. Kautz, “Hand gesture recognition with 3D convolutional neural networks”, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Boston, Amerika, 2015, ss. 1-7.
- [74] A. Tang, K. Lu, Y. Wang, J. Huang ve H. Li, “A real-time hand posture recognition system using deep neural networks”, *ACM Transactions on Intelligent Systems Technology*, c. 6, sayı 2, ss. 1-23, 2015.
- [75] D. Wu, L. Pigou, P.J. Kindermans, N.D.H. Le, L. Shao, J. Dambre ve J.M. Odobez, “Deep Dynamic Neural Networks for Multimodal Gesture Segmentation and Recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, c. 38, sayı 8, ss. 1583-1597, 2016.
- [76] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree ve J. Kautz, “Online Detection and Classification of Dynamic Hand Gestures with Recurrent 3D Convolutional Neural Networks”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Las Vegas, Amerika, 2016, ss. 4027-4215.
- [77] V. John, A. Boyali, S. Mita, M. Imanishi ve N. Sanma, “Deep Learning-Based Fast Hand Gesture Recognition Using Representative Frames”, *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Gold Coast, Avustralya, 2016, ss. 1-8.
- [78] O. Koller, H. Ney ve R. Bowden, “Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data is Continuous and Weakly Labelled”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Las Vegas, Amerika, 2016, ss. 3793-9802.
- [79] J. O. Pinzon Arenas, R. J. Moreno ve P. C. U. Murillo, “Hand gesture recognition by means of region-based convolutional neural networks”, *Contemporary Engineering Sciences*, c. 10, sayı 27, ss.1329-1342, 2017.
- [80] P. Bao, A. I. Maqueda, C. R. Del-Blanco ve N. Garcíá, “Tiny hand gesture

- recognition without localization via a deep convolutional network”, *IEEE Transactions on Consumer Electronics*, c. 63, sayı 3, ss. 1329-1342, 2017.
- [81] C. Cao, Y. Zhang, Y. Wu, H. Lu ve J. Cheng, “Egocentric Gesture Recognition Using Recurrent 3D Convolutional Neural Networks with Spatiotemporal Transformer Modules”, *Proceedings of the IEEE International Conference on Computer Vision*, Venice, İtalya, 2017, ss. 3763-3771.
 - [82] D. Núñez Fernández ve B. Kwolek, “Hand posture recognition using convolutional neural network”, *Iberoamerican Congress on Pattern Recognition*, Valparaiso, Şili, 2018, ss. 441-449.
 - [83] K. Roy, A. Mohanty ve R. R. Sahay, “Deep Learning Based Hand Detection in Cluttered Environment Using Skin Segmentation”, *Proceedings of 2017 IEEE International Conference on Computer Vision Workshops*, Venice, İtalya, 2017, ss. 640-649.
 - [84] A. Dadashzadeh, A. T. Targhi, M. Tahmasbi ve M. Mirmehdi, “HGR-Net: A fusion network for hand gesture segmentation and recognition”, *IET Computer Vision*, c. 13, sayı 8, ss. 700-707, 2019.
 - [85] G. Devineau, F. Moutarde, W. Xi ve J. Yang, “Deep learning for hand gesture recognition on skeletal data”, *13th IEEE Conference on Automatic Face and Gesture Recognition (FG'2018)*, Xi'An, Çin, 2018, ss. 166-113.
 - [86] E. Mathe, A. Mitsou, E. Spyrou ve P. Mylonas, “Arm Gesture Recognition using a Convolutional Neural Network”, *13th International Workshop on Semantic and Social Media Adaptation and Personalization (SMAP)*, Zaragoza, İspanya, 2018, ss. 37-42.
 - [87] E. I. Nikolaev, P. V. Dvoryaninov, Y. Y. Lensky ve N. S. Drozdovsky, “Using virtual data for training deep model for hand gesture recognition”, *Journal of Physics: Conference Series*, c. 1015, sayı 4, ss. 040245, 2018.
 - [88] U. Côté-Allard, C.L. Fall, A. Drouin, A. Campeau-Lecours, C. Gosselin, K. Glette, F. Laviolette ve B. Gosselin, “Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning”, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, c. 27, sayı 4, ss. 760-771, 2019.
 - [89] S. Skaria, A. Al-Hourani, M. Lech ve R. J. Evans, “Hand-Gesture Recognition Using Two-Antenna Doppler Radar with Deep Convolutional Neural Networks”, *IEEE Sensors Journal*, c. 19, sayı 8, ss. 3041-3048, 2019.
 - [90] C. Nuzzi, S. Pasinetti, M. Lancini, F. Docchio ve G. Sansoni, “Deep learning-based hand gesture recognition for collaborative robots”, *IEEE Instrumentation & Measurement Magazine*, c. 22, sayı 2, ss. 44-51, 2019.
 - [91] D. G. Alonso, A. Teyseyre, L. Berdun ve S. Schiaffino, “A Deep Learning Approach for Hybrid Hand Gesture Recognition”, *Mexican International Conference on Artificial Intelligence*, Xalapa, Meksika, 2019, s. 87-99.
 - [92] G. Li, H. Tang, Y. Sun, J. Kong, G. Jiang, D. Jiang, B. Tao, S. Xu ve H. Liu, “Hand gesture recognition based on convolution neural network”, *Cluster Computing*, c. 22, sayı 3, ss. 2719-2729, 2019.
 - [93] K. He, Z. Yang, L. Zhuang ve X. Zheng, “Hand gesture recognition based on convolutional neural network using a bistatic radar system”, *Eleventh*

- International Conference on Signal Processing Systems*, Chengdu, Çin, 2019, ss. 113840Q.
- [94] S. Sabour, N. Frosst ve G. E. Hinton. (2017, 18 Haziran). “Dynamic routing between capsules”, [Online]. Erişim: <http://arxiv.org/abs/1710.09829>.
- [95] M. Z. Alom, T.M. Taha, C.Yakopcic, S. Westberg, P. Sidike, M.S. Nasrin, M.Hasan, B.C. Van Essen, A.A.S. Awwal ve V.K. Asari, “A state-of-the-art survey on deep learning theory and architectures”, *Electronics*, c. 8, sayı 3, ss. 292, 2019.
- [96] J. Nagi, F. Ducatelle, G.A. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber ve L.M.Gambardella, “Max-pooling convolutional neural networks for vision-based hand gesture recognition”, *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, Kuala Lumpur, Malezya, 2011, ss. 342-347.
- [97] L. Wang ve D. Sng. (2015, 17 Haziran). “Deep Learning Algorithms with Applications to Video Analytics for A Smart City: A Survey”, [Online]. Erişim: <http://arxiv.org/abs/1512.03131>.
- [98] Ö. İnık ve E. Ülker, “Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri”, *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, c. 6, sayı 3, ss. 85-104, 2017.
- [99] F. Kinli, B. Ozcan ve F. Kirac, “Fashion image retrieval with capsule networks”, *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Seul, Güney Kore, 2019, ss. 0.
- [100] W. Rawat ve Z. Wang, “Deep convolutional neural networks for image classification: A comprehensive review”, *Neural Computation*, c. 29, sayı 9, ss. 2352-24049, 2017.
- [101] A. Adler, M. Elad, ve M. Zibulevsky. (2016, 10 Temmuz). “Compressed Learning: A Deep Neural Network Approach”, [Online]. Erişim: <http://arxiv.org/abs/1610.09615>.
- [102] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever ve R. R. Salakhutdinov. (2012, 12 Temmuz). *Improving neural networks by preventing co-adaptation of feature detectors* [Online]. Erişim: <http://arxiv.org/abs/1207.0580>.
- [103] G. Ser ve C. T. Bati, “Derin Sinir Ağları ile En İyi Modelin Belirlenmesi: Mantar Verileri Üzerine Keras Uygulaması”, *Yüziüncü Yıl Üniversitesi Tarım Bilimleri Dergisi*, c. 29, sayı 3, ss.406-417, 2019.
- [104] V. Kaya, S. Tuncer ve A. Baran, “Derin öğrenme yöntemleri kullanılarak nesne tanıma”, *International Science and Technology Conference*, Lefkoşa, Kıbrıs, 2020.
- [105] A. Kizrak. (2019, 10 Temmuz). “Comparison of Activation Functions for Deep Neural Networks”, [Online]. Erişim: <https://towardsdatascience.com/comparison-of-activation-functions-for-deep-neural-networks-706ac4284c8a>.
- [106] A. Mobiny ve H. Van Nguyen, “Fast CapsNet for lung cancer screening”, *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Granada, İspanya, 2018, ss. 741-749.
- [107] F. Doğan ve İ. Türkoğlu, “Derin Öğrenme Modelleri ve Uygulama Alanlarına

İlişkin Bir Derleme”, *Dijle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, c. 10, sayı 2, ss. 409-445, 2019.

- [108] H. Xiao, K. Rasul ve R. Vollgraf. (2017, 11 Temmuz). “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms”, [Online]. Erişim: <http://arxiv.org/abs/1708.07747>.
- [109] A. Krizhevsky, V. Nair ve G. Hinton. (2009, 11 Temmuz). “CIFAR-10 and CIFAR-100 datasets”, [Online]. Erişim: <https://www.cs.toronto.edu/~kriz/cifar.html>.
- [110] R. Chauhan, K. K. Ghanshala ve R. C. Joshi, “Convolutional Neural Network (CNN) for Image Detection and Recognition”, *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, Jalandhar, Hindistan, 2018, ss. 278-282.
- [111] A. Çayır, I. Yenidoğan ve H. Dağ, “Feature Extraction Based on Deep Learning for Some Traditional Machine Learning Methods”, *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, Saraybosna, Bosna Hersek, 2018, ss. 494-497.
- [112] V. M. Do Rosario, E. Borin ve M. Breternitz, “The Multi-Lane Capsule Network”, *IEEE Signal Processing Letters*, c. 26, sayı 7, ss. 1006-1010, 2019.
- [113] L. Luo, S. Duan ve L. Wang, “R-CapsNet: An Improvement of Capsule Network for More Complex Data”, *2019 IEEE Symposium Series on Computational Intelligence*, Xiamen, Çin, 2019, ss. 2124-2129.
- [114] H. F. Langroudi, Z. Carmichael, J. L. Gustafson ve D. Kudithipudi, “PositNN framework: Tapered precision deep learning inference for the edge”, *2019 IEEE Space Computing Conference*, Pasadena, Amerika, 2019, ss. 53-59.
- [115] G. Zhong, Y. Zheng, X. Y. Zhang, H. Wei ve X. Ling, “Convolutional Discriminant Analysis”, *Proceedings - International Conference on Pattern Recognition*, New Jersey, Amerika, 2018, ss. 1456-1461.
- [116] D. Bankman, L. Yang, B. Moons, M. Verhelst ve B. Murmann, “An Always-On 3.8 μ j/86% CIFAR-10 mixed-signal binary CNN processor with all memory on chip in 28-nm CMOS”, *IEEE Journal of Solid-State Circuits*, c. 54, sayı 1, ss. 158–172, 2019.
- [117] S. Gulshad, I. Sosnovik ve A. Smeulders. (2021, 14 Temmuz) “Built-in Elastic Transformations for Improved Robustness”, [Online]. Erişim: <http://arxiv.org/abs/2107.09391>.
- [118] Y. Abouelnaga, O. S. Ali, H. Rady, ve M. Moustafa, “CIFAR-10 : KNN-based Ensemble of Classifiers”, *2016 International Conference on Computational Science and Computational Intelligence*, Las Vegas, Amerika, 2016, ss. 1192-1195.
- [119] S. Thunder, P. Pal, Y. Wang ve P. Huang. (2021, 13 Temmuz). “Ultra Low Power 3D-Embedded Convolutional Neural Network Cube Based on α -IGZO Nanosheet and Bi-Layer Resistive Memory”, [Online]. Erişim: <https://arxiv.org/abs/2107.08178>.
- [120] S. Bhatnagar, D. Ghosal ve M. H. Kolekar, “Classification of fashion article images using convolutional neural networks”, *2017 4th International Conference on Image Information Processing*, Wagnaghat, Hindistan, 2017, ss. 1-6.

- [121] S. J. Devi, R. Priyanka, R. Soumya, H. Sowmyashree ve S. Sarveda, “Apparel Classification Using CNN”, *International Journal of Scientific Research and Reviews*, c. 7, sayı 3, ss. 2461–2464, 2019.
- [122] K. V. Greeshma ve K. Sreekumar, “Hyperparameter optimization and regularization on fashion-MNIST classification”, *International Journal of Recent Technology and Engineering*, c. 8, sayı 2, ss. 3713-3719, 2019.
- [123] A. F. Agarap. (2017, 15 Mayıs). “An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification”, [Online]. Erişim: <http://arxiv.org/abs/1712.03541>.
- [124] G. Sun, S. Ding, T. Sun, C. Zhang ve W. Du, “A novel dense capsule network based on dense capsule layers”, *Applied Intelligence*, c.1, sayı 11, 2021.
- [125] F. Fooladgar ve S. Kasaei. (2020, 12 Temmuz). “Lightweight residual densely connected convolutional neural network”, [Online]. Erişim: <https://arxiv.org/abs/2001.00526>.
- [126] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas ve M. J. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion”, *Pattern Recognition*, c. 47, sayı 6, ss. 2280-2292, 2014.
- [127] D. Avola, L. Cinque, G. L. Foresti, C. Mercuri ve D. Pannone, “A practical framework for the development of augmented reality applications by using ARUco markers”, *International Conference on Pattern Recognition Applications and Methods*, Roma, İtalya, 2016, ss. 645-654.
- [128] Y. Erdoğan. (2021, 10 Temmuz). “CNC Freeze Parça Sıfırlama”, [Online]. Erişim: <https://www.youtube.com/watch?v=blzUV0SmMgs>.
- [129] Y. Erdoğan. (2021, 10 Temmuz). “CNC Single Blok-Blok Skip-Optinal Stop”, [Online]. Erişim: <https://www.youtube.com/watch?v=dYZCwv1wRRQ>.

9. EKLER

9.1. EK 1: HG14 VERİ SETİ BAĞLANTI ADRESİ

HG14 veri setine aşağıda belirtilen bağlantı adresi üzerinden ücretsiz olarak ulaşılabilir.

<https://www.kaggle.com/dataset/6e7e59b5dc68f943ff488d6cf23d71806c6e1bae7d1495e8ca742f1e99c479f9>

9.2. EK 2: GENERATE_ARUCO.PY KODLAMA DOSYASI

```
import numpy as np
import argparse
import cv2
import sys

ap = argparse.ArgumentParser()
ap.add_argument('--output', default='./tags/4.png')
ap.add_argument('--id', type=int, default=1)
ap.add_argument("-t", "--type", type=str, default="DICT_4X4_1000", help="type of
ArUCo tag to generate")
args = vars(ap.parse_args())

# ArUco kütüphanesi tanımlama
ARUCO_DICT = {
    "DICT_4X4_50": cv2.aruco.DICT_4X4_50,
    "DICT_4X4_100": cv2.aruco.DICT_4X4_100,
    "DICT_4X4_250": cv2.aruco.DICT_4X4_250,
    "DICT_4X4_1000": cv2.aruco.DICT_4X4_1000,
    "DICT_5X5_50": cv2.aruco.DICT_5X5_50,
    "DICT_5X5_100": cv2.aruco.DICT_5X5_100,
    "DICT_5X5_250": cv2.aruco.DICT_5X5_250,
    "DICT_5X5_1000": cv2.aruco.DICT_5X5_1000,
    "DICT_6X6_50": cv2.aruco.DICT_6X6_50,
    "DICT_6X6_100": cv2.aruco.DICT_6X6_100,
    "DICT_6X6_250": cv2.aruco.DICT_6X6_250,
    "DICT_6X6_1000": cv2.aruco.DICT_6X6_1000,
    "DICT_7X7_50": cv2.aruco.DICT_7X7_50,
    "DICT_7X7_100": cv2.aruco.DICT_7X7_100,
    "DICT_7X7_250": cv2.aruco.DICT_7X7_250,
    "DICT_7X7_1000": cv2.aruco.DICT_7X7_1000,
    "DICT_ARUCO_ORIGINAL": cv2.aruco.DICT_ARUCO_ORIGINAL,
    "DICT_APRIILTAG_16h5": cv2.aruco.DICT_APRIILTAG_16h5,
    "DICT_APRIILTAG_25h9": cv2.aruco.DICT_APRIILTAG_25h9,
    "DICT_APRIILTAG_36h10": cv2.aruco.DICT_APRIILTAG_36h10,
    "DICT_APRIILTAG_36h11": cv2.aruco.DICT_APRIILTAG_36h11
}

# ArUCo etiketi doğrulama
if ARUCO_DICT.get(args["type"], None) is None:
    print("[INFO] ArUCo tag of '{}' is not supported".format(
        args["type"]))
    sys.exit(0)

# ArUCo kütüphanesi yükleme
arucoDict = cv2.aruco.Dictionary_get(ARUCO_DICT[args["type"]])

# Çıktı görüntüsündeki etiket
print("[INFO] generating ArUCo tag type '{}' with ID '{}".format(
```

```
    args["type"], args["id"]))
tag = np.zeros((300, 300, 1), dtype="uint8")
cv2.aruco.drawMarker(arucoDict, args["id"], 300, tag, 1)

cv2.imwrite(args["output"], tag)
cv2.imshow("ArUCo Tag", tag)
cv2.waitKey(0)
```

9.3. EK 3: EL TANIMA.PY KODLAMA DOSYASI

```
from ffpyplayer.player import MediaPlayer
from tensorflow.keras.models import model_from_json
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import cv2
import matplotlib.pyplot as plt
import matplotlib.animation as animation
import numpy as np
import threading
from matplotlib.pyplot import imshow
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.preprocessing.image import img_to_array
from tensorflow.keras import backend as K
from imutils.video import VideoStream
from collections import deque
import argparse
import imutils
import time
from image.augmented_reality import find_and_warp

histarray={'Hareket_0', 'Hareket_1', 'Hareket_2'}

def load_model():
    try:
        json_file = open('model_3.json', 'r')
        loaded_model_json = json_file.read()
        json_file.close()
        model = model_from_json(loaded_model_json)
        model.load_weights("weight_3.hdf5")
        print("Model yuklendi.")
        model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics =
['accuracy'])
        return model
    except:
        print("Model hatas1.")
        return None

def update(histarray2):
    global histarray
    histarray=histarray2

def realtime():
    kontrol = 1
    logo = cv2.imread('gesture2.jpg')
    size = 256
    logo = cv2.resize(logo, (size, size))
    img2gray = cv2.cvtColor(logo, cv2.COLOR_BGR2GRAY)
    ret, mask = cv2.threshold(img2gray, 1, 255, cv2.THRESH_BINARY)
```

```

ap = argparse.ArgumentParser()
ap.add_argument('--input', default='./videos/CNC1.mp4')
ap.add_argument('--input2', default='./videos/CNC2.mp4')
ap.add_argument('--cache', type=int, default=-1)
args = vars(ap.parse_args())

print("[INFO] initializing marker detector...")
arucoDict = cv2.aruco.Dictionary_get(cv2.aruco.DICT_4X4_1000)
arucoParams = cv2.aruco.DetectorParameters_create()

print("[INFO] accessing video stream...")
vf = cv2.VideoCapture(args["input"])
vf2 = cv2.VideoCapture(args["input2"])
Q = deque(maxlen=128)

(grabbed, source) = vf.read()
Q.appendleft(source)

print("[INFO] starting video stream...")
vc = VideoStream(src=0).start()

classes = ["Hareket_0", "Hareket_1", "Hareket_2"]
while len(Q) > 0:
    frame = vc.read()
    frame = imutils.resize(frame, width=1024)
    cv2.putText(frame, "CNC ARTIRILMIS GERCEKLIK UYGULAMASI - EL
HAREKETI ILE AG KONTROL UYGULAMASI", (180, 20),
cv2.FONT_HERSHEY_PLAIN, 1, 255)
    cv2.rectangle(frame, (758, 220), (1014, 476), (0, 255, 0), 1)
    frame2 = frame[220:476, 700:1014]
    frame2 = cv2.resize(frame2, (256, 256))
    frame2 = cv2.cvtColor(frame2, cv2.COLOR_RGB2GRAY)
    frame2 = frame2.reshape((1,) + frame2.shape)
    frame2 = frame2.reshape(frame2.shape + (1,))
    test_datagen = ImageDataGenerator(rescale=1. / 255)
    m = test_datagen.flow(frame2, batch_size=1)
    m.reset()
    y_pred = model.predict(m, 1)
    histarray2 = {'Hareket_0': y_pred[0][0], 'Hareket_1': y_pred[0][1], 'Hareket_2':
y_pred[0][2]}
    update(histarray2)
    text = classes[list(y_pred[0]).index(y_pred[0].max())]

    cv2.rectangle(frame, (758, 70), (1014, 150), (0, 255, 0), 1)
    cv2.putText(frame, "Tahmin Edilen Hareket:", (780, 100),
cv2.FONT_HERSHEY_PLAIN, 1, 255)
    cv2.putText(frame, text, (780, 130), cv2.FONT_HERSHEY_PLAIN, 1, 255)
    cv2.putText(frame, "El Hareketi Yakalama Kutusu", (760, 210),
cv2.FONT_HERSHEY_PLAIN, 1, 255)

```

```

if text=="Hareket_0":
    kontrol = 1
if text=="Hareket_1":
    kontrol = 0
roi = frame[-size - 10:-10, -size - 10:-10]
roi[np.where(mask)] = 0
roi += logo

warped = find_and_warp(
    frame, source,
    cornerIDs=(1, 2, 3, 4),
    arucoDict=arucoDict,
    arucoParams=arucoParams,
    useCache=args["cache"] > 0)

if warped is not None:
    frame = warped
    source = Q.popleft()

if len(Q) != Q.maxlen:
    (grabbed, nextFrame) = vf.read()
    (grabbed, nextFrame2) = vf2.read()

    if grabbed:
        if kontrol:
            Q.append(nextFrame)
        else :
            Q.append(nextFrame2)

cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF
if key == ord("q"):
    break
if key == ord("Q"):
    break
if text == "Hareket_2":
    break
cv2.destroyAllWindows("preview")
vc=None

model=load_model()
if model is not None:
    realtime()

```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Osman GÜLER

Yabancı Dili : İngilizce

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Doktora	Elektrik Elektronik ve Bilgisayar Mühendisliği	Düzce Üniversitesi	2021
Lisans	Bilgisayar Mühendisliği	Abant İzzet Baysal Üniversitesi	2021
Y. Lisans	Elektronik - Bilgisayar Eğitimi	Gazi Üniversitesi	2014
Lisans	Bilgisayar Sistemleri Öğretmenliği	Gazi Üniversitesi	2006

YAYINLAR

Uluslararası (SCI, SCIE) hakemli dergilerde yayımlanan makaleler:

1. O. Güler ve İ. Yücedağ, “Hand Gesture Recognition from 2D Images by Using Convolutional Capsule Neural Networks”, *Arabian Journal for Science and Engineering*, 2021.

Diğer Ulusal - Uluslararası hakemli dergilerde yayımlanan makaleler:

1. O. Güler ve İ. Yücedağ, “Mesleki Ortaöğretim Öğrencilerinin Alan Seçimi Problemine Bulanık Mantık Temelli Yaklaşım”, *Hacettepe Üniversitesi Eğitim Fakültesi Dergisi*, c. 32, sayı 1, 2017.
2. S. Savaş, N. Topaloğlu ve O. Güler, , “Türkiye'deki Kullanıcıların Bazı Alan Adları Üzerine Tercihlerinin Belirlenmesi: Bir Anket Uygulaması”, *Bilişim Teknolojileri Dergisi*, c. 8, sayı 2, ss. 51, 2015.
3. O. Güler ve A. Erdem, “Mesleki Eğitimde Etkileşimli 3B Eğitimin Uygulanması ve Stereoskopik 3B Teknolojisi Kullanımı”, *Bilişim Teknolojileri Dergisi*, c. 7, sayı 3, 2014.

Uluslararası bilimsel toplantılarda sunulan ve bildiri kitaplarında basılan bildiriler:

1. O. Güler ve İ. Yücedağ, “Augmented Reality in Anatomy Education and Designing an Augmented Reality Application for Learning Skeleton System”, *In 2019 International Conference on Artificial Intelligence and Applied Mathematics in Engineering (ICAIAE 2019)*, Antalya, Türkiye, 2019.
2. O. Güler ve İ. Yücedağ, “Developing an CNC lathe augmented reality application for industrial maintenance training”, *In 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, Ankara, Türkiye, 2018.

Ulusal bilimsel toplantılarda sunulan ve bildiri kitaplarında basılan bildiriler:

1. O. Güler ve İ. Yücedağ, “Derin Öğrenme İle El Hareketi Tanıma Üzerine Yapılan Çalışmaların İncelenmesi” , *21. Akademik Bilişim Konferansı Bildirileri*, Ordu, Türkiye, 2019.
2. S. Savaş, O. Güler, K. Kaya, G. Çoban ve M.S. Güzel, “Bilişim Teknolojileri Alanında Oyun ile Öğrenme Uygulaması”, *21. Akademik Bilişim Konferansı Bildirileri*, Ordu, Türkiye, 2019.
3. O. Güler ve S. Savaş, “Etkileşimli Tahtalara Yönelik Stereoskopik 3 Boyutlu Eğitim Materyali Tasarımı ve Kullanılabilirliğinin Analizi”, *FATİH Projesi Eğitim Teknolojileri Zirvesi*, Ankara, Türkiye, 2018.
4. O. Güler ve İ. Yücedağ, “Artırılmış Gerçeklik: Montaj Ve Bakım Uygulamalarında El Tanıma Teknolojisi İle Etkileşim Çalışmaları”, *20. Akademik Bilişim Konferansı Bildirileri*, Karabük, Türkiye, 2018.
5. K. Yaman, S. Savaş, O. Güler, H. Bodukcu, Ö. F. Özçelik ve H. Açıan, “Mesleki Eğitimde Stereoskopik 3B Görüntüleme Teknolojisinin Kullanımı ve Memnuniyet Ölçeği ile Değerlendirilmesi”, *17. Akademik Bilişim Konferansı Bildirileri*, Eskişehir, Türkiye, 2015.