

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

# MAKİNE ÖĐRENMESİ ALGORİTMALARIYLA TÜRK İŐARET DİLİNDE HARF VE DİNAMİK SÖZCÜK

## TANIMA

DOKTORA TEZİ  
ZEKERİYA KATILMIŐ

TEZ DANIŐMANI  
Prof. Dr. Cihan KARAKUZU

BİLECİK, 2021

10436575

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

**MAKİNE ÖĐRENMESİ ALGORİTMALARIYLA TÜRK  
İŐARET DİLİNDE HARF VE DİNAMİK SÖZCÜK  
TANIMA**

DOKTORA TEZİ  
ZEKERİYA KATILMIŐ

TEZ DANIŐMANI  
Prof. Dr. Cihan KARAKUZU

BİLECİK, 2021

10436575

## BEYAN

Türk İşaret Dili Sistemlerinde El Hareketi Tanıma adlı doktora tezi hazırlık ve yazımı sırasında bilimsel araştırma ve etik kurallarına uyduğumu, başkalarının eserlerinden yararlandığım bölümlerde bilimsel kurallara uygun olarak atıfta bulunduğumu, kullandığım verilerde herhangi bir tahrifat yapmadığımı, tezin herhangi bir kısmının Bilecik Şeyh Edebali Üniversitesi veya başka bir üniversitede başka bir tez çalışması olarak sunulmadığını, aksinin tespit edileceği muhtemel durumlarda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Bu çalışmanın, Bilimsel Araştırma Projeleri (BAP), TÜBİTAK veya benzeri kuruluşlarca desteklenmesi durumunda; projenin destekleyen kurumunun adı projenunumarası ile birlikte, ETİK KURUL onayı alınması durumunda ise ETİK KURUL tarih karar ve sayı bilgilerinin beyan edilmesi gerekmektedir.			
<b>DESTEK ALINMIŞTIR</b>	<input type="checkbox"/>	<b>DESTEK ALINMAMIŞTIR</b>	<input checked="" type="checkbox"/>
<b>Destek alındı ise;</b>			
<b>Destekleyen kurum;</b>			
<b>Desteğin Türü</b>		<b>Proje Numarası</b>	
<b>1- BAP (Bilimsel Araştırma Projesi)</b>			
<b>2- TÜBİTAK</b>			
<b>Diğer;.....</b> .....			
<b>ETİK KURUL onayı var ise;</b>			
<b>ETİK KURUL karar tarih/sayı:</b>		...../.... .....	

Zekeriya KATILMIŞ

20 / 11 / 2021

## ÖN SÖZ

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren ve benim için her zaman danışmandan çok daha fazlası olan sayın hocam **Prof. Dr. Cihan KARAKUZU**'ya ve tezin ilerlemesinde büyük emekleri olan tez jüri üyelerim; **Prof. Dr. Uğur YÜZGEÇ** ve **Prof.Dr.Cemil ÖZ**'e sonsuz teşekkürlerimi ve saygılarımı sunarım.

Üzerimdeki emeklerini hiçbir zaman ödeyemeyeceğim aileme ve manevi desteğini eksik etmeyen sevgili eşim Sultan KATILMIŞ'a tüm kalbimle teşekkür ederim.

**Zekeriya KATILMIŞ**

**20 / 11 / 2021**

## ÖZET

### MAKİNE ÖĞRENMESİ ALGORİTMALARIYLA TÜRK İŞARET DİLİNDE HARF VE DİNAMİK SÖZCÜK TANIMA

Toplumsal hayatın bir parçası haline gelen bilgisayarlar ve akıllı arabirimler, ses ve görüntü dışında, özellikle işaretlerin ve hareketlerin tanınmasında etkin bir şekilde kullanılarak hayatı kolaylaştırmaktadır. Sunulan bu çalışmada işaret dilini kullanan işitme ve konuşma engelliler ile diğer bireylerin kolay iletişime geçmesi ve işaret dilini bilmeyen bir kişinin bu dili anlayabilmesi çerçevesinde çalışılmıştır.

Günümüzde teknolojik eğilimin taşınabilir sistemlere doğru olması sebebiyle işaret tanıma çalışmalarında LMC ve Kinect cihazı kullanılmıştır. LMC cihazı bilgisayar, mobil cihazlar başta olmak üzere diğer akıllı arabirime gömülü yada doğrudan entegre edilebilir olması, el ve parmak hareketlerini anlık tarama hızı ve hassasiyetinin yüksek olması ve gelişime açık olması sebebiyle özellikle tercih edilmiştir. Kinect ise LMC'nin yetersiz kaldığı durumlarda destek olması amacıyla kullanılmıştır.

Bu çalışmada, LMC ve Kinect cihazı kullanılarak Türk işaret dili (TİD)'ne ait çift eli statik parmak alfabesi ve dinamik sözcük tanıma sistemleri üzerine çalışılmıştır. Çalışma ön işleme, izleme, görüntü karelerinin toplanması, özellik çıkarımı, özellik seçimi ve çıkarımı, boyut indirgeme, eğitim ve test olmak üzere 8 aşamadan oluşmaktadır. Çalışma dört uygulamadan oluşmaktadır. Birinci uygulama LMC kullanılarak statik parmak alfabesi tanıma üzerine gerçekleştirilmiştir. İkinci uygulamada LMC ile dinamik sözcük tanıma sistemi tasarlanmıştır. Bu uygulama için kendi arasında benzerlik ve farklılıkları dikkate alınarak hazırlanan 50 dinamik sözcük için 4 denek kullanılarak veri kümesi oluşturulmuştur. Üçüncü uygulamada Kinect cihazı kullanılarak dinamik sözcük tanıma sistemi üzerinde çalışılmıştır. Dördüncü uygulama ise LMC+Kinect cihazları kullanılarak gerçekleştirilen farklı uzunluk ve sürelerle sahip dinamik sözcük tanıma sistemidir. Bu uygulamada kullanılan histogram ve zamansal özellik çıkarımı veri kümelerinin boyutlarını eşitleyerek sınıflandırıcıya girişi kolaylaştırır ve aynı zamanda verilerin boyutunun küçültülmesini sağlar. Bu uygulamalardan elde edilen veri kümelerinden özellik seçim algoritması, özellik çıkarım yöntemleri ve PCA, LDA ve PCA+LDA boyut indirgeme yöntemleri kullanılarak yeni veri kümeleri elde edilmiştir. Bu veri kümeleri kullanılarak makine öğrenme yöntemlerinden geleneksel, sinir ağı ve ELM tabanlı sınıflandırıcılar ile işaret tanıma başarımları analiz edilmiştir. ELM mimarileri ilk kez bir işaret dili tanıma sisteminde sınıflandırıcı olarak bu çalışmada kullanılmıştır. Tanıma başarımları

açısında gürbüz ve kararlı genelleme yeteneđi sunan ELM'in 5 farklı mimarisi ve bunların kendine özgün öğrenme yöntemleri test edilerek sonuçlar karşılaştırılmıştır. Bu çalışma ile önerilen TİD tanıma sisteminin başarıml testi 10 katlamalı çapraz doğrulama yöntemi kullanılarak yapılmıştır. Elde edilen başarıml metriklerine dayalı olarak, ELM tabanlı mimari ve makine öğrenmesi yöntemlerinden ML-KELM'in tüm veri kümelerinde başarıml oranını koruduđunu ve en yüksek başarıml oranını verdiđi ve başarıml açısından kararlı bir yapıya sahip olduđu gözlemlenmiştir.

**Anahtar Kelimeler:** İşaret dili tanıma, el hareketi tanıma, dinamik sözcük hareketleri, LMC, makine öğrenimi, aşırı öğrenme makinesi (ELM).



## **ABSTRACT**

### **ALPHABET AND DYNAMIC WORD RECOGNITION IN TURKISH SIGN LANGUAGE WITH MACHINE LEARNING ALGORITHMS**

Computers and smart interfaces, which have become a part of social life, make life easier by using them effectively, especially in recognizing signs and movements, apart from sound and image. In this study, it has been studied within the framework of easy communication between hearing and speech impaired people who use sign language and other individuals and a person who does not know sign language can understand this language.

Due to the technological trend towards portable systems today, LMC and Kinect devices have been used in sign recognition studies. The LMC device has been especially preferred because it can be embedded or directly integrated into other smart interfaces, especially computers and mobile devices, its instantaneous scanning speed and sensitivity of hand and finger movements, and its openness to development. On the other hand, Kinect was used as a support in cases where the LMC was insufficient.

In this study, two-handed static finger alphabet and dynamic word recognition systems of Turkish sign language (TSL) were studied by using LMC and Kinect device. The study consists of 8 stages: preprocessing, monitoring, collection of image frames, feature extraction, feature selection and extraction, size reduction, training and testing. The study consists of four applications. The first application was carried out on static finger alphabet recognition using LMC. In the second application, a dynamic word recognition system was designed with LMC. For this application, a dataset was created by using 4 signers for 50 dynamic words prepared by considering their similarities and differences. In the third application, the dynamic word recognition system was studied using the Kinect device. The fourth application is a dynamic word recognition system with different lengths and durations using LMC+Kinect devices. The histogram and temporal feature extraction used in this application make it easier to enter the classifier by equalizing the size of the datasets, and also reduce the size of the data. From the datasets obtained from these applications, new datasets were obtained by using feature selection algorithm, feature extraction methods and PCA, LDA and PCA+LDA dimension reduction methods. Using these datasets, signal recognition performance was analyzed with traditional machine learning methods, neural network and ELM based classifiers. ELM architectures were used for the first time as a classifier in a sign language recognition system in this study. 5 different architectures of ELM, which offers robust and stable generalization

ability in terms of recognition performance, and their unique learning methods were tested and the results were compared. The performance test of the TSL recognition system proposed in this study was carried out using the 10-fold cross-validation method. Based on the performance metrics obtained, it has been observed that ML-KELM, one of the ELM-based architecture and machine learning methods, maintains the performance rate in all datasets, gives the highest performance rate and has a stable structure in terms of performance.

**Keywords:** Sign language recognition, hand gesture recognition, dynamic word movements, LMC, machine learning, Extreme Learning Machine (ELM).



# İÇİNDEKİLER

ÖN SÖZ.....	i
ÖZET.....	ii
ABSTRACT.....	iv
TABLolar LİSTESİ.....	i
ŞEKİLLER LİSTESİ.....	iii
KISALTMALAR VE SİMGELER LİSTESİ.....	v
1. GİRİŞ.....	1
2. LİTERATÜR TARAMASI.....	3
2.1. Elektronik Eldiven Tabanlı Çalışmalar.....	3
2.2. Kamera Tabanlı Çalışmalar.....	4
2.3. Elektromiyogram (EMG) Tabanlı Çalışmalar.....	7
2.4. Myo Armband Tabanlı Çalışmalar.....	7
2.5. Kinect Tabanlı Çalışmalar.....	9
2.6. LMC Tabanlı Çalışmalar.....	11
2.6.1. Statik parmak alfabesi tanıma çalışmaları.....	12
2.6.2. Dinamik sözcük tanıma çalışmaları.....	15
2.7. Türk İşaret Dili Üzerine Mevcut Çalışmalar.....	18
3. İŞARET DİLLERİ VE ÖZELLİKLERİ.....	22
3.1. İşaret Dilleri.....	22
3.2. Türk İşaret Dili (TİD).....	23
3.3. İşaret Dili Tanıma Sistemleri.....	26
4. KULLANILAN CİHAZLAR.....	28
4.1. Leap Motion Controller (LMC) Cihazı.....	28
4.1. Kinect Cihazı.....	33
5. ÖZELLİK ÇIKARIMLARI VE ÖZELLİK SEÇİMİ.....	39
5.1. Veri Toplama ve Ön İşlem Aşaması.....	39
5.2. LMC İçin Özellik Çıkarımı.....	40
5.1.1. Birinci özellik çıkarımı.....	41
5.1.2. İkinci özellik çıkarımı.....	43

5.3. Kinect Cihazından Özellik Çıkarımı .....	45
5.4. Özellik Seçimi Algoritması .....	49
5.5. Histogram Tabanlı Özellik Çıkarımı.....	53
5.6. Zaman Tabanlı Özellik Çıkarımı.....	54
5.6.1. Standart sapma (STD).....	54
5.6.2. Medyan (M).....	54
5.6.3. Varyans (V) .....	55
5.6.4. Karesel ortalamaların karekökü (RMS).....	55
5.6.5. Ortalama (AVG) .....	55
6. ALFABE VE SÖZCÜK TANIMA UYGULAMALARI.....	56
6.1. LMC Tabanlı Parmak Alfabeti Tanıma Uygulaması .....	56
6.2. LMC Tabanlı Dinamik Sözcük Tanıma Uygulaması.....	58
6.3. Kinect Tabanlı Dinamik Sözcük Tanıma Uygulaması.....	62
6.4. LMC + Kinect Tabanlı Dinamik Sözcük Tanıma Uygulaması .....	65
7. SINIFLANDIRICILAR VE BOYUT İNDİRGE ME YÖNTEMLERİ .....	72
7.1. Geleneksel Sınıflandırıcılar .....	72
7.2. Sinir Ağı Tabanlı Sınıflandırıcılar .....	74
7.3. Aşırı Öğrenme Makinesi (ELM) Tabanlı Sınıflandırıcılar .....	75
7.4. Boyut İndirgeme Yöntemleri.....	78
7.4.1. Temel Bileşenler Analizi (PCA).....	78
7.4.2. Doğrusal Ayırım Analizi (LDA) .....	79
8. DENEYSEL SONUÇLAR.....	80
8.1. LMC Tabanlı Parmak Alfabeti Tanıma Uygulaması .....	80
8.2. LMC Tabanlı Dinamik Sözcük Tanıma Uygulaması.....	82
8.3. Kinect Tabanlı Dinamik Sözcük Tanıma Uygulaması.....	86
8.4. LMC + Kinect Tabanlı Dinamik Sözcük Tanıma Uygulaması .....	88
9. SONUÇ VE ÖNERİLER.....	93
KAYNAKÇA .....	96

## TABLULAR LİSTESİ

<b>Tablo 2.1.</b> Statik işaret tanıma literatür özeti.....	17
<b>Tablo 2.2.</b> Dinamik işaret tanıma literatür özeti .....	18
<b>Tablo 3.1.</b> LMC cihazının ölçüleri ve birimleri .....	29
<b>Tablo 3.2.</b> Yerel arabirimi oluşturan sınıflar ve açıklamaları (LMC) .....	31
<b>Tablo 3.3.</b> Kinect sürümleri arasındaki teknik farklılıklar .....	34
<b>Tablo 3.4.</b> Kinect v2 25 iskelet eklem noktası isimleri .....	35
<b>Tablo 3.5.</b> Kinect v2 cihazı SDK mimarisi .....	36
<b>Tablo 3.6.</b> Kinect v2 temel veri erişim aşamaları.....	36
<b>Tablo 3.7.</b> Kinect v2 veri akışı görüntü özellikleri.....	37
<b>Tablo 3.8.</b> Kinect v2 cihazı koordinat eşleyicileri.....	37
<b>Tablo 3.9.</b> Kinect v2 veri çerçeve kaynakları ve özellikleri.....	37
<b>Tablo 3.10.</b> LMC & Kinect arasındaki teknik farklılıklar (Pauchot vd., 2015) .....	38
<b>Tablo 5.1.</b> LMC tarafından desteklenen önemli el ve parmak özellikleri .....	40
<b>Tablo 5.2.</b> Birinci özellik çıkarımının ölçü birimi, türü, isimleri ve sayısı .....	43
<b>Tablo 5.3.</b> İkinci özellik çıkarımının ölçü birimi, türü, isim ve adetleri .....	44
<b>Tablo 5.4.</b> Börek sözcüğüne ait sağ ve sol el avuç içi noktası vektör bilgileri.....	45
<b>Tablo 5.5.</b> Kinect iskelet eklemlerinin özellik çıkarım bilgisi .....	47
<b>Tablo 5.6.</b> Börek sözcüğünün 8 adet kollara ait iskelet eklem açısı bilgileri.....	49
<b>Tablo 6.1.</b> Yapılışı itibarıyla birbirine benzer dinamik sözcükler.....	59
<b>Tablo 6.2.</b> 15 özellik alt kümesi seçimi için tepe değeri ve özellik matrisi sayısı .....	61
<b>Tablo 6.3.</b> Kinect cihazı için kullanılan dinamik sözcükler .....	64
<b>Tablo 6.4.</b> LMC ve Kinect cihazı için kullanılan dinamik sözcükler.....	67
<b>Tablo 6.5.</b> Belirlenen özelliklerin cihaz, ölçü birimi, türü, isim ve adetleri.....	68
<b>Tablo 6.7.</b> Histogram için belirlenen özellik çıkarımlarına ait bilgiler ve aralık sayıları.....	69
<b>Tablo 7.1.</b> Sinir ağı tabanlı sınıflandırıcılarda kullanılan fonksiyon, katman ve hücre sayısı	75

<b>Tablo 7.2.</b> ELM tabanlı sınıflandırıcılarda kullanılan fonksiyon, katman ve hücre sayısı.....	78
<b>Tablo 8.1.</b> Orijinal veri kümesi ile sınıflandırma başarımları ve süreleri .....	80
<b>Tablo 8.2.</b> PCA uygulanmış veri kümesi ile sınıflandırma başarımları ve süreleri .....	81
<b>Tablo 8.3.</b> LDA uygulanmış veri kümesi ile sınıflandırma başarımları ve süreleri .....	81
<b>Tablo 8.4.</b> PCA+LDA uygulanmış veri kümesi ile sınıflandırma başarımları ve süreleri .....	82
<b>Tablo 8.5.</b> Veri kümelerinin oluşum aşamaları .....	83
<b>Tablo 8.6.</b> 75 özellik içeren veri kümelerinin sınıflandırma başarımları ve süreleri .....	84
<b>Tablo 8.7.</b> 50 özellik içeren veri kümelerinin sınıflandırma başarımları ve süreleri .....	84
<b>Tablo 8.8.</b> 25 özellik içeren veri kümelerinin sınıflandırma başarımları ve süreleri .....	85
<b>Tablo 8.9.</b> 75 özellik içeren veri kümelerinin sınıflandırma başarımları ve süreleri .....	87
<b>Tablo 8.10.</b> 75 özellik içeren veri kümelerinin sınıflandırma başarımları ve süreleri .....	88
<b>Tablo 8.11.</b> Histogram özellik çıkarımına ait veri kümelerinin sınıflandırma başarımları ve süreleri .....	90
<b>Tablo 8.12.</b> PCA+LDA eğitim veri kümesinin ML-KELM ait tanıma performansı için karışıklık matrisi.....	91
<b>Tablo 8.13.</b> PCA+LDA test veri kümesinin ML-KELM ait tanıma performansı için karışıklık matrisi.....	91
<b>Tablo 8.14.</b> Zamansal özellik çıkarımına ait veri kümelerinin sınıflandırma başarımları ve süreleri .....	92
<b>Tablo 9.1.</b> Uygulamalarda önerilen tanıma modeli aşamalarının özet gösterimi .....	94

## ŞEKİLLER LİSTESİ

Şekil 3.1. TİD alfabesinde A Harfi (Güngör vd., 2015) .....	26
Şekil 3.2. TİD hareketleri (Sözcük) (Güngör vd., 2015) .....	26
Şekil 4.1. LMC cihazının görünümü (LMC) .....	28
Şekil 4.2. LMC cihazının etkileşim hacmi ve görüş alanı (LMC).....	29
Şekil 4.3. LMC el ve parmak noktaları (LMC).....	30
Şekil 4.4. LMC SDK mimarisine genel bakış (LMC).....	30
Şekil 4.5. LMC sağ el koordinat sistemi (LMC).....	32
Şekil 4.6. El koordinat sistemi (LMC).....	32
Şekil 4.7. Kinect v2 görünümü .....	33
Şekil 4.8. Kinect 25 noktalı iskelet yapısı görünümü .....	35
Şekil 5.1. LMC tarafından desteklenen el ve parmakların özellik gösterimi (LMC) .....	41
Şekil 5.2. Belirlenen özellik çıkarımlarına ait eklem noktaları ve açı bilgileri .....	48
Şekil 6.1. Statik parmak alfabesi için önerilen tanıma modelinin iş akışı .....	56
Şekil 6.2. Çift elle gerçekleştirilen 17 statik TİD parmak alfabesi .....	57
Şekil 6.3. LMC parmak alfabesi tanıma uygulaması örnek gösterimi.....	58
Şekil 6.4. LMC tabanlı dinamik sözcükler için önerilen tanıma modelinin iş akışı.....	59
Şekil 6.5. "Fırın" (üst sıra) ve "Gemi" (alt sıra) sözcüklerin yapılaş görüntü dizileri (TID)....	60
Şekil 6.6. Benzer sözcüklerin işaret hareketleri için kutu grafiği. ....	62
Şekil 6.7. Benzer sözcüklerin işaret hareketleri için 2D çizim. ....	62
Şekil 6.8. Kinect tabanlı dinamik sözcükler için önerilen tanıma modelinin iş akışı .....	63
Şekil 6.9. Kinect dinamik sözcük uygulaması örnek gösterimi .....	65
Şekil 6.10. LMC+Kinect tabanlı dinamik sözcükler için önerilen tanıma modelinin iş akışı .	66
Şekil 6.11. Açı bilgisine ait aralık kümesinin 12 eşit aralığa bölünmüş gösterimi.....	70
Şekil 6.12. Börek sözcüğüne ait sağ dirsek açı histogramı .....	70
Şekil 6.13. LMC+Kinect dinamik sözcük uygulaması örnek bir gösterimi.....	71

**Şekil 8.1.** LMC ve Kinect cihazları görüş alanı gösterimi..... 89



## KISALTMALAR VE SİMGELER LİSTESİ

- LMC : Leap Motion Controller (Leap Motion Kontrolör)
- LMS : Leap Motion Service (Leap Motion Hizmeti)
- k-NN : K-Nearest Neighbor (K-En Yakın Komşu)
- ANN : Artificial Neural Networks (Yapay Sinir Ağı)
- NBC : Naive Bayes Classifier (Naif Bayes Sınıflandırıcı)
- HMM : Hidden Markov Model (Saklı Markov Modelleri)
- CHMM : Coupled Hidden Markov Model (Birleşik Gizli Markov Modeli)
- FCM : Fuzzy-C-Means (Fuzzy-C-Means Algoritması)
- CGP : Cartesian Genetic Programming (Kartezyen Genetik Programlama)
- SVM : Support Vector Machine (Destek Vektör Makinesi)
- MLP : Multilayer Perceptron (Çok Katmanlı Algılayıcı)
- TİD : Turkish Sign Language (Türk İşaret Dili)
- ASL : American Sign Language (Amarikan İşaret Dili)
- AUSL : Australia Sign Language (Avustralya İşaret Dili)
- ISLS : Indonesian Sign Language System (Endonezya İşaret Dili Sistemi)
- ISL : Indonesian Sign Language (Endonezya İşaret Dili)
- SIBI : Sistem Isyarat Bahasa Indonesia Sign Language (Endonezya İşaret Dili)
- GSL : German Sign Language (Alman İşaret Dili)
- BSL : British Sign Language (İngiliz İşaret Dili)
- JSL : Japanese Sign Language (Japon İşaret Dili)
- TSL : Thai Sign Language (Tayland İşaret Dili)
- BSL : Brazilian Sign Language (Brezilya İşaret Dili)
- ArSL : Arabic Sign Language (Arap İşaret Dili)
- API : Application Programming Interface (Uygulama Programlama Arayüzü)
- SDK : Software Development Kit (Yazılım Geliştirme Kitleri)

DLL : Dynamic Link Library (Dinamik Bağlantı Kütüphanesi)

HTTP : Hyper Text Transfer Protocol (Köprü Metni Aktarım Protokolü)

JSON : JavaScript Object Notation (JavaScript Nesne Gösterimi)

WPF : Windows Presentation Foundation (Windows Sunum İskeleti)

CHT : Circle Hough Transform (Dairesel Hough Dönüşüm)

DLT : Direct Linear Transformation (Doğrudan Lineer Dönüşüm)

PCA : Principal Component Analysis (Temel Bileşen Analizi)

LDA : Linear Discriminant Analysis (Doğrusal Ayırım Analizi)

SIFT : Scale Invariant Feature Transform (Değişmeyen Ölçek Özellik Dönüşüm)

LBP : Local Binary Patterns (Yerel İkili Desenler)

GKD : Gradient Kernel descriptor (Gradyan Çekirdek Tanımlayıcısı)

HOG : Histograms of Oriented Gradients (Gradyan Odaklı Histogram)

DCT : Discrete Cosine Transform (Ayrık Kosinüs Dönüşüm)

DOG : Difference Of Gaussians (Ayrık Kosinüs Dönüşüm)

GHT : Generalised Hough transform (Genelleştirilmiş Hough Dönüşüm)

RBF : Radial Basis Function Network (Radyal Temelli Fonksiyon ağları)

FFT : Fast Fourier Transform (Hızlı Fourier Dönüşümü)

DWT : Discrete Wavelet Transform (Ayrık Dalgacık Dönüşümü)

RMS : Root Mean Square (Karesel Ortalamaların Karekökü)

STD : Standard Deviation (Standart Sapma)

AVG : Average (Ortalama)

ELM : Extreme Learning Machine (Aşırı Öğrenme Makinesi)

KELM : Kernel Extreme Learning Machine (Çekirdek Tabanlı ELM)

MLP-ELM: Multi Layer Perceptron ELM (Çok Katmanlı Algılayıcı ELM)

ML-ELM : Multi Layer ELM (Çok Katmanlı ELM)

ML-KELM : Multi Layer Kernel ELM (Çok Katmanlı Çekirdek Tabanlı ELM)

SLFN : Single-hidden Layer Feedforward Neural Network  
(Tek Gizli Katmanlı İleri Beslemeli Sinir Ađı)

EMG : Elektromiyogram

ANFIS : Adaptive-Network Based Fuzzy Inference Systems  
(Uyarlamalı Sinirsel Bulanık Çıkarım Sistemi)

GMM : Gaussian Mixture Model (Gaussian Karışım Modeli)

DBN : Deep Belief Network (Derin İnanç Ađı)

CC : Cross Correlation (Çapraz Korelasyon)

RF : Random Forest (Rastgele Orman)

BPGANN : Backpropagation Genetic Neural Network  
(Geri Yayılım Genetik Sinir Ađı)

RB-BPGANN : Rule Based Backpropagation-Genetic Neural Network  
(Kural Tabanlı Geri Yayılım Genetik Sinir Ađı)

CNN : Convolutional Neural Networks (Evrışimsel Sinir Ađları)

LSTM : Long Short-Term Memory (Uzun-Kısa Vadeli Bellek)

SLR : Sign Recognition System (İşaret Tanıma Sistemi)

BLSTM-NN: Bidirectional Long Short-Term Memory Neural Networks  
(Çift Yönlü Uzun Kısa Süreli Hafızalı Sinir Ađı)

DT : Decision Tree (Karar Ađaçları)

YSA : Artificial Neural Networks (Yapay Sinir Ađı)

ÇKYSA: Multi Layer Artificial Neural Networks (Çok Katmanlı Yapay Sinir Ađı)

FFNN : Feedforward Neural Network (İleri Beslemeli Sinir Ađı)

RNN : Recurrent Neural Network (Geri Beslemeli Sinir Ađı)

PNN : Probabilistic Neural Network (Olasılıksal Sinir Ađı)

CFNN : Cascade-Forward Neural Network (Kaskat İleri Beslemeli Yapay Sinir Ađı)

## 1. GİRİŞ

Teknolojinin gelişimine bağlı olarak insan ve bilgisayar etkileşimini daha kolay ve kullanışlı yapmak için akıllı arabirimler geliştirilmektedir. İnsan bilgisayar etkileşimi içinde işletim sistemleri, yazılımlar, robotlar ve otonom araçlar başta olmak üzere bunların kontrollerine yönelik araştırma ve uygulama çalışmaları yapılmaktadır. Bu sistemlerin komuta edilebilmesi ve etkileşimlerinin iyi ve sağlıklı bir şekilde sağlanmasında ses ve görüntü sistemlerinin yanı sıra el, parmak, yüz ve vücut gibi görsel işaret hareketleri de kullanılmaktadır. Son yıllarda özellikle el ve parmak hareketleri ile denetlenebilen akıllı arabirimler kullanıldığı çalışmaların sayısında önemli artışlar görülmektedir. Bunlar arasında, işitme ve konuşma engellilerin kullandıkları işaret dillerinin tanınmasına yönelik çalışmalarda mevcuttur. Özellikle kullanım kolaylığı, taşınabilirliği ve basitliği olan akıllı arabirimler tercih edilmektedir. Bu sistemleri tasarlamak insan, bilgisayar ve arabirimler arası etkileşim açısından çok önemli bir gelişmedir.

İşaret dili, işitme engelli kişilerin birbirleriyle ve diğer kişiler arasındaki iletişimi sağlamak için kullanılan önemli bir iletişim aracıdır. İşaret dili, işitme engellilerin ana dili olarak tanımlanır ve iletişimlerinin çok büyük bir bölümünü bu dil ile sağlarlar. Dünya Sağlık Örgütü'nün istatistiklerine göre, işitme engelli insan sayısı dünya üzerinde 466 milyon, Türkiye'de ise 3 milyona yaklaşmaktadır. Bu sayı gün geçtikçe artmakta olup %90'ının gerekli cihazlara sahip olmadığı bilinmektedir. Bu sebeple bilgisayar sistemleri tarafından işaret dillerinin tanınması, yorumlanması hem sosyal hem de teknolojik açıdan önem arz etmektedir.

İşaret dili tanıma konusunda makine öğrenmesi yöntemleri ile birçok tanıma sistemi tasarlanmış ve geliştirilmiştir. Bu sistemler elektronik eldiven tabanlı (Öz ve Leu, 2007), görüntü tabanlı (Starner ve Pentland, 1995), Kinect tabanlı (Zhu ve Wong, 2012), EMG tabanlı (Kosmidou vd., 2006), LMC tabanlı (Mapari ve Kharat, 2017) ve bunların birlikte kullanımları ile çoklu model tabanlı (Kumar vd., 2017) sistemler olarak kategorize edilebilir. LMC dışındaki sistemlerin her ne kadar kendine özgü üstünlükleri olsa da günlük yaşam içinde kullanılabilirlikleri çok düşüktür. Bu sebeple çalışmamızda özellikle günlük yaşamda taşınabilir bir sistem geliştirilmesi için LMC tercih edilmiştir.

Bu çalışmada, dijital bir sensör sistemine dayalı hareket temelli bir platform olan LMC ve Kinect cihazı kullanılarak Türk işaret dili (TİD) çift elle gerçekleştirilen statik (posture) parmak alfabesi harflerinin ve dinamik (gesture) sözcüklerin tanınması üzerine çalışılmıştır.

LMC cihazının benzer sensör ve kameralara göre daha ucuz, hafif ve kolay taşınabilir olması ve günümüzde kullanılan her türlü teknolojik cihaza kolayca gömülü olarak ya da doğrudan entegre edilebilir olması bu çalışma için tercih edilmesinin sebebidir. Kinect insan iskeleti odaklı bir cihazken, LMC el, parmak ve eklem odaklı bir cihaz olması dolayısıyla parmak hareketlerini tanıma önemli bir üstünlük sağlamaktadır. Bu cihazın sayılan bu özellikleri işitme engelli kişilerin her an her yerde diğer kişiler ile kolayca iletişim kurabilmesi açısından önemlidir. Literatürde LMC tabanlı statik TİD parmak alfabesi üzerine çok az sayıda çalışma bulunurken, dinamik sözcük tanıma üzerine çalışma bulunmamaktadır. Bunun yanı sıra LMC tabanlı diğer dillerde dinamik sözcük tanıma çalışması az sayıda bulunmaktadır. Bu çalışmalarda ASL, ArSLR, ISL ve SIBI gibi dillerin sözcükleri üzerinde çalışılmıştır.

İşaret dillerine ait el ve parmak hareketleri statik ve dinamik hareketlerden oluşmaktadır. Bu çalışmada, TİD'e ait çift elle gerçekleştirilen statik parmak alfabesi ve üç adet dinamik sözcük tanıma sistemi tasarlanmıştır. Sistemin önerilen tanıma modeli LMC ve Kinect cihazlarından veri alma, veri işleme ve uygulama olmak üzere 3 ana aşamadan oluşmaktadır. Veri işleme aşamasında farklı ve benzer çeşitli dinamik hareketleri ayırt etmemizi sağlayacak sinyal işleme teknikleri kullanan yeni bir özellik seçim algoritması kullanılmıştır. Bununla birlikte farklı uzunluk ve sürelerle sahip dinamik sözcüklerin uzunluklarının eşitlenmesinde 2 tür özellik çıkarım yöntemi kullanılmıştır. Bu özellik seçim ve çıkarımlardan elde edilen veri kümelerine PCA, LDA ve PCA+LDA boyut indirgeme yöntemleri kullanılarak yeni bir yaklaşım sunulmuştur. Bu iki aşamalı süreç içerisinde, özellik sayısı azaltarak ve boyut küçülterek, oluşturulan veri kümeleri hesaplama, süre, maliyet, depolama ve başarımlardan elde edilen veri kümelerine PCA, LDA ve PCA+LDA boyut indirgeme yöntemleri kullanılarak yeni bir yaklaşım sunulmuştur. Bu iki aşamalı süreç içerisinde, özellik sayısı azaltarak ve boyut küçülterek, oluşturulan veri kümeleri hesaplama, süre, maliyet, depolama ve başarımlardan elde edilen veri kümelerine PCA, LDA ve PCA+LDA boyut indirgeme yöntemleri kullanılarak yeni bir yaklaşım sunulmuştur.

Çalışmada geliştirilen sistem ile kolay etkileşim kurmak için özellikle kullanım kolaylığı ve sistem basitliği ön plana alınmış olup taşınabilir kullanılması hedeflenmiştir. Etkileşimde kullanılan öğelerin bilgisayar sistemleri tarafından yüksek başarımları ile tanınması da gerekmektedir. Bu çalışmada sistemlerin eğitilmesi ve tanınması aşamasında güncel ve ilgi duyulan makine öğrenmesi yöntemi olan ELM tabanlı algoritmalar kullanılmıştır.

## 2. LİTERATÜR TARAMASI

Bu bölümde literatür çalışmaları elektronik eldiven tabanlı sistemler, kamera tabanlı sistemler, Elektromiyogram (EMG) tabanlı sistemler, Myo Armband tabanlı sistemler, Kinect tabanlı sistemler, LMC tabanlı sistemler ve TİD üzerine yapılan çalışmalar olmak üzere yedi alt başlıkta incelenmiştir.

### 2.1. Elektronik Eldiven Tabanlı Çalışmalar

İşaret dillerinin algılanmasına yönelik yapılan ilk çalışmalarda elektronik eldiven tabanlı sistemler kullanılmıştır. Sistemi kullanan kişinin işaret diline ait hareketleri yapması suretiyle alınan verilerden dinamik tanıma sistemi geliştirmiştir. Japon işaret diline ait 10 sözcüğün yapay sinir ağları (YSA) yöntemi kullanarak tanınması %98 oranında başarımla sağlanmıştır (Murakami ve Taguchi, 1991). Saengsri ve arkadaşları (Saengsri vd., 2012)'de, elektronik eldiven ve hareket izci cihazı kullanılarak Tayland işaret dili tanıma sistemi önermiştir. Tayland işaret diline ait tek elle gerçekleştirilen parmak heceleme alfabesi hareketleri kullanarak dinamik tanıma sistemi tasarlamışlardır. Elektronik eldivenin 14 farklı noktasında bulunan sensör ve diğer 4 tane hareket izci cihazından alınan 20 değer için toplam 1000 örnek değerlendirilmiştir. Sistemin doğruluğunu artırmak için YSA ve veri bölütleme teknikleri kullanılarak %94 oranında başarımla sağlanmıştır. Tangsuksant ve arkadaşları (Tangsuksant vd., 2014)'deki çalışmada, 6 farklı renkli belirteçlere sahip elektronik eldiven ve 2 kamera kullanarak ASL'e ait hareketlerin tanınmasına yönelik bir statik tanıma sistemi geliştirmiştir. Renkli belirteçlerin tespit edilmesi dairesel Hough dönüşüm tekniği kullanılarak sağlanmıştır. Belirteçlerin 3D koordinat noktalarını 2 kamera eklenerek tespit etmede doğrudan lineer dönüşüm (DLT) algoritması kullanılmıştır. ASL'ye ait 24 adet harf üzerinde yapılan çalışmada YSA sınıflandırıcı ile % 95 doğruluk oranı yakalanmıştır. Belirgin bir gelişme olarak (Wang vd., 2006)'daki çalışmada, çok boyutlu Gizli Markov Modelleri (HMM) temel alınarak geliştirilen gerçek zamanlı ASL tanıma sistemi sunulmuştur. Özellikleri çıkarmak için elektronik eldiven ve 3 boyutlu hareket izleyicisi kullanılmıştır. ASL'e ait olan 26 alfabe ve 36 temel el tipi için ortalama % 95 doğruluk elde edilmiştir. Öz ve arkadaşlarının (Öz ve Leu, 2007)'deki çalışmasında, ASL sözcüklerini tanımak için YSA kullanılarak geliştirilen 2 farklı sözcük tanıma sistemi sunmuşlardır. Özellikler ASL sözcüklerinin dilbilimsel özelliklerine dayanarak sensör verileri kullanılarak çıkarıldığı bildirilmektedir. İlk sistemde 5 kez anlık sözcüklerin özellik vektörleri kullanılmış, ikinci sistemde ise özellik vektörlerinin histogramları kullanılmıştır. Sistem özellik çıkarımı için

elektronik eldiven ve 3 boyutlu hareket izleyicisi kullanır. Eldiven içindeki gerilimölçerlerden elde edilen parmak eklem açısı verileri el şeklini tanımlar ve izleyiciden gelen veriler ise el hareketinin yörüngesini tanımlar. Her 2 sistemde, bu cihazlardan gelen veriler iki sinir ağı tarafından işlenmiştir: Birincisi hız ağı olarak adlandırılmış olup, sözcüklerin süresini belirlemek için el hızını kullanır. İşaretler, el şekli, el yeri, yönlendirme, hareket, sınırlayıcı kutu ve mesafe gibi özellik vektörleriyle tanımlanır. İkinci ağ olan sözcük tanıma ağı ise, ASL işaretlerinin özellikleri veya histogramlarının özelliklerine dayalı sözcüklere dönüştürmek için bir sınıflandırıcı olarak kullanılmıştır. Toplamda 7 hareket özelliği altında ilk sistemde 423 giriş sağlanırken, ikinci sistemde histogram sayesinde 91 girişe düşürülmüştür. 360 örnek veri kümesi kullanılarak 60 ASL sözcüğü için YSA modelleri eğitilip test edilmiştir. Bu iki sistem birbirleriyle karşılaştırılarak, tanımanın doğruluğunun sırasıyla % 92 ve % 95 olduğunu göstermiştir. Öz ve arkadaşlarının (Öz ve Leu, 2011)'deki benzer çalışmasında, YSA kullanarak elektronik eldiven ile ASL sözcük tanıma sistemi önerilmiştir. Toplamda 7 hareket özelliği altında 151 özellik çıkarımı belirlenmiştir. Öncelikle sistemin gürültüyü azaltmak için bir filtre, sonra işaretin bir ASL sözcüğü olup olmadığı belirlemek için bir hız ağı kullanılmıştır. 50 ASL sözcük grubu için tek ve çoklu kullanıcılar için YSA eğitilmiş ve test edilmiştir. Test sonuçları, sistemin tanıma doğruluğunun yaklaşık %90 olduğunu göstermiştir. Sarawate ve arkadaşlarının (Sarawat vd., 2015)'deki benzer çalışmasında, YSA'ya dayalı bir ASL sözcük tanıma sistemi ve olasılıksal bir model geliştirilmesi sunulmuştur. El şekli, el konumu, el yönelimi ve el hareketi olmak üzere 4 hareket özelliği, geri yayılım sinir ağlarını içeren farklı fonksiyonlar kullanılarak tanınması sağlanmıştır. Bu özelliklerin sırası, olasılıklı bir modele dayalı olarak ASL'de belirli bir işaret veya sözcük oluşturmak için kullanılmıştır. Sistem, ASL işaretlerini gerçek zamanlı olarak tanıma ve veritabanını etkileşimli olarak güncellemektedir. 40 ASL sözcük kullanılarak %95 doğruluk oranına ulaşmıştır.

## **2.2. Kamera Tabanlı Çalışmalar**

İşaret dillerinin kamera tabanlı olarak tanınmasına yönelik ilk dikkat çeken çalışma (Starner ve Pentland, 1995)'dir. Bu çalışmada, kamera kullanarak ASL'e ait sözcükleri gerçek zamanlı tanımaya çalışılmıştır. Alınan görüntülerde eller bölütlenmiş ve ellere ait öznitelik vektörleri HMM yöntemi ile dinamik işaret tanıma işlemi gerçekleştirilmiştir. ASL'e ait 494 adet sözcük üzerinde yapılan çalışmada %90'ın üzerinde başarımlar sağlanmıştır. Starner (Starne vd., 1997), aynı zamanda işaret dili tanınması yapabilen bir giyilebilir bilgisayar tabanlı sistem sunmuştur. Bu çalışmada bir şapkaya yerleştirilen kameradan alınan görüntülerin ellere ait özellikler bölütlenerek tanıma yapılmıştır. 40 sözcükten oluşan bir işaret veri kümesi üzerinde

HMM yöntemi kullanılarak bir başarıml ölçümü yapılmış ve %99 oranında başarıml elde edilmiştir. Silanon ve arkadaşlarının (Sılanon ve Suvonvorn, 2014)'deki çalışmasında, bulanık parmak şekil ve el görünüm analizine dayalı alternatif bir esnek statik işaret tanıma yöntemi sunmuştur. Kamera yardımıyla alınan görüntülerde, eller bölütlenerek ellere ait derinlik, uzaklık, şekil, konum ve alan bilgilerinin görünüm özelliklerinden bulanık mantık kullanılarak tanıma gerçekleştirilmiştir. Sistemin başarımlı 26 ASL parmak alfabesini kullanarak gerçek zamanlı bir ortamda test edilerek 70% tanıma oranı elde edilmiştir. ASL'ye ait parmak alfabesi hareket imgelerinin tanınmasına yönelik çalışma olan (Amin ve Yan, 2007) ise, harfin olduğu bölge normalleştirilerek, Radon dönüşümü ile hizalanarak Gabor süzgeçlerinden geçirilmiştir. Temel bileşen analizi (PCA) ile boyut olarak indirgenen imgeler üzerinde %93 oranında tanıma gözlemlenmiştir. Çalışmada sınıflandırma amacıyla Fuzzy-c-Means (FCM) algoritması kullanmıştır. Bilgisayar teknolojisinin gelişimi ile akıllı bilgisayar destekli farklı uygulamalar ile el hareketlerinin tanınmasına yönelik çalışmalar yapılmaktadır. Bu yönde (Ullah, 2011)'de ASL'ye ait işaretler Kartezyen Genetik Programlama (CGP) tekniği kullanarak tanınmaya çalışılmıştır. Görüntüler ikili (binary) formata dönüştürüldükten sonra CGP ağ tekniği yardımıyla işareti tanıma sistemi geliştirilmiştir. Bu sistem, 26 Amerikan alfabesini temsil eden el hareket imgelerinin bu sistem yardımı ile farklı test ve eğitim veri kümesi hazırlanarak test edilmiş olup % 90 doğruluk oranı başarımlı elde edilmiştir. Munib ve arkadaşları (Munib vd., 2007)'de, ASL'deki alfabe ve işaretlerin statik hareketlerinin otomatik olarak tanınması için bir sistem geliştirmeyi amaçlamışlardır. Bu sistemde, işaretleri tanımak için eğitilmiş Hough dönüşümü ve YSA yöntemi kullanılmıştır. Ellerin görüntüleri işlenerek, özellik vektörüne dönüştürülür ve bir işaret eğitim kümesinin özellik vektörleriyle karşılaştırılır. Elde edilen sonuçlar sistemin, 300 adet el işareti imgesi veri kümesi kullanılarak seçilen ASL işaretlerini% 92 doğruluk oranı ile tanıdığını göstermiştir. Arap işaret dili (ArSL) alfabesi üzerine yapılan bir çalışmada (El-Bendar vd., 2010) el hareketlerinin otomatik bir çeviri sistemi sunulmaktadır. Önerilen ArSL alfabetik sistemi, kullanıcının sistemle doğal bir şekilde etkileşime girmesine izin veren çıplak ellerin görüntülerini kullanır. Önerilen bu sisteminin, en küçük uzaklık yöntemi kullanılarak % 91'lik bir doğruluk oranı ile 30 Arap alfabesini tanıdığı bildirilmiştir. Mukai ve arkadaşları (Mukai vd., 2017)'deki çalışmasında, toplamda 287 sayısal kamera görüntüsünden oluşan hareketsiz 41 Japon alfabesi için tanıma yöntemi önermişlerdir. Bazı parmak alfabesi sınıflandırma ağacı modeli ile kolaylıkla tanınırken, bazıları için ise SVM modeli tercih edilmiştir. Araştırma sonuçlarına göre ortalama tanıma oranı %86 olarak bulunmuştur. Özbay ve arkadaşı, iki farklı

yaklaşım kullanılarak evrensel bir işaret dili tanıma sistemi önermiştir. Önerilen sistem tanınmayı evrensel hale getirmek için yaygın olarak kullanılan ASL, İngiliz işaret dili (BSL) ve TİD olarak üç işaret dili alfabesi için geliştirmiştir. Önerilen yaklaşımlardan biri öncelikle Hausdorff mesafesi ve Hu değişmez (invariant) özellik vektörlerine dayanmaktadır. İkinci yaklaşım, ileri beslemeli bir YSA yapısı ile gerçekleştirilir. Her iki yaklaşımın başarılı bir şekilde çalıştığı, deneysel sonuçların ise YSA yönteminin daha üstün başarımlar sağladığı bildirilmiştir. Tasarlanan sistem, Microsoft Visual Studio ve Open CV kullanarak C ++ dili ile gerçekleştirilmiştir (Özbay ve Safar, 2017). Al-Jarrah ve arkadaşı, 30 ArSL alfabesi tanıma sistemi geliştirmişlerdir. Önerilen ANFIS modeli yaklaşımı ile %93 oranında başarımlar elde edilmiştir (Al-Jarrah ve Halawani, 2001). Kelly ve arkadaşları, işaret dilinde kullanılan el duruşlarının tanınması için bağımsız bir sistem önermişlerdir. İrlanda işaret dili alfabesi için destek vektörü makine tabanlı tanıma sistemi kullanmışlardır (Kell vd., 2010). Dahmani ve arkadaşı, ArSL alfabesi üzerine benzer bir çalışma yapmışlardır (Dahmani ve Larabi, 2014). Bilgin ve arkadaşı (Bilgin ve Mutludogan, 2019)'deki çalışmasında, işaret dili karakterlerinin ASL'deki harflerin görüntüleri kullanılarak eğitilmiş bir sistem aracılığıyla tanınması amaçlanmıştır. Son zamanlarda önerilen kapsül ağlar, eğitim ve test süreçleri için kullanılarak ve bir derin öğrenme modeli kullanan LeNet ile karşılaştırılmıştır. Çalışma kapsül ağların işaret dili tanıma için LeNet'e göre daha başarılı sonuçlar ürettiği görülmüştür. Hasan ve arkadaşları (Hasan vd., 2020)'de, ASL işaret dili alfabelerinin başarılı bir şekilde tanımlanmasını sağlamak için derin bir evrişimli sinir ağı modeli önermişlerdir. Model, MINST veri kümesi kullanılarak %97 gibi bir genel doğruluk oranı sağlamış ve öncekiler ile karşılaştırıldığında, önerilen modelin daha daha iyi performans gösterdiği görülmüştür. Benzer bir çalışmada giriş görüntüsü alınarak bir dizi katmandan geçirilerek çıktı üretilen derin CNN (Inception V3 modeli) kullanılarak bir görüntü tanıma modeli uygulanmıştır. Hint işaret dili (ISL) 23 sözcük kullanarak %93 doğruluk elde edilmiştir (Varsha ve Nair, 2021). Sincan ve arkadaşları, evrişimsel sinir ağları (CNN), öznitelik biriktirme modülü (FPM) ve uzun kısa-vadeli hafıza ağları (LSTM) birlikte kullanılarak bir ayrık işaret dili tanıma sistemi önermiştir. Bu model ile Montalbano veri kümesinde %93 doğruluk oranı yakalanmıştır (Sincan vd., 2019). Tasmere ve arkadaşları (Tasmere ve Ahmed, 2020)'deki çalışmasında, Bangla işaret diline yönelik yeni bir el hareketi tanıma çerçevesinde toplamda 37 harf derin evrişim sinir ağları tarafından tanınmıştır. 6 farklı denekten alınan 3219 görüntü karesi ile elde edilen veri kümesi için %99 doğruluk elde edilmiştir.

### **2.3. Elektromiyogram (EMG) Tabanlı Çalışmalar**

Kosmidou ve arkadaşlarının (Kosmidou vd., 2006)'daki çalışmasında; ASL hareketlerinin tanınması için yüzey elektromiyogramı (sEMG) sinyalinin analizi önerilmiştir. Kullanıcının önkolundan alınan 2 kanallı sEMG sinyallerinden çıkartılan 16 özellik, 9 ASL hareketinin tanınması için değerlendirilmiştir. Öklid mesafe ölçütü metriğine dayalı ayırık analiz içeren istatistiksel süreçten sonra, yeterli ayırıştırıcı kabiliyetine sahip olan özellikler on bire indirilmiştir. Sınıflandırma sonuçları, incelenen ASL hareketlerinin % 97'sinin sEMG tabanlı özellikler kullanılarak doğru bir şekilde tanındığını ve otomatik ASL hareket tanıma sorununa umut verici bir çözüm sağladığını ortaya koymaktadır. Amatanon ve arkadaşları (Amatanon vd., 2014)'de, 10 Tayland işaret dili alfabesi hareketi EMG sinyallerini, Myo Arm cihazına benzer elektrot düzenlemesi ile kaydedilmiş ve analiz edilmiştir. 8 kanallı EMG sinyallerinin farklı özelliklerini ayıklamak için, hareket eden varyans ve ortalama mutlak değer seçilmiştir. Sınıflandırıcı olarak doğrusal olmayan model olan YSA yoluyla sınıflandırma algoritması seçilmiştir. Sistem, doğruluk oranı % 95 üzeri olarak ölçülmüştür. Kim ve arkadaşlarının (Kim vd., 2008)'deki çalışmasında; Alman işaret dilindeki (GSL) yedi sözcük bilgisini tanımak için ivmeölçer (ACC) ve elektromiyogramın (EMG) karşılıklı tamamlayıcı işlevselliğini araştırılmıştır. En uygun onbeş özelliği kullanarak, ortalama doğruluk oranı % 96 olarak tüm denekler için kanıtlanmıştır.

### **2.4. Myo Armband Tabanlı Çalışmalar**

İşaret dili jestlerinin bilgisayar sistemleri tarafından başarılı bir şekilde tanınması, sağır ve dinleyiciler arasındaki iletişimi büyük ölçüde artıracaktır. Abreu ve arkadaşlarının (Abreu vd., 2016)'daki çalışmasında, Myo Arm kol bandı tarafından sağlanan elektromiyogram (EMG) verileri, 20 Brezilya işaret dili alfabesi hareketlerinin sınıflandırılması aşamasında özellik çıkarımı için kullanılmıştır. Sınıflandırma, bir bütünlük stratejisi ile eğitilmiş ikili SVM tarafından gerçekleştirilmiştir. Elde edilen sonuçlar, alfabenin tanımlamanın mümkün olduğunu göstermiş, ancak ileri çalışmalar ile ele alınması gereken önemli sınırlamalar olduğu da bulunmuştur. Madushanka ve arkadaşları (Madushanka vd., 2016)'da, Sinhala işaret dili hareketlerini tanımak ve bunları doğal dile çevirmek için bir sistem önermiştir. Çözümün hem işlevsel hem de kullanılabilirlik yönlerini korumak için, Myo Armband cihazı kullanmıştır. Bu yaklaşım, işaretin tanınması için kol kaslarını, hareketlerini ve mekansal verileri (ivme ölçer, jiroskop ve yönlendirme) ölçen bir hareket verileri kombinasyonunu kullanmaktadır. Sistem makine öğrenimi tekniği altında çoklu YSA

uygulanarak gerçekleştirilmiştir. Sonuç olarak, çalışma kişiye bağımlı (kişiselleştirilmiş) çalışma için %100 doğruluk ve bağımsız kişi (genelleştirilmiş) çalışma için %94 doğruluk sağlamıştır. Vachirapipop ve arkadaşları, Myo Arm kolluklarını kullanarak 6 Tayland işaret dili sözcüğünü tanıtmaya amaçlamıştır. Oluşturulan bu tahmin modelinden en iyi doğruluk oranını test etmek için kullanılan algoritmalar karar ağacı, sıralı minimal optimizasyon ve çok katmanlı algılayıcıdır. Ek olarak, en uygun özellik seçimini test etmek için ortalama ve standart sapma kullanılır. Testten sonra en iyi sonucu veren tahmin modeli, en uygun özelliklere sahip olan ortalama ve standart sapmaya sahip MLP ve SMO algoritmaları olmuştur. Geliştirilen bu iki model için toplamda 84 öznelikten 30 tanesi kullanılarak %100'lük bir doğruluk sağlanmıştır (Vachirapipop vd., 2017). Rahagiyanto ve arkadaşları (Rahagiyanto vd., 2017)'de; Statik ve dinamik hareketlerle kullanıcıya bağlı 26 SIBI alfabesi hareketleri için 10 değişken kullanılarak test edilmiştir. Çalışmada, el hareketinin vektör özelliğini çıkarmak için moment değişmez (invariant) yöntemi kullanılmıştır. Veri kümesini sınıflandırmak için YSA kullanarak min-max normalizasyon önerilmiş olup bir gizli katman ile %93'lük bir doğruluk tespit edilmiştir. Fatmi ve arkadaşları (Fatmi vd., 2017)'deki çalışmada, 3 farklı işaretçi tarafından 13 ASL'ye ait sözcükleri Myo Armband adlı 2 adet giyilebilir kol bandı kullanarak konuşmaya çevirmek için etkili ve müdahalesiz bir çözüm önermiştir. Bu çalışmada kullanılan düzenli Myo Armband kollukları, eldiven tabanlı teknikler, kamera tabanlı sistemler ve 3D derinlik sensörlerinin kullanımını içeren mevcut çözümlerden çok daha pratiktir. Sınıflandırma yöntemi olarak gaussian karışım modeli - Gizli Markov Modeli (GMM-HMM) uygulayarak ASL sözcükleri (hareketler) için % 96 oranında başarımlı elde edilmiştir. Khamid ve arkadaşları işaret dili tanıma çalışmalarında çok modlu LMC ve Myo Armband birlikte kullanarak yeni bir yaklaşım getirmiştir. Çok modlu LMC ve Myo Armband kontrol cihazlarını kullanarak Endonezya işaret dili sistemi (ISLS) için hareket tanıma sistemi geliştirmişlerdir. ISLS, sözcükleri veya cümleleri temsil eden statik ve dinamik hareketlerden oluşur. Bununla birlikte, işaret dilini gerçekleştirmede bireysel varyasyonlar, özellikle otomatik çeviri geliştirmek için büyük bir zorluk oluşturmaktadır. İşaretleri tanımanın doğruluğu, hareketlerin varyasyonlarının artmasıyla doğrusal olarak azalacaktır. Burada, çok modlu yöntemleri uygulayarak bu sorunları çözmek hedeflenmiştir. Bu iki cihaz verisini birleştirilmesi ve Naive Bayes sınıflandırıcı kullanılarak, ISLS için hareket tanıma sisteminin doğruluğunun önemli ölçüde artırılacağı düşünülmüştür. El duruşlarından elde edilen veri akışları, oluşturulan veri kümesinin doğru bir şekilde senkronize edilmesini arttıracak olan zaman-alan serisi yöntemine dayanmaktadır. LMC için seçilen özellikler

parmak pozisyonlarına, açılara ve yüksekliklere dayanırken, Myo Arm kol bandı, bağlı parmağın ve önkol kaslarının aktiviteleriyle ilgili 8 EMG elektrodu kanalı tarafından üretilen elektrik sinyallerine dayanmaktadır. Çok modlu strateji için, her iki özellik veri kümesi birleştirilerek toplamda 55 özellik kullanılarak bir model oluşturulmuştur. Elde edilen sonuçlar, doğruluğu üzerinde belirgin bir iyileşme olduğunu, tek modlu LMC kullanarak % 91, çok modlu için % 98'e çıkmıştır. Çok modlu yöntemin karışıklık matrisi de tek modelden daha iyidir. Ek olarak, ISLS'nin hareket tanımlaması için çok modlu denetleyicilerin uygulanmasının, tek modlu LMC kullanmaya kıyasla daha iyi bir doğruluk ve başarımlar gösterdiğini belirlenmiştir (Khamid vd., 2017).

## 2.5. Kinect Tabanlı Çalışmalar

Porfirio ve arkadaşları (Porfirio vd., 2013)'deki çalışmada Kinect sensörleri ile elde edilen elin 3D kafes ve 2D gösterimi kullanılarak Brezilya işaret dili harflerinin tanınmasına için bir yöntem sundular. İşaret diline ait 61 farklı RGBD video görüntüsünde, eller bir ön bir yan çerçeveye bölütlenmiştir. Her çerçeve çifti elin 3D kafes karaltı şekil yöntemi kullanılarak inşa edilmiştir. Özelliklerin çıkarılmasında küresel harmonik yöntemine başvurulmuştur. Geliştirilen sistem, beş farklı işaretçi tarafından toplamda 610 Brezilya işaret dili harf video görüntüleri üzerinde test edilmiştir. Deney sonuçları en yüksek sınıflandırma oranının % 96 ile SVM ile elde edildiğini göstermiştir. Zhu ve arkadaşları (Zhu ve Wong, 2012)'deki çalışmada, Kinect cihazı kullanılarak, renk ve derinlik görüntülerinden bilgi birleştirilmesiyle tek çerçeveli el hareketlerinin tanınmasına yönelik çalışmıştır. Görüntü bilgilerinin çıkarımında en çok kullanılan SIFT ve LBP renk ve derinlik çekirdek tanımlayıcıları yoluyla renk ve derinlik bilgisi birleştirilmiştir. 5 farklı mesafeden 5 işaretçinin 24 statik ASL'ye ait toplamda 2460 parmak heceleme imge veri kümesi üzerinde başarımlar ölçümü yapılmıştır. Sonuçlar % 88 başarımlar oranı ile SVM sınıflandırıcısı kullanılarak elde edilmiştir. Kinect cihazı yardımıyla özellik çıkarımında LBP ve SVM sınıflandırıcısı kullanılan benzer çalışma da (Weerasekera vd., 2013) ise, ASL'ye ait alfabenin tanınmasına yönelik %90 oranında bir başarımlar sağlanmıştır. Rodríguez ve arkadaşı (Otiniano-Rodríguez ve Camara-Chavez, 2014)'deki çalışmada, Kinect sensörlerinden alınan RGB-D bilgileri kullanılarak parmak heceleme tanıma için karma bir sistem yaklaşımı hedeflemiştir. Bu karma sistem görüntü tabanlı kamera ve kızılötesi derinlik sensörleri birleşimidir. Görüntülerde, eller bölütlenerek ellere ait derinlik, şekil gibi görünüm özelliklerinden tanıma gerçekleştirilmiştir. RGB görüntü özellik çıkarımı için SIFT tanımlayıcısı, derinlik görüntü özellik haritası çıkarımı için gradyan çekirdek tanımlayıcısı kullanılmıştır. Beş farklı işaretçi tarafından 24 statik işaretin

toplamda Kinect sensörleri ile elde edilen 120000 ASL imgesi üzerinde test edilmiştir. Dene sonuçları % 91 oranı ile SVM sınıflandırıcı kullanılarak elde edilmiştir. Benzer bir çalışma (Rioux-Maldague ve Giguere, 2014), ASL'ye ait alfabenin tanınmasına yönelik Derin İnanç Ağı (DBN) sınıflandırma yöntemini tercih etmiştir. Bu kapsamda başka bir çalışmada (Sun C. vd., 2013) ise, Kinect sensörleri tarafından alınan renk, derinlik ve iskelet bilgileri kullanılarak ASL'ye ait görüntülerin otomatik tanınması sağlanmıştır. Bu sistemin yaklaşımı Gradyan Odaklı Histogram (HOG) tanımlayıcı tarafından özellik çıkartmaya dayanmaktadır. Dene sonuçları %86 oranı ile SVM ile elde edilmiştir. Sun ve arkadaşları, 2 boyutlu kameraya dayalı görüntü sistemleri dışında, derinlik bilgisi içeren Kinect cihazı kullanılmıştır. Görsel olarak farklı olan 73 ASL sözcüğü 9 denek tarafından 3 tekrar yapılarak veri kümesi oluşturulmuştur. Her biri bir dizi renkli görüntü, bir dizi derinlik haritası ve bir dizi iskelet bilgisi içeren toplam 1971 örnek toplanmıştır. Kinect cihazından belirlenen özellikleri alarak DEC yaklaşımı ile birlikte HOG özellik çıkarımı yöntemi kullanılarak %85 oranında başarımlı gözlenmiştir (Sun vd., 2013). Rakun ve arkadaşları (Rakun vd., 2013)'de; giriş sensörü olarak Microsoft Kinect kullanarak Endonezya işaret dili (SIBI) sözcüklerini tanımak için bir model önermiştir. Bu model, SIBI'den metne otomatik çevirinin bir parçasıdır. Her sözcüğün özellikleri, Kinect tarafından üretilen iskelet ve renk derinliği verilerinden çıkarılır. İskelet veri özellikleri, insan eklemleri ve Kartezyen eksenleri arasındaki açıyı gösterir. Çapraz Korelasyon (CC) işlemi ile Ayrık Kosinüs Dönüşümü (DCT) kullanılarak renkli görüntüler gri skalaya dönüştürülür ve öznelikleri çıkarılır. Veri kümesi 10 SIBI sözcüğün her biri için 12 örnek alınarak oluşturulmuştur En yüksek doğruluğun %96 Rastgele Orman (RF) tarafından elde edildiğini göstermiştir. Sombandith ve arkadaşları (Sombandith vd., 2017)'deki çalışmasında, bir Kinect algılayıcısı kullanarak Lao cümle işaret dilini tanımak için bir teknik sunmuştur. Kinect sensörü kullanılarak insan vücudunun özellik eklem pozisyonları ile birlikte bitişik eklem konumlarının her bir eklem vektörü çifti arasındaki eklem açılarını kullanmıştır. Kullanıcı, Lao işaret dilinin basit bir cümlesini temsil eden bir dizi duruş gerçekleştirdiğinde, sistem bir duruştan diğerine önemli ölçüde değişen eklem açılarını kaydeder. Lao işaret dilinin 10 basit cümlesi ile yapılan çalışmalardan yaklaşık %75 oranında bir tanıma başarımlı elde edilmiştir. Hisman ve arkadaşı (Hisham ve Hamouda, 2019)'da; Kinect V2'sini kullanarak ArSL işaretlerini tanıyabilen yeni bir model geliştirmiştir. Hem eller hem de vücut bölümleri tarafından gerçekleştirilen dinamik hareketler üzerinde çalışmış ve derinlik görüntüsünden el ve iskelet eklemlerini yakalamanın ve tespit etmenin etkili bir yolunu sunmuştur. İşitme engelli hastalar ve doktorları arasındaki iletişimi

kolaylaştırmak için 40 Arapça tıbbi işaretten oluşan ArSL veri kümesi modeli hem iskelet (32 öznitelik) hem de elin (30 öznitelik) özellik çıkarımları kullanılarak 3 farklı denek tarafından 1200 örneklemeden oluşturulmuştur. Sınıflandırıcıların doğruluğu KNN için %89 ve SVM için %79 olarak elde edilmiştir. Sutarman ve arkadaşları (Sutarman vd., 2015)'deki çalışmasında, dinamik Malezya işaret dilinin tanınması için Geri Yayılım Sinir Ağı (BNN) kullanarak 3 aşamalı bir yapı önermiştir. İskeletsel özellik çıkarımı için omurga ve baş ile ilgili 8 eklem pozisyonunun değeri iskelet veri takibi kullanılarak Kinect sensöründen alınan görüntülerden elde edilmiştir. Veri kümesi 5 farklı denek tarafından 5 tekrar yapılarak toplam 375 örnekten oluşmuştur. Sonuçlar, sistemin 15 dinamik Malezya işaret dilini %80 doğrulukla tanıyabildiğini göstermiştir. Khotimah ve arkadaşlarına ait çalışmada (Khotimah vd., 2017), Kinect kullanarak kullanıcı iskeleti yakalanarak ISL sözcük hareketlerini tanınması için bir yapı önermiştir. Bu iskeletlerden vektör değeri, açı değeri ve mesafe değeri hesaplanarak 9 iskelet kullanılarak toplam 28 öznitelik çıkarılmıştır. Sınıflandırıcı için karar ağacı ve Geri Yayılım Genetik Algoritma Sinir Ağı (BPGYSA) kombinasyonu uygulanmıştır. Veri kümesi, 8 Endonezya işaret dili (ISL) sözcüğü 2 kişi tarafından toplamda 240 örnekten oluşturulmuştur. Bu yapının tanıma doğruluğu, küçük bir sözcük dağarcığı ile değerlendirilmesine rağmen, %96 gibi bir oranda sonuç elde edilmiştir. Unutmaz ve arkadaşlarına ait çalışmada (Unutmaz vd., 2019), TİD işaretleri evrişimsel sinir ağları yardımıyla sözcüklere dönüştüren bir sistem önermiştir. Önerilen sistemde Microsoft'un Kinect cihazından alınan iskelet bilgisi kullanılmıştır. Kinect algılayıcısının karşısına geçen kullanıcı belirli bir süre içerisinde ilgili sözcüğe ait işaretleri yaptıktan sonra iskelet eklemleri bulunmuştur. Son olarak ardışık çerçevelerdeki iskelet noktaları birleştirilerek evrişimsel sinir ağları ile sözcük sınıflandırma yapılmıştır. Kinect cihazının sağladığı iskelet noktalarını kullanan bu sistemle 12 farklı denek üzerinden yakın ve uzakta olma durumu olmak üzere 9 farklı sözcük için veri kümesi üretilmiştir. Önerilen evrişimsel sinir ağının (CNN) Gauss SVM, doğrusal SVM, ağırlıklı KNN ve karar ağacı gibi sınıflandırıcılara göre daha iyi başarıma sahip olduğu görülmüştür.

## 2.6. LMC Tabanlı Çalışmalar

Literatürde işaret dili üzerine birçok çalışma mevcuttur. Bu çalışmalardan bu çalışmanın konusu ve kapsamı bağlamında LMC tabanlı çalışmalar statik parmak alfabesi ve dinamik sözcük tanıma olarak iki başlık altında incelenmiştir. Statik parmak alfabesi ve dinamik sözcük tanıma çalışmalarının özeti sırasıyla Tablo 2.1 ve Tablo 2.2'de sunulmuştur. Statik harf tanıma çalışmalarının literatür özeti, dinamik sözcük tanıma çalışmaları ile

karşılaştırma sunmak için hazırlanmıştır. Tablolarda ayırım sağlamak amacıyla farklı işaret dilleri için farklı renkler kullanılmıştır.

### **2.6.1. Statik parmak alfabeti tanıma çalışmaları**

Mohandes ve arkadaşlarının (Mohandes vd., 2014)'deki çalışmasında, LMC cihazı ile ArSLR tanıma sistemi için yeni bir yaklaşım önermiştir. Çalışmada işaretlerle ilgili 12 özellik seçilmiştir. Belirlenen özellik çıkarımları ile birlikte bir denekten her bir harf için 10 çerçeveli (frame) 10 tekrar içeren toplam 2800 örnekli bir veri kümesi oluşturulmuştur. 28 adet işaret üzerinde yapılan çalışmada Naive Bayes (NB) sınıflandırıcı ile %98, çok katmanlı algılayıcı (MLP) sınıflandırıcı ile % 99 doğruluk oranı yakalanmıştır. İkinci çalışmada (Mohandes vd., 2015) ise, çift LMC cihazı kullanarak ArSLR tanıma sistemi için yeni bir yaklaşım önermişlerdir. Bu yaklaşımda, bir parmağın başka bir parmak veya el tarafından perdelenmesini önlemek için çift LMC cihazı tercih edilmiştir. Bu çalışmada da sistem için işaretlerle ilgili 12 özellik seçilmiştir. Belirlenen özellikler ile birlikte bir denekten her bir harf için 10 çerçeveli (frame) 10 tekrar içeren toplam 2800 örnekli bir veri kümesi oluşturulmuştur. 28 adet alfabe üzerinde yapılan çalışmada LDA sınıflandırıcı ile %97 doğruluk oranı yakalanmıştır. Mapari ve arkadaşı (Mapari ve Kharat, 2015)'de; Amerikan, Hint ve Çin işaret dillerinin tanınması üzerinde durmuştur. Sınıflandırma yöntemi olarak MLP, GFF ve SVM olmak üzere 3 farklı sinir ağı analizi sunmuşlardır. İşaretler için 18 konum değeri, 15 uzaklık değeri ve 15 açı değeri olmak üzere toplam 48 adet özellik belirlenmiştir. Bu çoklu işaret dili tanınmasına yönelik kurdukları sistemin başarı oranları %90 üzerinde çıkmıştır. Mapari ve arkadaşı benzer (Mapari ve Kharat, 2017)'deki çalışmasında; her iki eli kullanarak Hint işaret dili (ISL) alfabetinin tanınması üzerinde durmuştur. Sınıflandırma yöntemi olarak öklid uzaklık ve kosinüs benzerliği yöntemlerini sunmuşlardır. 10 farklı işaretçi tarafından yapılan 31 Hint alfabeti için 48 özellik kullanılmıştır. İşaret dili tanınmasına yönelik kurdukları iki sistemin başarı oranları sırasıyla %88 ve %90 üzerinde çıkmıştır. Naglot ve arkadaşı (Naglot ve Kulkarni, 2016)'da; farklı işaretleri tanımak için çok katmanlı algılayıcı (MLP) sinir ağı kullanılarak 26 Amerikan alfabetine ait veri kümesi üzerinde gerçek zamanlı tanıma sistemi önermiştir. Önerilen sistemin her alfabeden 20 tekrar içeren 520 örnekli veri kümesinin tanıma başarı oranı %96'dır. Chuan ve arkadaşları (Chuan vd., 2014)'de, LMC kullanarak ASL'e ait hareketleri tanıma sistemi geliştirmiştir. Bu sistemde ASL'ye ait 26 İngiliz alfabetini sensörlerden elde edilen verilere ait özellikler ile KNN ve SVM sınıflandırıcı yöntemleri kullanarak tanıma yapılmıştır. İki işaretçi tarafından toplanan 4 veri kümesi 7900'ün üzerinde örnekten

oluşmaktadır. Yapılan deney sonuçları en yüksek sınıflandırma oranının %72 oranı ile KNN ve %79 oranı ile SVM ile elde edildiğini göstermiştir. Chopuk ve arkadaşları (Chopuk vd., 2018)'de, LMC kullanarak ASL'deki 7 yumruk işaretli alfabenin tanınmasını 4 denek tarafından karar ağaçları yöntemi ile sınıflandırma yaparak %96 başarımla elde etmişlerdir. Eqab ve arkadaşları (Eqab ve Shanableh, 2017)'deki çalışmasında, gerçek zamanlı ArSLR alfabeti tanıma için bir mobil uygulama geliştirmiştir. Sistem, izole edilmiş işaret dili sözcükleri üzerinde çalışacak şekilde tasarlanmıştır. Ortalama, varyans ve kovaryans gibi istatistiksel özelliklere dayalı basit özellik çıkarma yaklaşımları kullanılmıştır. Ayrıca, minimum mesafe sınıflandırıcıya dayalı basit sınıflandırma yaklaşımı kullanılmıştır. 15 farklı işaret dili sözcüğü kullanılarak yapılan sınıflandırma sonuçları, yüksek oranda sınıflandırma sonuçları elde edildiğini ortaya koymuştur. Deneysel sonuçlar 15 işaret dili sözcük için %98 üzeri bir başarımla sonuç elde edildiğini ortaya koymuştur. Wibowo ve arkadaşları (Wibowo vd., 2017)'de; LMC denetleyicisi ve makine öğrenimi sınıflandırıcısı NB ile birlikte 24 ISL alfabeti kullanmıştır. Sağ elini kullanan 3 farklı işaretçiden her bir harf için 30 tekrardan oluşan toplam 720 örnek veri kümesi oluşturulmuştur. Kullanılan SIBI harflerinin statik olduğu ve tek elle yapıldığı göz önüne alınarak doğruluk oranının %95'e ulaştığı görülmüştür. Anwar ve arkadaşları, SIBI alfabeti tanıma sistemi önermiştir. Bu sistem, LMC cihazı ve özellik çıkarım yöntemi kullanır. Bu önerilen sistem, k-NN sınıflandırıcı kullanılarak %95 ve SVM sınıflandırıcı kullanılarak %93 tanıma doğruluk oranı vermiştir. SIBI'de 26 alfabe işareti 5 farklı kişi tarafından 10 tekrar yapılarak veri kümesi oluşturulmuştur (Anwar vd.,2017). İnsani ve arkadaşları, oda ışıklarının çeşitli aydınlatma koşullarını karşılaştırarak SIBI alfabeti tanımayı amaçlamıştır. Birkaç farklı oda aydınlatma koşullarında el hareketi özelliklerinin çıkarılması aşamasında LMC ve 24 SIBI alfabetinin sınıflandırılması için SVM kullanılmıştır. 25W beyaz ışık altında %96 doğrulukla en yüksek başarımla elde edilmiştir (Insan vd., 2019). Endonezya, işitme engelli kişi sayısında dünyada ilk dört arasında bulunmaktadır. Khotimah ve arkadaşları (Khotimah vd., 2017)'de; tek el ile yapılan 26 SIBI alfabeti tanıma odaklanmıştır. Ayrıca, bu işaret dillerini tanımak için yeni bir yöntem olan kural tabanlı geri yayılım genetik algoritma sinir ağı (RB-BPGYSA) kullanılmıştır. Bu çalışmada, 10 ardışık çerçeveden (frame) bir el hareketi elde edilmiştir. Bu çalışmada her çerçevenin 34 özelliği çıkarılmıştır. Bu çerçevelerden gelen özelliklerin ortalaması alınmış ve bir örnek verinin özellikleri olarak kullanılmıştır. Bu veri kümesi bir kullanıcıdan 5 tekrarlı 260 örnekten oluşmuştur. Bu uygulamada %93 başarımla doğruluk gözlenmiştir. Naidu ve arkadaşları, Hint işaret dili (ISL) işaretlerinin tanıması için LMC cihazı kullanarak bir

sistem önermiştir. 26 harf,10 rakam ve 9 statik sözcükten oluşan 45 Hint işaret dili (ISL) alfabesi 10 farklı işaretçi ile test edilmiştir. Bu çalışmada, eğitim ve sınıflandırma aşamasında öklid mesafe, kosinüs, zar ve jaccard benzerlik ölçütü kullanılmıştır. Bu benzerlik ölçütlerinden kosinüs benzerlik ölçütü ile %90ın üzerinde tanınma başarımı elde edildiği gösterilmiştir (Naidu ve Ghotkar, 2016). Literatür incelemesine göre, LMC tabanlı statik TİD tanıma çalışmaları çok azdır. Demircioğlu ve arkadaşları LMC ile TİD'e ait seçilen temel el hareketlerinin tanınmasına yönelik bir çalışma yapmışlardır. Çalışmanın verisi, statik ve tek el ile yapılan 18 el işaretinden 5'er örnek, her bir örnek için 30 çerçeve içeren 139 özelliğe sahip 2700 kayıttan oluşmaktadır. Elde edilen bu veriler kaydedilerek çeşitli makine öğrenmesi yöntemleri ile test edilmiş ve elde edilen sonuçlar karşılaştırılmıştır (Demircioğlu vd., 2016), (Demircioğlu vd., 2016). Karacı ve arkadaşları ise LMC cihazından TİDde 5 tek el işareti ve 5 çift el işaretini 2 farklı kişiden alarak veri kümesi oluşturmuşlardır. 120 özellikli veri kümesi için en başarılı sınıflandırıcılar karar ağacı ve derin öğrenme modelleridir. 390 özelliğe sahip veri kümesi için ise derin öğrenme modeli olmuştur (Karacı vd., 2018). Katılmış ve Karakuzu çift elli statik TİD alfabesi tanıma sistemi önermişlerdir. Çalışmada 17 harf için 2 farklı kişiden 50 tekrar ile 3672 örnekten oluşan veri kümesi oluşturulmuştur. Ayrıca PCA, LDA ve PCA+LDA yöntemleri uygulanarak 3 adet boyutu indirgenmiş veri kümesi oluşturulmuştur. Geleneksel makine öğrenmesi yöntemleri ile test edilerek sonuçlar karşılaştırılmıştır. Orijinal veri kümesi için en yüksek başarı oranı %98,5 ile RF ile elde edilmiştir. Boyutu indirgenen veri kümeleri arasında LDA için en yüksek başarı oranı %96,4 ile RF ile elde edilmiştir (Katılmış ve Karakuzu, 2020). Enikeev ve arkadaşı, derin evrişimli sinir ağı kullanarak işaret dilini tanıma ve sınıflandırma yeteneğine sahip bir sistem sunmaktadır.10 el hareketini 10 farklı denek ile ortalama %95'lik bir başarıyla sınıflandırdığı sonucuna varmışlardır (Enikeev ve Mustafina, 2020). Benzer bir çalışmada, işaret dilinin ayrı bir bileşeni olan parmakla hecelemeye odaklanıyoruz. Sunulan sistem, LMC ile elde edilen statik Rusça parmak alfabesi hareketlerine odaklanılmıştır.Derin öğrenme işaret tanıma sisteminin doğruluğunu değerlendirmek için 400 statik işaret görüntüsünden oluşan yeni bir veri kümesi oluşturulmuştur.Sistem %97 başarı oranına ulaştığı tespit edilmiştir (Enikeev ve Mustafina, 2021). Hein ve arkadaşlarının (Hein vd., 2021) 'de hareket tabanlı işaret dili tanıma sistemi üzerine makine öğrenmesi algoritmaları ile özellik çıkarma yöntemi ve tanıma önermiştir. Bu önerilen model Myanmar ulusal işaret diline dayanmaktadır.35 denek kullanılarak oluşturulan veri veri kümesi setleri içerisinde en iyi başarı oranı %94 ile elde edilmiştir.

## 2.6.2. Dinamik sözcük tanıma çalışmaları

Aliyu ve arkadaşları (Aliyu vd., 2017)'de; parmak tıkanmaları ve eksik verilerle ilgili zorlukları aşmak için çift LMC tabanlı ArSLR tanıma sistemi önermiştir. Özellik çıkarımı için, her iki cihazdan 17 adet geometrik özellikler seçilirken, sınıflandırma için Bayes yaklaşımlı Gauss karışım modeli (GMM) ve basit LDA yaklaşımı kullanmışlardır. Veriler, izole edilmiş çift eli 100 Arap dinamik işaret sözcüğü için bir yetişkin işaretçiden toplanarak %94 oranında bir tanıma başarımı elde edilmiştir. Aynı yazarların ikinci çalışmasında (Deriche vd., 2019) ise 2 yetişkin işaretçi kullanılarak %92 başarımla elde edilmiştir. Pramunanto ve arkadaşları (Pramunanto vd., 2017)'de; SIBI'de 5 statik ve 5 dinamik tek elle gerçekleştirilen el hareketi işaretlerini inceleyerek 19 özellik belirlemiştir. Çalışmada veri kümesi bir denek tarafından her bir işaret için 25 tekrar yapılarak oluşturulmuştur. Naive Bayes (NB) yöntemi kullanılarak yapılan sınıflandırmada %80 doğruluk elde edilmiştir. Avola ve arkadaşlarının (Avola vd., 2019)'daki çalışmasında, dinamik el hareketleri zaman içinde değişen özellik vektör kümeleriyle temsil edilmiştir. Tekrarlayan sinir ağları (RNN), insan elinin parmak kemiklerinin oluşturduğu açılar özellik olarak alınarak eğitilmiştir. Veri kümesi 12 dinamik ve 12 statik ASL sözcüğü içerir. Veri kümesi 20 farklı kişiden alınan 1200 el hareketi dizisinden oluşmaktadır. Her bir işaret 15 erkek ve 5 kadın tarafından yapılmıştır. Önerilen bu yöntemde %96'nın üzerinde bir başarımla elde edilmiştir. Mittal ve arkadaşları (Mittal vd., 2019)'da; ardışık hareket dizileri için LSTM modeli ve ardışık bağlı hareket dizilerini tanıyan işaret tanıma sistemi (SLR) önermiştir. Sistem, sürekli işaretlerin alt birimlere bölünmesi ve sinir ağları ile modellenmesi üzerine kurulmuştur. Kullanılan veri kümesi 6 işaretçiye her işaret 15 kez tekrar ettirilerek oluşturulmuştur. Veri kümesi için 36 dinamik özellik çıkarılmıştır. Önerilen sistem, Hint işaret diline (ISL) ait çift eli izole 35 farklı işaret sözcüğü ve 942 işaret cümlesi kullanılarak test edilmiştir. Sırasıyla %72 ve %89 ortalama doğruluk elde edilmiştir. Kumar ve arkadaşlarına ait çalışmada (Kumar vd., 2017), birleşik gizli Markov modeli (CHMM) kullanarak işaret dili tanıma (SLR) için yeni bir birleşik çoklu sensör yapı önerilmiştir. Bu yapı, işitme engelli kişiler tarafından gerçekleştirilen dinamik izole işaret hareketlerini tanımak için kullanılmıştır. En iyi tanıma doğruluğu, CHMM ile %90 gibi yüksek bir oranda elde edilmiştir. Hint işaret diline (ISL) ait 25 dinamik tek el ile yapılan işaret sözcüğü için 10 farklı işaretçi tarafından 8 kez tekrar yapılarak veri oluşturulmuştur. Bu çalışmada, LMC ve Kinect cihazları işaret hareketini yakalamak için paralel olarak kullanılmıştır. Veri kümesi için 66 özellik belirlenmiştir. Kumar ve arkadaşları benzer çalışmada (Kumar vd., 2017), sensör cihazları kullanarak izole edilmiş

işaret dili tanıma (SLR) için yeni çoklu model yapı önermişlerdir. Kinect ve LMC cihazları, hareket sırasında iki farklı görünümünden parmak ve avuç içi pozisyonlarını yakalamak için oluşturulan çoklu model yapıda kullanılmıştır. Her iki sensörle yakalanan ham verilerden bir dizi özellik çıkarılır. Tanıma, gizli Markov modeli (HMM) ve çift yönlü uzun kısa süreli hafızalı sinir ağı (BLSTM-NN) tabanlı sıralı sınıflandırıcılar tarafından ayrı ayrı gerçekleştirilmiştir. Yapı, 50 farklı işaret sözcüğü içeren 7500 Hint işaret dili (ISL) hareketinden oluşan bir veri kümesi üzerinde test edilmiştir. Tek sensör tabanlı tanıma ile karşılaştırıldığında her iki sensörden gelen veriler birleştirilirse doğrulukların iyileştiği görülmüştür. Başarım oranları HMM ve BLSTM-NN için sırasıyla %97 ve %94 olarak tespit edilmiştir. Hisham ve arkadaşı (Hisham ve Hamouda, 2017)'de; işaretlerin gerçekleştirilmesinde statik ve dinamik hareketleri tanımak için bir model önermişlerdir. Önerilen model, aralarında ortak özelliklere sahip 2 farklı avuç içi ve kemik özellik seti kümesine dayanmaktadır. Çalışma ArSL'e ait 38 statik hareket ve 20 dinamik hareketi içermektedir. Statik ve dinamik hareketler için KNN modeli ile sırasıyla %99 ve %98 başarı oranı; DTW modeli ile sırasıyla %97 ve %96 başarı oranının yakalandığı rapor edilmiştir. Katılmış ve Karakuzu çift eli dinamik TİD sözcük tanıma sistemi önermişlerdir. Çalışmada 50 dinamik sözcük için 4 farklı denekten 40 tekrar ile 8000 örnekten oluşan veri kümesi oluşturulmuştur. Özellik seçimi algoritması kullanılarak kademeli olarak özellik sayıları düşürülmüştür. Sonrasında PCA, LDA ve PCA+LDA yöntemleri uygulanarak toplamda 12 adet boyutu indirgenmiş veri kümesi oluşturulmuştur. Aşırı öğrenme (ELM) tabanlı sınıflandırıcılar ile test edilerek sonuçlar karşılaştırılmıştır. Orijinal ve boyutu indirgenen veri kümeleri için ML-KELM sınıflandırıcının başarım oranlarını koruduğunu ve en iyi başarım oranlarını verdiği gözlemlenmiştir (Katılmış ve Karakuzu, 2021). Wang ve arkadaşı (Wang ve Zhang, 2021 )'de, çok parçalı iskelet verileriyle temsil edilen ASL'yi metne çevirmek için yeni bir sürekli işaret dili tanıma çerçevesi sunmuştur. Çerçevede, çok akışlı uzamsal-zamansal grafik evrişim ağları (Multi-Stream ST-GCN), grafik yapısı ile dinamik çok parçalı iskeletleri modellemek için kullanılmıştır.

Tablo 2.1 ve Tablo 2.2'de verilen ve yukarıda kısaca özetlenen çalışmalardan da anlaşılacağı üzere bu çalışma farklı bir bakış açısı ve çözüm yöntemi önermektedir. Kısaca, bu çalışmanın katkıları şunlardır:

- a. İşaret dili tanıma için bir sınıflandırma yöntemi olarak ELM öğreniminin kullanılmasına yönelik ilk uygulama olması
- b. İşaret dili tanıma için yeni bir özellik seçim algoritması yaklaşımının kullanılması.

- c. TİD’de histogram ve zamansal özellik çıkarımı yöntemi yaklaşımının kullanılması
- d. TİD’de dinamik sözcük tanıma konusunda bu bağlamda ilk çalışma olması.
- e. TİD’de LMC+Kinect cihazlarının kullanılması bu bağlamda ilk çalışma olması
- f. TİD’de dinamik sözcük tanıma için kapsamlı bir veri kümesi oluşturma
- g. Üç farklı boyut indirgeme yönteminin özellik çıkarma ve seçme yaklaşımlarının bütünleştirilmesiyle oluşturulan 9 farklı veri kümesi ile tanıma başarımlarının analizinin sunulması.

**Tablo 2.1.** Statik işaret tanıma literatür özeti

Kay.	Yaklaşım (Özellik çıkarma, Özellik sayısı ve Dil)		El	Veri kümesi, Sınıflandırıcı, %Başarımlar			
Mohandes vd., 2014	3D parmak uzunluk, genişlik, uç ve avuç içi konum, el pitch, roll ve yaw bilgisi	12	ArSLR	Tek el	28 alfabe, 1 denek, 10 tekrar, 10 çerçeve	MLP NBC	99 98
Mohandes vd., 2015	3D parmak uzunluk, genişlik, uç ve avuç içi konum, el pitch, roll ve yaw bilgisi	12	ArSLR	Tek el	28 alfabe, 1 denek, 10 tekrar, 10 çerçeve	LDA	97
Mapari ve Kharat, 2015	Konum, mesafe ve açı değerleri	48	ASL ISL CSL	Tek el tek el iki el	32 alfabe, 146 denek, 1 tekrar 34 alfabe, 100 denek, 1 tekrar 33 alfabe, 10 denek, 10 tekrar	MLP SVM GFF	90 93 96
Mapari ve Kharat, 2017	Konum, mesafe ve açı değerleri	48	ISL	Tek el	31 alfabe, 10 denek, 1 tekrar	ED CS	88 90
Naglot ve Kulkarni, 2016	Parmak ucu konumundan avuç içi konumuna, her parmağın parmak ucu konumu	9	ASL	Tek el	26 alfabe, 4 denek, 5 tekrar	MLP	96
Chuan vd., 2014	Mesafe, yayılma ve üçlü yayılma ortalamaları	35	ASL	Tek el	26 alfabe, 2 denek, 4 veri kümesi	KNN SVM	72 79
Chophuk vd., 2018	3D parmak ucu ve 3D avuç içi konumu	18	ASL	Tek el	7 alfabe, 4 denek, 30 örnek	DT	96
Eqab ve Shanableh, 2017	3D el yönleri, 3D Parmak ucu pozisyonları, el kavrama gücü	19	ArSLR	Tek el	15 sözcük, 1 denek, 5 tekrar	MDC	98
Wibowo vd., 2017			ISL	Tek el	24 alfabe, 3 denek, 10 tekrar	NBC	95
Anwar vd., 2017	3D parmak ucu ve 3D avuç içi konumu	11	ISL	Tek el	26 alfabe, 5 denek, 10 tekrar, 10 çerçeve	KNN SVM	95 93
Insan vd., 2019	Konumsal ve mesafe değerleri	17	ISL	Tek el	24 alfabe, 1 denek, 40 tekrar	SVM	96
Khotimah vd., 2017	Konumsal ve mesafe değerleri	34	ISL	Tek el	26 alfabe, 1 denek, 5 tekrar, 10 çerçeve	BPGA-NN	93
Naidu ve Ghotkar, 2016	3D parmak ucu ve 3D avuç içi konumu	8 16	ISL	Tek el çift el	45 işaret, 10 denek, 1 tekrar	ED	90
Demircioğlu vd., 2016	Parmak durumu, avuç içi ve parmak yönü, parmak mesafesi, parmak ucu pozisyonu, parmak yönü, kemik açıları	139	TSL	Tek el	18 işaret, 1 denek, 5 örnek, 30 çerçeve	MLP RF	96 93
Demircioğlu vd., 2016	Parmak durumu, avuç içi ve parmak yönü, parmak mesafesi, parmak ucu pozisyonu, parmak yönü, kemik açıları	139	TSL	Tek el	18 işaret, 1 denek, 10 örnek, 30 çerçeve	RF	93
Karacı vd., 2018	Kemik yönleri, avuç içi pozisyonu, kol yönü, bilek pozisyonu, dirsek pozisyonu	120 390	TSL	Tek el çift el	5 alfabe, 2 denek, 5 tekrar 5 alfabe, 2 denek, 5 tekrar	DNN YSA DT	100 93 98
Katılmış ve Karakuzu, 2020	Parmak durumu, konum mesafesi, yön açıları, 3D avuç içi, el, kol, bilek ve dirsek konumu	119	TSL	Çift el	17 alfabe, 2 denek, 50 tekrar	SVM RF LDA	95 98 93

**Tablo 2.2.** Dinamik işaret tanıma literatür özeti

sn	Cihaz	Yaklaşım (Özellik çıkarma, Özellik sayısı ve Dil)		El	Veri kümesi, Sınıflandırıcı, %Başarım			
Aliyu vd., 2017	Çift LMC	3D parmak ucu konumu, 3D avuç içi konumu, parmak mesafesi, el pitch, roll ve yaw bilgisi, el küre yarıçapı	16	ArSLR	Çift el	100 sözcük, 1 denek, 10 tekrar	LDA GMM	94
Deriche vd., 2019	Çift LMC	3D parmak ucu konumu, 3D avuç içi konumu, parmak mesafesi, el pitch, roll ve yaw bilgisi, el küre yarıçapı	16	ArSLR	Çift el	100 sözcük, 2 denek, 10 tekrar	LDA GMM	92
Pramunanto vd., 2017	LMC	Parmak ve avuç açısı, avuç içi mesafesi, avuç içi yükseklikleri	19	SIBI	Tek el	5 sözcük, 1 denek, 25 tekrar	NBC	80
Avola vd., 2019	LMC	3D parmak ucu konumları, 3D avuç içi konumu, iç açılar, parmak içi açılar	23	ASL	Tek el çift el	12 sözcük, 20 denek, 2 tekrar	RNN LSTM	96
Mittal vd., 2019	LMC	3D parmak ucu ve 3D avuç içi konumu	36	ISL	Çift el	35 sözcük, 6 denek, 15 tekrar	LSTM	89
Kumar vd., 2017	LMC Kinect	3D parmak ucu konumu, 3D parmak ucu yönü	66	ISL	Tek el	25 sözcük, 10 denek, 8 tekrar	CHMM	90
Kumar vd., 2017	LMC Kinect	3D parmak ucu konumu, 3D parmak ucu yönü	132	ISL	Tek el çift el	50 sözcük, 10 denek, 15 tekrar	HMM BLSTM-NN	97 94
Hisham ve Hamouda, 2017	LMC	3D yön ve konum özellikleri, parmak açıları, her parmak kemiğinin son konumu ve avuç içi konum vektörlerinin çıkarma	170	ArSLR	Tek el çift el	16 sözcük, 3 denek, 25 tekrar	DTW SVM KNN YSA	97
		3D yön ve konum özellikleri, parmak açıları, parmak kemiğinin başlangıç konumunu parmak kemiğinin bitiş konumu vektörlerinden çıkarma	140					96
Katılmış ve Karakuzu, 2021	LMC	Parmak durumu, konum mesafesi, yön açıları, 3D avuç içi, el, kol, bilek ve dirsek konumu	119	TSL	Çift el	50 sözcük, 4 denek, 40 tekrar	ELM ML-KELM	96 99

## 2.7. Türk İşaret Dili Üzerine Mevcut Çalışmalar

Selda ve arkadaşlarının (Bayrak ve Nabyev, 2006)'daki çalışmasında, TİD sesli harflerin tanınması üzerinde durulmuştur. Görüntü işleme modülü yardımıyla görüntüdeki gereksiz bilgilerin çıkarılması ve YSA girişleri için, görüntü 8 bitlik gri yapıya dönüştürülmüş sonrasında ise ikili görüntü için histogram çıkarılması işlemleri yapılmıştır. Probleme YSA ile çözüm getirilmeye çalışılarak, %100'e yakın başarımlar elde edilmiştir. Aran'a ait çalışmada (Aran, 2008) işaret dili örneklerinin video tabanlı olarak tanınması problemi ele alınmıştır. İşaret tanıma probleminde yoğunlukla kullanılan üretici ve ayırıcı modellerin sınıflandırma yaklaşımları HMM'den türetilen fisher çekirdekleri kullanılarak birleştirilmiştir. İşaretlerin bu çok kipli yapısı ardışık tümleştirme yöntemi kullanılarak tanıma sistemi geliştirilmiştir. Haberdar (Haberdar, 2005) yapmış olduğu çalışmada kameradan alınan görüntüden gerçek zamanlı TİD'e ait işaretleri tanıma sistemi geliştirmiştir. Görüntüyü oluşturan çerçevelerden çıkarılan özellikler ile HMM ve KNN sınıflandırıcı kullanarak bir tanıma sistemi

hedeflenmiştir. TİD konusunda ilk olması ve elde edilen sonuçlarının %93 ve % 91 oranlarda tanıma başarımı dikkat çekicidir. TİD hareketlerine ait bir görüntü arşivi olmadığı için, oluşturulan tanıma sisteminin kullanıcıya bağımlı olduğu belirtilmiştir. Bayrak (Bayrak S. , 2009) çalışmasında, TİD sözcüklerini içeren video görüntüsünden hareketlerin tanınmasını yönelik çalışmıştır. Görüntüden seçilmiş el hareketinin takibi ve kullanılacak özelliklerin çıkarımı için *zernike moment* yöntemi kullanılmıştır. İşaretlerin yorumlanması aşamasında HMM yöntemi kullanılmıştır. TİD’de parmak heceleme yoluyla ifade edilen harflerin tanınması amacıyla yapılmış çalışmalar da mevcuttur. Altun ve Albayrak tarafından sunulan yaklaşım (Altun ve Albayrak, 2011) bu hedefe yöneliktir. DOG ilgi bölgeleri ve SIFT yerel tanımlayıcılar ile geliştirilmiş Hough dönüşüm yaklaşımı kullanılarak TİD’e ait durağan hareketleri tanıyan bir bilgisayar görme sistemi geliştirilmiştir. Geliştirilmiş Hough dönüşüm tekniği yaygın olarak resimde parametrik çizgi ve daire şekilleri çıkarmak için kullanılır. TİD’e ait 29 karakterin her biri için 16 video örnek veri kümesi kullanılmıştır.3 farklı arka plan kullanılarak bir veri kümesi üzerinde test edilerek %93 başarımla elde edilmiştir. Benzer bir çalışmada (Altun O. , 2010), TİD alfabesi harflerinin tanınması için her bir videodan çıkartılan anahtar çerçeveler üzerinden SIFT noktalarının oluşturduğu yıldız – meteor – yay sistemleri yardımı ile sınıflandırma yapılmıştır. En başarılı sınıflayıcıların KNN ve SVM olduğu bildirilmiştir. İlgi bölgeleri ve yerel tanımlayıcılar ile geliştirilmiş Hough dönüşüm yaklaşımı kullanarak bir tanıma sistemi hedeflenmiştir. Dört farklı işaretçi tarafından icra edilmiş toplamda 116 TİD harfi videoları üzerinde %94 başarımla gözlemlenmiştir. Göknaar ve Yıldırım (Göknaar ve Yıldırım, 2005), çalışmalarında kamera sistemi ile el işaretlerinin algılanması için çıkarılan şekil geometri özellikleri ile çok katmanlı algılayıcı (MLP) ve radyal temelli fonksiyon ağırları (RBF) sınıflandırıcı olarak kullanmıştır. Eğitim ve test için toplamda 600 adet resim üzerinde başarımla ölçümü yapılmış ve %80 oranlarında doğruluk gözlemlenmiştir. Bu sistem başarımla seviyesi genelde kullanılan HMM sınıflandırıcılara göre daha düşük olduğu bildirilmiştir. Ayrıca Işıkdöğün ve Albayrak tarafından yapılan bir çalışmada (Işıkdöğün ve Albayrak, 2011) ise parmak harf görüntülerinin otomatik tanınması sağlanmıştır. Bu çalışmada işaretlerin anlam ayırımına yardımcı en etkili özelliklerin çıkarılmasına dayalı bir TİD’e ait parmak heceleme tanıma sistemi hedeflenmiştir. Bu sistemin temel yaklaşımı temel bileşen analizi (PCA) tekniği tarafından boyut azaltma ve gradyan odaklı histogram (HOG) tanımlayıcılar tarafından özellik çıkartmaya dayanmaktadır. Test veri kümesi 4 farklı işaretçi ve 493 parmak heceleme görüntüsü üzerinde başarımla ölçümü yapılmış ve %99 oranında başarımlar elde edilmiştir.

Yakut (Yakut, 2013)'deki çalışmasında, işaret dili harflerinin tanınması için iki görüntü işleme yöntemi kullanmıştır. Yöntem açısız tarama ve ağırlıklandırılmış matris tabanlı yöntemleri kullanılarak şekillerin kenar takibine dayanmaktadır. Çalışma genel kabul görmüş 10 harf için toplamda 240 test veri kümesi imgesinde % 82 oranında başarı sağlamıştır. Büyüksaraç, her bir el için toplam 23 özellik vektörü çıkararak ayrık HMM yöntemi ile tanıma gerçekleştirmiştir. 10 TİD sözcüğü için kullanıcı bağımsız durumda %85, bağımlı durumda %100 doğruluk oranı ile sistemin çalıştığı görülmüştür (Büyüksaraç, 2015). Demirler, Sınır bölgeleri bulunan görüntülerin Hu moment öznitelikleri çıkararak, temel bileşen analizi (PCA) işlemi uygulayarak gerçek zamanlı bir sistem geliştirmiştir. Hu momentlerinin sayısı yediye tanıtma başarımlı oranı %76, PCA yardımıyla üçe düşürülmüş Hu momentler ile %95 başarımlı oranına yükselmiştir (Demirler, 2015). Gökmar, basit bir kamera kullanarak el işaretlerinin bilgisayar tarafından tanınmasını sağlamıştır. Kamera ile elin görüntüsü alınarak çeşitli görüntü işleme teknikleri yardımıyla geometrik öznitelik vektörleri çıkarılarak MLP sınıflandırıcı kullanılmıştır (Gökmar G. , 2015). Kındırođlu, Türk, Çek ve Rus el alfabelerinde yarı gerçek zamanlı el alfabeleri tanıma sistemi geliştirmiştir. El hareketi temsil yöntemlerinden eliptik Fourier betimleyicileri, Hu momentleri, ışınsal uzaklık fonksiyonu, yerel ikili örüntüler gibi öznitelikler kullanılmıştır. İzole el hareketleri için en iyi tanıma oranı %92 ile KNN sınıflandırıcı kullanılarak yerel ikili örüntü betimleyici vermiştir (Kındırođlu, 2011). Memiş (Memiş ve Albayrak, 2013)'deki çalışmasında, video görüntülerinden alınan TİD'e ait durađan olmayan hareketlerin tanınmasına yönelik çalışmıştır. Kinect algılayıcının derinlik algılayabilme özelliđi de kullanılarak TİD'e ait veri kümesinde RGB-D video görüntüleri tanıma başarımlı oranını arttırmada etkin olarak kullanılmıştır. Ardışıl görüntü hareket farklarına dayalı toplamsal imge yaklaşımı ve ayrık kosinüs dönüşüm (DCT) yöntemi kullanılarak uzam-zamansal öznitelikleri çıkaran bu sistemle işaret diline ait görüntüler tanınabilmiştir. Yaklaşımın başarımlı oranını sınamak için Manhattan uzaklığını kullanan KNN sınıflandırıcısı TİD veri kümesinde %90 oranında başarımlı elde edilmiştir. Memiş benzer bir başka çalışmada (Memiş A. , 2013) ise, TİD'e ait 111 sözcük, ASL'ye ait 20 sözcükten oluşan durađan olmayan bir görüntü veri kümesi için Kinect algılayıcı kullanılarak %90 üzerinde bir başarımlı oranı ile tanıma sağlanmıştır. Işıklıgil, 13 farklı işaretçiden elde edilen 40 farklı veri kümesini Kinect duyargası yardımıyla yapılan işaretçi-bağımsız deneylerde %59 ve işaretçi-bağımlı deneylerde %91 tanıma başarımlı oranlarına ulaşmıştır (Işıklıgil, 2014). Kakayev, 3 işaretçi tarafından TİD'e ait 15 sözcük Kinect kamerası ile toplamda 180 kayıt oluşturularak tanıma sistemi geliştirmiştir. Sistemi eğitmek ve test etmek

için HMM sınıflandırıcı kullanılarak %100 yakın bir başarı oranı sağlanmıştır (Kakayev, 2016). Ketenci ve arkadaşları (Ketenci vd ., 2016)'da, yüzey elektrotları kullanılarak ön koldan toplanan EMG sinyalleriyle TİD'deki rakamları tanımaya çalışmışlardır. Dört kanaldan alınan veriler ön işlemden geçirilerek özellik çıkarımı sağlanmıştır. İşaretin etkin değeri, standart sapması, dalga form uzunluğu, Fourier dönüşümü katsayıları ve çapraz ilinti fonksiyonunun standart sapması özellikleri kullanılmış ve ayrık doğrusal analiz sınıflandırıcı ile hareket tanıma başarımının arttığı görülmüştür. Demirel, ön kol kas aktivitelerini ve hareketlerini ölçmek için Myo Arm kol bandı cihazı kullanarak genel bir işaret tanıma sistemi geliştirilmesi üzerinde çalışmıştır.10 test kullanıcısı tarafından 13 farklı müzik hareketini YSA kullanarak ortalama %96 başarı oranında tanıma gerçekleştirilmiştir (Demirel, 2017). Kaya, Myo Arm kol bandı ile TİD'deki 10 adet rakamın tanınmasını amaçlamıştır. Toplanan sinyallerden anlamlı ve ayırt edici 7 öznelik bu çalışma için uygun bulunmuştur. KNN, Karar Ağacı, SVM ve YSA sınıflandırma algoritmaları uygulanmış ve el hareketleri tanımlama başarımları karşılaştırılmıştır (Kaya, 2018).

### 3. İŞARET DİLLERİ VE ÖZELLİKLERİ

#### 3.1. İşaret Dilleri

Dünya üzerinde diller, ortaya çıkışı açısından doğal diller ve yapay olarak ikiye ayrılır. İşaret dilleri anlam olarak yapılan işareti karşı tarafa iletmede görsel kanalı kullanan dillere verilen genel addır. İşaret dilinin tarihi oldukça eski çağlara kadar uzanmaktadır. Tarih boyunca konuşma ve duyma yetilerinin kullanamayan işitme engelli insanlar iletişim kurabilmek için kendi içlerinde kullandıkları farklı işaret dilleri oluşturmuşlardır. İşaret dilleri kendilerine özgü sözcük dağarcığı ve dilbilgisel yapısına sahip doğal dillerdir. Doğal insan dillerine Türkçe, İngilizce, Fransızca gibi konuşma dilleri ve işaret dillerine Türk işaret dili (TİD), Amerikan işaret dili (ASL), Arap işaret dili (ArSL) gibi işaret dilleri olarak iki kategori altında incelenir.

Dilbilimciler hem işaret yoluyla hem de konuşma yoluyla yapılan iletişimi doğal dil saymışlar, her ikisi de soyut, uzun süren bir süreç ve zaman içerisinde belirli aşamalardan geçerek ortaya çıkmışlardır. İşaret dilleri sözsüz bir iletişim çeşidi olarak kullanılan beden dili ile karıştırılmamalıdır.

İşitme engelli bireylerin bulunduğu toplumların olduğu her yer ve alanda işaret dilleri önemli bir iletişim yolu olarak gelişmiş ve aynı zamanda yerel işitme engelli kültürlerinin temelini oluşturmuşlardır. İşaret dili çoğunlukla sağır ve ağır işitenler tarafından kullanılması ile birlikte fiziksel olarak konuşamayan, sesli dili farklı engel sebepleriyle kullanamayan ve sağır aile bireyleriyle birlikte bulunan aile üyeleri gibi işiten bireyler tarafından da kullanılmaktadır.

İşaret dillerine baktığımızda konuşma dillerinden türememişlerdir. Özellikle Amerika Birleşik Devletleri, İngiltere ve Avustralya gibi İngilizce konuşan ülkelerde birbirine benzer İngilizce lehçeleri konuşulmasına rağmen farklı işaret dilleri kullanılmaktadır. Bu üç işaret dili de (ASL, BSL ve Avustralya İşaret Dili (AUSL)) İngilizceden türemediği gibi, aynı zamanda birbirlerinden de dilbilgisel yönden farklılık göstermektedir.

İşaret dillerinin gösterimleri sadece konuşma dillerindeki sözcüklere karşılık gelen işaretlerin yan yana getirilmesi ile oluşturulmamaktadır. İşaret dillerinin kendilerine özgü bir dilbilgisi yapısı da kullanılmaktadır. İşaret dilleri Türkçe dilinde olduğu gibi sondan eklemeli bir dil değildir. Genel olmamakla beraber, işaret dilleri arasında ciddi benzerlikler de bulunur. İşaret dillerinin sondan eklemeli dil olmaması, bunları zayıf bir dil olduğunu göstermez.

Çünkü konuşma dillerine baktığımızda çoğunluğu sondan eklemeli bir dil olmadığı gibi bazılarında çok az son ek bulunmaktadır.

İşaret dilleri, konuşma dillerinin bireylere sunduğu özellikle sosyal ve bilişsel işlevleri benzer biçimde yerine getiriyor olmasına karşın, bilgi iletiminde görsel ve uzamsal yöntemleri kullandığı için, konuşma dillerinin işitsel ve sözel yöntemlerinden daha farklı yapısal özellikler sunmaktadır. Konuşma dili, oluşum sırasında ses dizgesinden yararlanırken, işaret dili başlıca parmaklar, eller, gövde ve yüzü kullanmaktadır. Bu da işaret dilindeki tüm dilsel yapıların oluşumunda görsel ve uzamsal ilişkilerin konuşma diline göre daha çok ön plana çıkmasına neden olmaktadır

Sesbilim ve biçimbilim gibi temel dil yapılar incelendiğinde, işaret dili ve konuşma dili ile arasındaki farklılıklar ortaya çıkmaktadır. Konuşma dili, dil seslerini oluşturmakta ve oluşturulan bu sesleri, işitsel sistem yoluyla algılamaktadır. İşaret diline baktığımızda ise, konuşma dilinden ayıran 5 temel sesbilimsel değişkene sahiptir. Bunlar el şekli (hand shape), ellerin konumu (location), el yönelimi (orientation), hareketler (movement) ve el dışı hareketler (non manual movements) olarak söyleyebiliriz. Konuşma dilinde sözcük içindeki bir sesin değişimine bağlı olarak farklılık olsa da anlamlı sözcük oluşmaktadır.

İşaret dili kendine özgü dilbilgisi, alfabeti ve işaretleri bulunan normal bir dile ait bütün özellikleri taşımaktadır. Bir cümlenin yapısından başlayarak işareti yapana kadar geçen bütün süreç içerisinde işaret dilinin kendine özgü kuralları içinde olmaktadır. Her sözcüğün işaret dili karşılığı olmasa da çoğunun işareti bulunmaktadır. Sözcüğün işaret karşılığı olmadığı ya da diğer bazı özel durumlarda alfabe kullanılmaktadır (Dikyuva vd., 2015).

### **3.2. Türk İşaret Dili (TİD)**

Her ülkenin işaret diline ait ifadeler aynı olmamakla birlikte yapılan bu işaretler ülkelere ve lehçelere göre farklılık göstermektedir. Hatta aynı ülkedeki farklı bölgelerde kullanılan bir sözcük farklı şekilde gösterim sağlayabilmektedir. Bilinen en eski ve köklü işaret dillerinden biri olduğu bilinen TİD ise , diğer dünya işaret dilleri ile tarihsel olarak doğrudan ve doğrusal ilişkili değildir. TİD başka bir işaret dilinden türememiş ve aynı zamanda kendine özgü yapısal özelliklere sahip bir işaret dilidir. Bunun yanında TİD, BSL ile benzerliklerde göstermektedir. Bugün ülkemizde kullanılan işaret dilinin temelleri ise Osmanlı dönemine kadar uzanmaktadır. Günümüze kadar zamanla yaşanan ve yapılan değişikliklerle birlikte kullanılan işaret dili bu dönemde kullanılan işaret dilden farklılıklar

göstermektedir. TİD tarihinin oldukça eski olmasına rağmen TİD için yapılan araştırmalar ise çok eskiye dayanmamaktadır.

TİD, işitme engelli birey toplum içinde önemli bir iletişim aracıdır. Kişi işitme engelli olmasa bile iş, aile ve sosyal toplum yaşantısında kendini sağır bir toplum içerisinde bulabilir ya da yaşamının bir yerinde hayatı bu kişilerle kesişebilir. Sağır toplumunu tanımak ve onlarla gerektiğinde iletişime geçebilmek için TİD dilbilgisi yapısının nasıl işlediğini ortaya koymak toplumsal bir gereksinimdir.

TİD, kendi dilbilimsel özelliklerine sahip ve işitme engelliler tarafından birincil dil olarak kullanılan dil olma özelliğini taşır. TİD'in kendine özgü kullanılan ifade araçları olduğu ve kullanıcıları tarafından özgün bir şekilde oluşturulduğu bilinmektedir. Bu özelliklere sahip TİD, sanılanın aksine, Türkçenin işaretlerle taklit edilmesinden öte, farklı bir dil olarak değerlendirilmelidir. TİD, Türkçe dil yapısından türetildiği için Türkçe ile benzerlikler göstermektedir. Fakat cümle yapılarının karmaşıklık düzeyi arttıkça bu benzerliklerin azaldığı görülmektedir.

TİD, temel dilbilimsel özellikleri bakımından Türkçeden ayrıldığı noktalar vardır. Fakat söz dizimi ve anlamsal açıdan bakıldığında ise TİD ile Türkçe birçok yönden benzerlikler göstermektedir. Arasındaki asıl farklılıklar olumsuzlaştırma, çoğullaştırma, zaman gibi dilbilimsel özellikler açısından bakıldığında ortaya çıkmaktadır.

Güncel TİD sözlüğünde yaklaşık 2000 üzerinde sözcük bulunmaktadır. Bu sözcükler alfabe, sayılar, hayvanlar, meslekler, görsel kavramlar, zaman, anlatımlar, yer isimleri ve zamirler gibi çeşitli kategorilere ayrılmışlardır. TİD bu sözcükler üzerinden ifade edilmektedir. Yani her sözcüğün bir işaret dili karşılığı bulunmaktadır. İşaret dili sözcükleri türetmek için bir nesnenin veya olgunun algısal görünümü göz önüne alınır. Örneğin, *merhaba* sözcüğü günlük yaşamımızda da sıklıkla yaptığımız başımıza eli hafif vurarak ifade edilir. Diğer bir örnek ise, *araba* sözcüğü için iki eli direksiyon tutar gibi tutup hafif sağa sola sallayarak ifade edilir.

TİD sözcüklerinin yeterli olmadığı durumlar da söz konusu olmaktadır. Böyle durumlar ve özel gereksinimlerde TİD alfabesi devreye girer. İşaret dili alfabesi TİD parmak alfabesinin her harfi için bir işaret karşılığı belirlenerek oluşturulmuştur. Bazı işaret dili sözcükleri Türkçede bulunan birden çok sözcüğün anlamını taşıyabilir. Örnek vermek gerekirse, bir el avuç içi yere bakacak şekilde düz dururken, diğer elin parmakları hafif açık aşağı doğru dik duracak şekilde bu elin üstünden kapanarak yukarı doğru

çekildiğinde *dağ* ve *tepe* anlamlarının gelirken, aynı zamanda özel isimlerden de *Ağrı*, *Mardin* ve *İsveç* sözcüklerinin anlamlarını taşır. Türkçe ile TİD arasında bu ve benzeri sebeplerden ötürü büyük farklar bulunur. Aynı konuşma dilinde olduğu gibi bu dilin de bir gramer yapısı bulunmaktadır. Hatta konuşma diline göre karşılaştırsak daha da zor bir yapıya sahiptir.

Türkiye’de özellikle son yıllarda işaret dili öğrenmeye yönelik duyulan ilgi ve alaka oldukça artmaktadır. Sadece konuşma ve duyma yetilerine sahip olmayan bireyler değil, bu özelliklere sahip olup sözel dili kullanamayan bireylerle anlaşılabilir onlarla iletişimlerinin aksamamasını isteyen toplum içinde önemli bir kültür düzeyine sahip insanlarda bu dili öğrenmeye oldukça isteklidir. Toplumsal olaylarla ilgili sosyal sorumluluk projeleri başta olmak üzere seçim kampanyalarında, televizyon programlarında ve diğer birçok ortamda karşımıza çıkan işaret dilinin öğrenilmesine karşı artan istek bulunmaktadır.

Her ülkenin işaret dili kendine özgüdür. Dünyada bilinen ve üzerine bilimsel araştırma yapılan 100’ün üzerinde işaret dili vardır. 1963 yılında ise uluslararası işaret dili çıkarılmıştır. Anadili işaret dili olan sağırın bütün işitme engellilere veya sağırlara oranı yaklaşık %10 oranındadır. Türkiye İstatistik Kurumu verilerine göre, işaret engelli nüfus yüzdesinin yaş ile arttığı da görülmektedir. Aynı zamanda Türkiye nüfusunun %3,5’sinin engelli, %0,35’sinin işitme engelli olduğu, %65’sinin sonradan işitme engelli olduğu araştırmalar göstermektedir. 29 harfin parmak karşılığı işaretler bulunmaktadır. Bununla birlikte Türkçede 2000’ün üzerinde sözcüğün karşılığı olan TİD hareketleri bulunmaktadır. Bunun yanı sıra Çin ya da Japon benzeri fonetik alfabelerde kısıtlı sayıda harflerin parmak işaretleri mevcuttur.

İşitme engelliler genellikle 3 farklı iletişim yöntemi kullanmaktadır. Parmak alfabesi, işaret dili ve dudak okuma. İlk iletişim yönteminde kullanıcılar her bir harf ya da rakam için bir işaret yapar. Parmak heceleme (finger spelling) olarak da adlandırılan bu ilk yöntem genelde sözcük işareti bulunmayan ya da anlaşılabilen sözcüklerin anlatımı için kullanılır. İkinci yöntemde ise her bir sözcük ya da cümle için bir işaret yapılır. İşaret dili sözcükler, mimikler ve jestler kullanılarak oluşturulmaktadır. Parmak alfabesi harf tabanlı, işaret dili ise sözcük tabanlı simgelerden oluşmaktadır. Bir dilin kendine özgü dilbilgisi ve sözcükleri olmakta ve buna özgü simgeleme yapılmaktadır. Ellerle birlikte dudak, mimikler ve gözler de önem taşımaktadır. İşitme engellilerin iletişiminde dudak okuma ile iletilmesi gereken bilginin tanınması güçlendirilmektedir. Bir örnek olması açısından, Şekil 3.1’de TİD’de **A** harfinin ve Şekil 3.2’de de TİD hareketlerinden *asla* ve *aslan* sözcüklerinin gösterimi verilmiştir. İşaret dillerinde yer alan harf ve sözcük kullanımı statik (postures) ve dinamik

(gestures) hareketlerden oluşmaktadır. Statik işaretler elin herhangi bir anındaki duruşu ya da tek bir imgeyle temsil edilmektedir. Dinamik işaretler ise elin işaret sırasındaki hareket konum ve şekillerindeki değişimlerden meydana gelmektedir.



Şekil 3.1. TİD alfabesinde A Harfi (Güngör vd., 2015)



Şekil 3.2. TİD hareketleri (Sözcük) (Güngör vd., 2015)

### 3.3. İşaret Dili Tanıma Sistemleri

İşaret dili tanıma konusunda, farklı makine öğrenme yöntemleri kullanılarak günümüze kadar birçok sistem tasarlanmıştır. İşaret dilleri üzerine yapılan ilk çalışmalar elektronik eldiven tabanlı sistemlerdir. İşareti gerçekleştiren kişinin eldiveni giymesi gerektiği için kullanım kolaylığı olmayan ve kişinin hareketlerini kısıtlayan özelliklere sahiptir. Görüntü tabanlı sistemler bir veya daha fazla kamera aracılığı ile alınan görüntülerin kullanıldığı sistemlerdir. Görüntü tabanlı sistemler için web kamerası, derinlik kamerası, renkli kamera, Kinect gibi farklı giriş aygıtları kullanılmaktadır. Bu sistemler üzerine, elektronik eldiven tabanlı sistemlere göre nispeten kullanım kolaylığı, düşük maliyet gibi

etmenlerden dolayı son yıllarda çok fazla çalışma ve araştırma yapılmıştır. Bu iki yaklaşım tarzı her ne kadar kendine özgü üstünlükleri olsa da günlük yaşam içinde kullanılabilirlikleri çok düşüktür. Bunların yanı sıra elektronik eldiven tabanlı çalışmalara benzer ivmeölçer (ACC), Elektromiyogram (EMG) ve üretici tarafından geliştirme süreci durdurulan kol kaslarını, hareketlerini ve mekansal verileri ölçen giyilebilir bir kol bandı olan Myo Armband cihazı kullanılarak yapılan çalışmalarda bulunmaktadır. Bu cihazlarında elektronik eldiven ve görüntü tabanlı sistemlerde olduğu gibi günlük yaşam içinde kullanılabilirlikleri düşüktür. Bu çalışmada hem günlük yaşamın her yerinde hem de taşınabilir bir sistem geliştirilmesi için ellerin ve parmakların hareketini ve konumunu hassas bir şekilde algılayan ve izleyen LMC cihazı özellikle tercih edilmiştir. Bu alanda çeşitli ülkelerde çalışmalar olmasına rağmen TİD tanıma üzerine yeteri kadar çalışma yoktur. Bu çalışma kapsamında LMC ve Kinect cihazı kullanılarak, TİD sistemleri için hareket tanıma sistemi geliştirilmiştir.

## 4. KULLANILAN CİHAZLAR

Bu çalışma kapsamında gerçekleştirilen uygulamalarda LMC ve Kinect cihazları önce ayrı ayrı kullanılmış, son uygulamada ise birlikte kullanılmıştır. Bu cihazların teknik özellikleri ve donanım yapıları aşağıda ele alınmıştır.

### 4.1. Leap Motion Controller (LMC) Cihazı

LMC şirketi, 2010 yılında San Francisco'da Michael Buckwald ve David Holtz tarafından kurulmuştur. LMC cihazı, şirket tarafından geliştirilerek ilk olarak 2013 Temmuz ayında piyasaya sürülmüştür. Cihaz yazılımı, teknolojiyi yenilikçi şekillerde kullanan yeni programlar oluşturmak isteyen geliştiricilere açıktır. LMC el hareketlerini kullanarak yazılımla etkileşime girilmesine izin veren Şekil 4.1'de verilen ön görünüme sahip bir cihazdır.



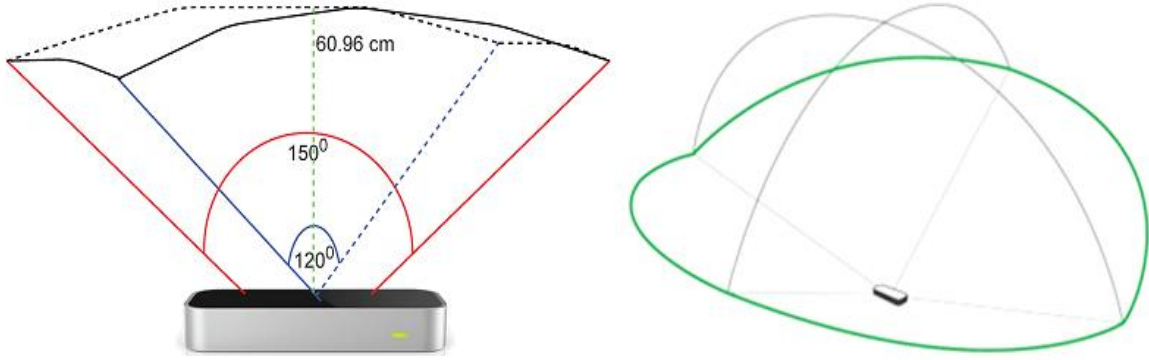
Şekil 4.1. LMC cihazının görünümü (LMC)

LMC, klavye, fare veya izleme dörtgeni arayüzü (trackpad ) ile birlikte çalışan, tüm bu öğeleri tamamlayan (herhangi bir özel adaptör olmadan) ve kullanıcıya ekstra hareket özgürlüğü sağlayan el hareketlerine dayalı bir insan bilgisayar arabirim aygıtıdır. Bu aynı zamanda, bilgisayar ve kullanıcı arasındaki etkileşimin yalnızca iki boyutlu bir alanda (fare veya klavye) çalışan fiziksel bir cihazla sınırlı olmadığı göstermektedir. Kullanıcı, bilgisayarla etkileşim kurmak için kendi vücudunu (elleri ve parmakları) kullanarak üç boyutlu bir deneyim yaşayabilir. LMC, her bir elin, parmağın veya aletin hareketini algılamak ve izlemek için kullanıcı ve bilgisayar arasındaki üç boyutlu alanı kullanır. LMC boyutları 13x13x76 milimetredir ve ağırlığı 45 gramdır, bu da onu taşınabilir ve gömülü bir cihaz yapar. Ayrıca LMC, veri iletimi için kullanılan aynı USB arabirimi tarafından beslenir ve dizüstü bilgisayarlar ve tabletlerle bütünleşmesini artırır. LMC, Tablo 3.1'de görülebileceği gibi, metrik sistemi kullanarak elde edilen verilerin birimleri sunulmuştur.

**Tablo 3.1.** LMC cihazını ölçüleri ve birimleri

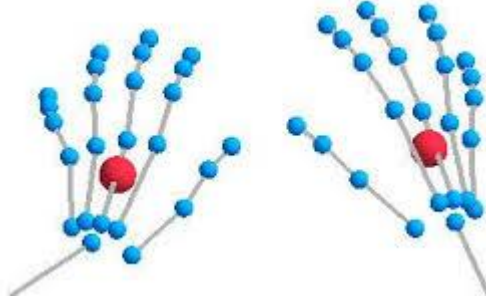
Ölçü	Birim
Uzaklık	mm
Zaman	us
Hız	mm/s
Açı	rad

Bu cihaz el, parmak ve benzer nesnelerin pozisyonlarını hareket ve eylemlerini tanır. Bu cihaz derinlik bilgisine sahip el görüntüsünün elde edilmesinde 2 kamera ve 3 kızılötesi LED'den oluşur. Bunlar, kızılötesi ışığı, görünebilir spektrum dışındaki 850 nm dalga boyu ile takip eder. Her iki IR kamerada LMC'nin merkezinden 20 milimetre simetrik olarak ayrılmıştır. Cihaz geniş açılı lensleri sayesinde, 150 derecelik geniş bir etkileşim alanına ve Z eksen derinliğine sahiptir. Bu alanı cihazın tam orta noktasında ters bir piramit görünümündedir. Etkin görüş alanı yaklaşık 25 ile 600 mm arasındadır. 290 fps gibi yüksek bir hız ile 10 parmak tarama yapabilen bu cihaz, oldukça yüksek bir hassasiyet oranı sunmaktadır.



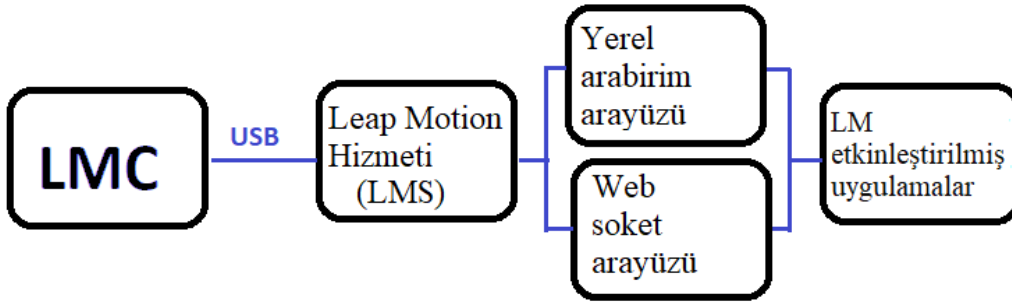
**Şekil 4.2.** LMC cihazının etkileşim hacmi ve görüş alanı (LMC)

Şekil 4.2'de verilen LMC cihazının etkileşim hacmi ve görüş alanındaki el ve parmak hareketlerini saniyede 200 kareye kadar tarayarak sürekli olarak bir veri dizisi ve görüntü kareleri (frame) sağlayarak bilgisayara gönderir. Şekil 4.3'da görünümü verilen el ve parmak hareketleri ile ilgili çıkarılan başlıca duyuşsal veri tip özellikleri yön, konum, hız, güvenilirlik, yakalama gücü, sıkıştırma gücü, daire merkezi ve yarıçaptır.



Şekil 4.3. LMC el ve parmak noktaları (LMC)

LM yazılımı, LMC tarafından yakalanan verileri, el pozunu, parmaklar ve jestlerin tahminini iyileştirmek için bir insan elinin tüm yapısı hakkında bilgi sağlayan bir insan elinin dahili modeliyle birleştirir. LM yazılımı, LMC ve LM etkinleştirilmiş uygulamalar arasında doğrudan bir iletişim kanalı sağlamaz. Bunun için, her iki öge arasındaki iletişimi sağlanmasında bir köprü olarak bir ara iletişim noktasının ihtiyaç duyar. LMC ve LM etkinleştirilmiş uygulamalar arasındaki bu iletişim köprüsü, kullanıcı müdahalesi olmadan arka planda çalışan bir bilgisayar programı olan bir Windows hizmeti tarafından sağlanır. LM hizmeti (LMS) çalıştığında, USB veri yolu üzerinden LMC'ye bağlanır. Ardından, LM etkinleştirilmiş uygulamalar veri almak için LMS'ye erişim sağlar. Bu verilere erişim iki farklı şekilde yapılabilir, yani LMC SDK iki çeşit uygulama programlama ara yüzü (API) sağlar. Bu API'ler, yazılım uygulamaları oluşturmak için bir dizi genel işlev ve araçtan oluşur ve LMC'den veri alınmasını kolaylaştırır. Bu iki API, yerel arabirim ve web socket ara yüzünden oluşur. LM SDK mimarisi ve LMC'nin nasıl bağlandığı ve uygulamalara nasıl veri sağladığı Şekil 4.4'de sunulmuştur.



Şekil 4.4. LMC SDK mimarisine genel bakış (LMC)

Yerel arabirim ara yüzü, yöntemleri ve veri yapılarını içeren bir dizi sınıftan oluşan dinamik olarak yüklenen bir kütüphane (DLL) aracılığıyla sağlanır. Veriler, LMS aracılığıyla bu kütüphane kullanılarak LM etkinleştirilmiş uygulamalara sağlanabilir. Doğrudan C++ ,

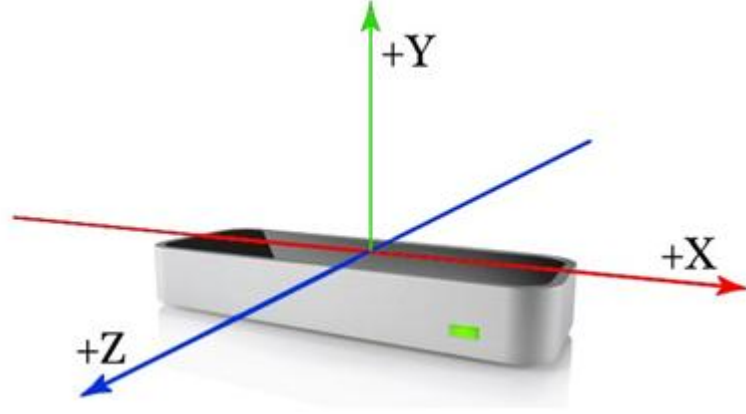
Objective-C, Java, C# ve Python için sağlanan dil bağlantılarından biri aracılığıyla programlanabilir. LMC tarafından elde edilen verileri temsil etmek için veri toplama modülü tarafından kullanılan en önemli sınıf ve yapıların özeti Tablo 3.2’de sunulmuştur.

Web soket ara yüzü, LMS ile bir ara yüz oluşturmak ve bir web sunucusu ve bir web tarayıcısı kullanarak LMC'den veri almak için diğer API'dir. İstemci ve sunucu arasındaki iletişim, köprü metni aktarım protokolü (HTTP) aracılığıyla yapılır. LMS, LMC tarafından alınan verileri bu bağlantı noktası numarasında müşterinin isteklerini dinleyen web sunucusuna göndermek için yerel ana bilgisayar etki alanında bir web soketi kullanır. Web sunucusu ve web tarayıcısı arasındaki verilerin kontrolü ve yönetimi, JavaScript Object Notation (JSON) mesajları ile gerçekleştirilir.

**Tablo 3.2.** Yerel arabirimi oluşturan sınıflar ve açıklamaları (LMC)

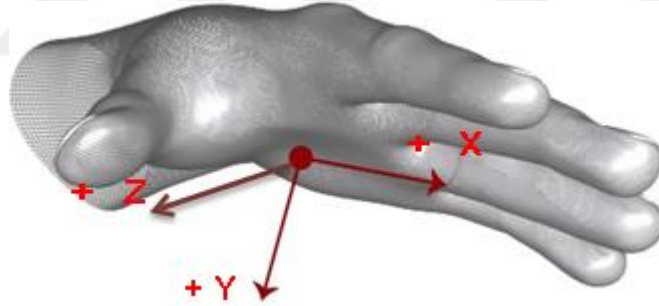
Sınıf	Açıklama
Frame	Veri modelinin köküdür ve izlenen tüm varlıklara erişim sağlar. Her güncelleme aralığında yeni bir çerçeve oluşturulur.
Hand	Eller, LMC tarafından izlenen ana varlıktır. LMC, dahili modeli mevcut verilerle eşleştirir. Her bir elin konumu, yönü, görünen parmaklar hakkında bilgi sağlar.
Arm	Bir elin bağlı olduğu kolun pozisyonunu, yönünü ve yönünü tanımlar.
Pointable: finger or tool	Algılanan her nokta uçlu nesnenin yönünü, uzunluğunu, uç konumunu veya hızını tanımlar.
Bone	Her kemiğin konumunu ve yönünü temsil eder (Metacarpal, proximal, medial ve distal).
Gesture	LMC tarafından tanımlanan dört tür hareket: CircleGesture, KeyTapGesture, ScreenTapGesture ve SwipeGesture.
Image	LMC için ham sensör verilerini ve kalibrasyon ızgarasını sağlar.
Vector	Bazı yararlı matematik işlevlerini tanımlamanın ve LMC tarafından yakalanan her nesnenin noktalarını ve yönlerini tanımlamanın kolay bir yolunu sağlar.
Matrix	Rotasyon gibi özellikleri temsil eder.
Interaction Box	LMC görüş alanı içinde kullanıcının LMC koordinatlarını normalleştirmesine ve eşlemesine izin veren doğrusal bir alan tanımlayan bazı işlevler sağlar.

LMC tarafından gerçekleştirilen tüm ölçümler farklı koordinat sistemlerine atıfta bulunabilir. Bunlar LMC, el ve uygulama olmak üzere 3 adet koordinat sistemi kullanır. LMC koordinat sistemi, LM SDK tarafından tanımlanan koordinat sistemine dayanır. LMC'nin merkezinde koordinat sistemi orijinini ayarlar. Şekil 4.5’de verilen sağ el kartezyen koordinat sistemi ve üç eksenin orijinini ve düzeni gösterimi sunulmuştur.



Şekil 4.5. LMC sağ el koordinat sistemi (LMC)

El koordinat sistemi, LM SDK tarafından tanımlanan el varlığına dayanır. Koordinat sistemi orijinini el düzleminin (avuç içi) ortasına ayarlar. Şekil 4.6, üç eksenin düzenini gösterir. Uygulama koordinat sistemi, uygulamanın çalışma alanı tarafından tanımlanır. LM koordinatları, uygulama koordinatlarına eşlenir. Bu amaçla kullanıcı, LMC koordinatlarını uygulamaya uyacak şekilde nasıl ölçekleyeceğine karar verilmelidir. Ölçek faktörü ne kadar büyük olursa, ara yüz küçük fiziksel hareketlere o kadar duyarlı olacaktır. Hız ve hassasiyet arasındaki en iyi dengenin bulunması gerekir.



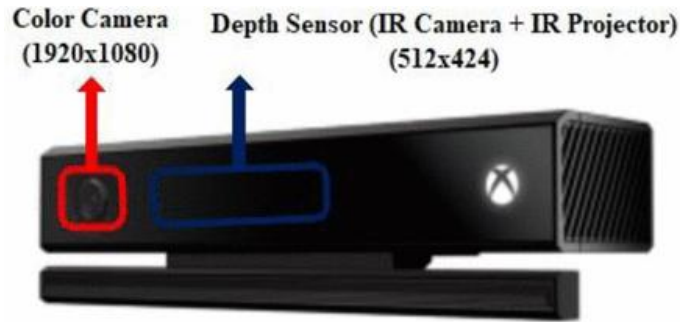
Şekil 4.6. El koordinat sistemi (LMC)

Hareket temelli işlemlerin algılanmasında birçok teknoloji kullanılmaktadır. Gerek kamera gerekse sensörlerden oluşan bu sistemler insan bedeni ve eklemlerini algılanarak insan hareketlerinin tespiti yapılmaya çalışılmaktadır. Ancak yapılan çalışmalar incelendiğinde başarımın istenen düzeylerde olmadığı görülmektedir. Son yıllarda insan hareketlerinin algılanmasında Microsoft Kinect cihazı da kullanılmaktadır. Bu cihaz üzerinde bulunan kızılötesi kamera sayesinde insan hareketleri ve eklemleri algılanabilmekte ve izlenebilmektedir. Kinect, insanda bulunan 25 farklı noktayı algılayabilme özelliğine sahiptir. Ancak parmak hareketlerinin algılanmasında yeterli hassasiyeti sağlayamamaktadır. Kinect, tüm vücut hareketlerini tanıyabilmek için daha büyük ölçekte çalışmak üzere tasarlanmışken,

LMC ise üç boyutlu uzayda gerçekleşen el ve parmak hareketlerini milimetrenin yüzde birine varan hassaslıkta algılanması ve izlenmesi konusunda gayet hassas ve başarılıdır.

#### 4.1. Kinect Cihazı

İnsan hareketinin algılanmasındaki kullanılan bir platform olan Kinect, yapısında bulundurduğu sensörler vasıtasıyla insan vücudu hareketlerini sayısal ortama aktarabilmektedir. Kinect, içinde renkli kamera, dâhili bir mikrofon, kızılötesi ışın (IR) projektörü ve kızılötesi kamera (IR) barındıran Microsoft, Rare ve PrimeSense kuruluşları tarafından ortak oluşturulan bir hareket algılayıcısıdır. Piyasaya oyun konsollarının bir parçası olarak çıkmasına rağmen biyomedikal ve robotik gibi farklı alanlarda kullanılır hâle gelmiştir. Microsoft Xbox 360 ve Windows uyumlu Kinect sensörü son derece gelişmiş algılama sensör donanımları içermektedir. Şekil 4.7’de görüldüğü üzere Kinect kamera üzerinde RGB kamera, çoklu dizi mikrofonlar ve yüz ve ses tanıma yetenekleri ile birlikte tam vücut 3D şekilde hareketleri belirtilen aralıklarla yakalama yeteneğine sahip derinlik sensörü bulundurmaktadır (Kramer vd., 2012). Halen piyasada kullanıcıların ve geliştiricilerin kullanımına sunulmuş durumda olan 2 farklı Kinect kamera sürümü bulunmaktadır. İki kamera sürümü arasındaki teknik farklılıklar Tablo 3.3’de sunulmuştur (Caruso vd., 2017) (Corti vd., 2016).



Şekil 4.7. Kinect v2 görünümü

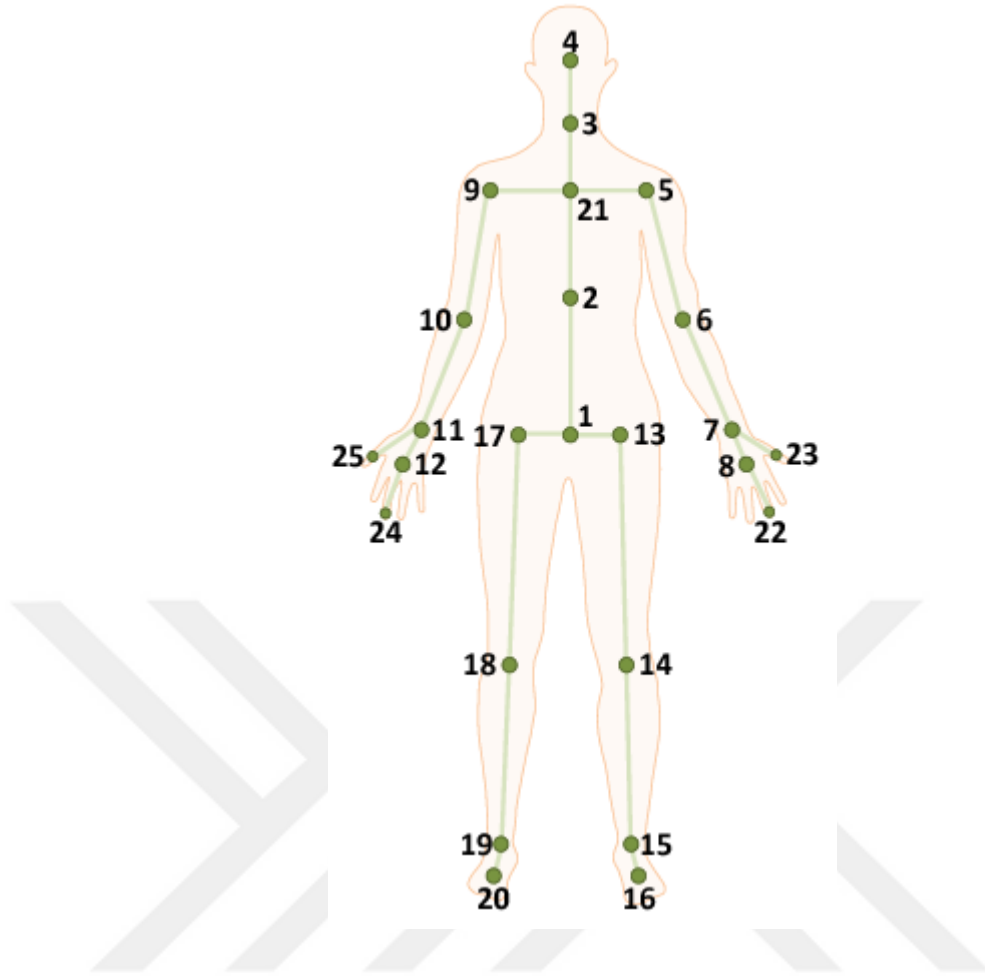
Kinect kamerası kullanıcıyı 80 cm ile 400 cm (4 m) arasında algılayabilmektedir. En ideal uzaklık 120 cm ile 350 cm arasındadır. Kullanıcının yüzünün kameraya dönük halde iken çalışmaktadır. Kamera kullanıcı 350 cm uzaklıktayken ortalama 2 metrelik bir insan boyunu algılayabilmektedir. Kinect yazılım geliştirme kitinde bir iskelet, insan vücudunda buldukları yerlere göre numaralandırılmış 25 adet eklemden oluşur ve bu eklemlerin pozisyonları ve uzaklıkları takip edilerek vücut hareketlerini görüntüye işlemektedir. Alınan görüntü, *skeleton* kütüphanesi yardımıyla bu eklemlerin, eklem tipi ve pozisyon (X,Y,Z)

bilgisi döndürülür. Şekil 4.8’de görüldüğü üzere alınan görüntüler üzerinde kişiye ait 25 eklem noktası belirlenmekte ve iskelet yapısı oluşturulmaktadır. Belirlenen 25 eklem noktasının düşey ve yatay eksendeki koordinatları belirli bir çerçeve sayısı boyunca alt alta eklenerek uzam-zamansal özellik vektörü oluşturulmaktadır. Ortadaki göz lazer gösterimi ile veri toplarken, sağdaki kızılötesi sensör bu ışınların gidiş ve geliş süresini hesaplayarak her bir noktanın mesafesini ve piksellerin koordinatlarını bildirmektedir. Kinect gömülü yazılımı ise bu verileri kullanarak iskelet yapısına göre 25 farklı bölgenin koordinatlarını çıkarır ve bunu XBOX ya da PC'ye göndermektedir. Kinect v2’e ait 25 iskelet eklem noktasının isimleri Tablo 3.4’de sunulmuştur.

**Tablo 3.3.** Kinect sürümleri arasındaki teknik farklılıklar

<b>Kinect özellikleri</b>	<b>Kinect 1</b>	<b>Kinect 2</b>
Renk Kamera	640 x 480	1920 x 1080
Saniyede Yakalanan Kare	30 fps	30 fps
Derinlik Kamera	320 x 240	512 x 424
Maksimum Derinlik Mesafesi	~4.5 M	~4.5 M
Minimum Derinlik Mesafesi	40 cm	50 cm
Yatay Görüş Alanı	57 derece	70 derece
Dikey Görüş Alanı	43 derece	60 derece
Eğim Motoru	Var	Yok
Tanımlı İskelet Eklemleri	20 Eklem	25 Eklem
İzlenen Tam İskelet	2	6
Desteklene USB Standardı	2.0	3.0
Desteklenen İşletim Sistemi	En az Windows 7, 8	En az Windows 8

Kinect cihazının RGB kamerası, 640x480 piksel saniyede 30 kare veya 1280x960 piksel saniyede 12 kare olmak üzere 2 farklı renk çözünürlüğü sunmaktadır. 3D derinlik sensörü ise 80x60 piksel, 320x240 piksel veya 640x480 piksel çözünürlükte saniyede 30 kare olmak üzere 3 farklı derinlik görüntüsünü, 16 bit derinlik bilgisiyle sunabilmektedir. Aynı anda 6 farklı kişiden yalnız ikisi için gerekli işlemleri gerçekleştirebilmektedir.



Şekil 4.8. Kinect 25 noktalı iskelet yapısı görünümü

Tablo 3.4. Kinect v2 25 iskelet eklem noktası isimleri

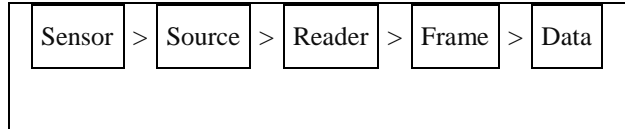
No	Eklem	No	Eklem	No	Eklem
1	SpineBase Omurga tabanı	10	ElbowRight Sağ dirsek	19	AnkleRight Sağ ayak bileği
2	SpineMid Omurga ortası	11	WristRight Sağ bilek	20	FootRight Sağ ayak
3	Neck Boyun	12	HandRight Sağ el	21	SpineShoulder Omurga omuz
4	Head Kafa	13	HipLeft Sol kalça	22	HandTipLeft Sol el ucu
5	ShoulderLeft Sol omuz	14	KneeLeft Sol diz	23	ThumbLeft Sol başparmak
6	ElbowLeft Sol dirsek	15	AnkleLeft Sol ayak bileği	24	HandTipRight Sağ el ucu
7	WristLeft Sol bilek	16	FootLeft Sol ayak	25	ThumbRight Sağ başparmak
8	HandLeft Sol el	17	HipRight Sağ kalça		
9	ShoulderRight Sağ omuz	18	KneeRight Sağ diz		

**Tablo 3.5.** Kinect v2 cihazı SDK mimarisi

Native App	.Net Apps	Windows Store App C#, VB, JS, C++/CX
Native API	.Net API	WinRT API
Kinect Runtime		
Kinect Drivers		
Physical Kinect Sensor		

Tabloda en altta fiziksel Kinect sensörleri bulunur. Üstünde Kinect sürücülere gerçek sensör donanımı ile iletişim kurmak için kullanılır. Kinect runtime bölümünde çalışması için gerekli olan bileşenler sağlanır. Program geliştiriciler için SDK, kendi uygulamalarını geliştirmek için kullanılacak Native API, .Net API ve WinRT API sağlar. Kinect v2 SDK mimarisi Tablo 3.5’de sunulmuştur. İçerisinde bulundurduğu fonksiyonlar sayesinde ses algılama, insan iskeletini tanıma gibi bir çok işlem kolaylıkla yapılabilmektedir. Birkaç sensör kullanmak mümkündür ve ayrıca birkaç uygulama aynı sensörü kullanabilir. Kinect v2 aynı çalışma zamanı modunu kullandığından, aynı anda birden fazla uygulama sensörü kullanabilir. Temel veri erişimi sürecinde, Kinect v2 SDK mimari tasarımını kullanır. Kinect v2 temel veri erişimi 5 aşamadan oluşmakta olup gösterimi Tablo3.6’da sunulmuştur. Bu aşamalar sensörü seçme, veri kaynağını seç, veri çerçeve okuyucusunu açma, çerçeve döngüsüne girme ve veriyi elde etmeden oluşur. Her verinin elde edilmesi diğer kaynaklardan bağımsız ve birbirine bağlı çalışmaz.

**Tablo 3.6.** Kinect v2 temel veri erişim aşamaları



Görüntüler bir akış şeklinde bilgisayara aktarılır. Bu akış görüntü çerçevelerinden oluşur. Bu akışa görüntü akışı (ImageStream) denilmektedir. Veri akışı görüntüleri renk, derinlik ve kızılötesinden oluşur. Veri akışı görüntülerine ait teknik özellikler Tablo 3.7’de sunulmuştur.

**Tablo 3.7.** Kinect v2 veri akışı görüntü özellikleri

Veri Akışı	Çözünürlük	Görüş alanı	Frames/Second
Color	1920x1080	+/- 85°(H) – 55°(V)	15 / 30 fps
Depth & IR	512x424	70°(H) – 60°(V)	30 fps

Koordinat eşleyici noktası, 3 koordinat sisteminden oluşur. Bunlar renk, derinlik ve kamera alanı noktası koordinat sisteminden oluşur. Her sistem arasında dönüşüm sağlar. Tekli veya çoklu dönüştürme yapmak mümkündür. Kinect v2 cihazı koordinat eşleyicileri gösterimi Tablo 3.8’de gösterimi sunulmuştur.

**Tablo 3.8.** Kinect v2 cihazı koordinat eşleyicileri

isim	Veri kaynağı	Boyut	Birim	Çözünürlük	Başlangıç
ColorSpacePoint	Color	2	pixels	1920x1080	Sol üst köşe
DepthSpacePoint	Depth, Infrared, Body index	2	pixels	512x424	Sol üst köşe
CameraSpacePoint	Body	3	metre		Depth & IR

Kinect V2 cihazı, sensörlerinden ya da veri giriş akışlarından 7 çeşit veri çerçeve kaynağı sunar. Bunlar renk (ColorFrameSource), kızılötesi (InfraredFrameSource), derinlik(DepthFrameSource), gövde index (BodyIndexFrameSource), gövde nesnesi (BodyFrameSource) ve ses (AudioFrameSource) ve çoklu çerçeve kaynağıdır. Bu veri çerçeve kaynaklarını gösterimi ve özellikleri Tablo 3.9’da sunulmuştur.

**Tablo 3.9.** Kinect v2 veri çerçeve kaynakları ve özellikleri

Veri çerçeve kaynağı	Özellikleri
Color frame source	1920 x 1080 @ 30 /15 fps Görüntü formatları (RGBA, BGRA, YUY2) Piksel başına 4 bayttan oluşur.
Infrared frame source	512 x 424 @ 30 fps Piksel başına 2 bayttan (16 bit) oluşur. Yalnızca kızılötesi ışığın yansımalarını alır.
Depth frame source	512 x 424 @ 30 fps Mesafe aralığı: 0.5 – 4.5 m. Piksel başına 2 bayttan (16 bit) oluşur.
Body Index frame source	512 x 424 @ 30 fps Piksel başına 1 bayttan oluşur.
Body frame source	Mesafe aralığı: 0.5 -4.5 m @ 30fps 6 adede kadar eşzamanlı gövde, 2 gövdede el durumu mevcuttur. 25 eklem gövde nesnesi içerir.
Audio frame source	Bir zaman aralığında yakalanan ses örneğidir. Bir ses ışını ile ilişkilendirilir
Multi frame source	Çok kaynaktan eşleşen bir çerçeve kümesi sunar. Seçilen kaynaklardan en düşük fps sunar.

Kinect cihazı insan iskelet yapısının tespiti, duruş ve hareketlerin belirlenmesindeki ön plana çıkmaktadır. LMC ise el ve parmak yapısına odaklanmıştır. Kinect v2 iskelet sistemini 25 eklem, LMC elleri 22 eklem parçasına bölmektedir. Bunun yanında menzilsel, görüntü karesi alımı, güç tercihi gibi farklar bulunmaktadır. LMC ve Kinect arasındaki teknik farklılıkları Tablo 3.10’de sunulmuştur.

**Tablo 3.10.** LMC & Kinect arasındaki teknik farklılıklar (Pauchot vd., 2015)

<b>Teknik özellikler</b>	<b>Kinect</b>	<b>LMC</b>
<b>Üretici</b>	Microsoft	LMC Inc.
<b>Üretim Yılı</b>	v1 (2010), v2 (2014)	2013
<b>Teknoloji</b>	1 Kızılötesi verici 1 Kızılötesi kamera (1.3 MP) 1 RGB kamera 4 Yönlü mikrofon	3 Kızılötesi verici 2 Kızılötesi kamera (1.3 MP)
<b>Görüntü karesi tazeleme</b>	9-30 Hz	200 Hz
<b>Saniyede Yakalanan Kare</b>	30 fps	290 fps
<b>Tanımlı İskelet Eklemleri</b>	25 eklem	22 eklem
<b>Tanıma</b>	Vücut hareketleri Yüz tanıma Ses tanıma	El hareketleri Parmak hareketleri
<b>Hassasiyet</b>	Santimetre	Milimetrenin yüzde biri
<b>Görüş alanı</b>	Yatay 57 derece Dikey 43 derece	Ön arka 120 derece Sağ sol 150 derece
<b>Menzili</b>	1.2 - 3.5 m	0.002 - 0.61 m
<b>Çalışma alanının zemin yüzeyi</b>	6 metrekare	1.16 metrekare
<b>Yapısı</b>	Kinect SDK	LMC SDK

## 5. ÖZELLİK ÇIKARIMLARI VE ÖZELLİK SEÇİMİ

Makine öğrenmesi ve örüntü tanıma gibi alanlarda gözlemlenen bir verinin ölçülebilir bir niteliğine özellik denir. Bunun ayırt edici, anlaşılır ve bağımsız özellikler içermesi sınıflandırma aşamasında önemli bir adımdır. Veri kümesini oluşturan özellikler gereksiz, benzer ya da aynı değerler içerebilir ve büyüklüğünden dolayı işlenmesi zor olabilir. Bu yüzden özelliklerin bir alt kümesinin seçilmesi ya da indirgenmiş özellikler kümesinin oluşturulması gerekir. Doğru yapılan bir özellik çıkarımı ya da seçimi sınıflandırma sürecinde başarıyı etkileyen en önemli unsurdur. Öncelikle veri toplama ve ön işlem aşama süreci anlatılmıştır. Daha sonra LMC ve Kinect cihazlarından özellik çıkarımları hakkında bilgi verilmiş. Son aşamada ise kullanılan özellik seçimi algoritması ve özellik çıkarımları işlenmiştir.

### 5.1. Veri Toplama ve Ön İşlem Aşaması

Önerilen sistem için Intel(R) Core(TM) i7-3632QM işlemci @2.20 GHZ hızı ve 6 GB RAM belleğe sahip bir bilgisayar kullanılmıştır. Veri kümesinin oluşturulması için LMC SDK sürüm 4.0 kullanılarak C# dili ile bir veri toplama sistemi geliştirilmiştir. Kinect cihazı kullanılan uygulamalarda ise Kinect SDK sürüm 2.0 kullanılarak ara yüz kodlaması WPF, yazılım kodlaması C# dili ile yazılmış bir veri toplama sistemi geliştirilmiştir. Sistemin güvenilirliğini ve doğruluğunu arttırmak için özellikle çift elle gerçekleştirilen statik parmak alfabeti harfleri ve dinamik sözcükler seçilmiştir. Bunun yanı sıra birbirine benzer dinamik sözcükler seçilerek sistemin doğruluğunun karşılaştırılması sağlanmıştır. Kinect ve özellikle LMC cihazın donanımsal yapısından dolayı, özellikle çift el kullanımlarında üst üste gelme ve iç içe geçme gibi durumlarda algılama güçlükleri ve veri kesilmeleri oluşmaktadır. Bununla birlikte özellikle çift elle gerçekleştirilen harflerin gösterimleri esnasında geçici parmak tıkanıklıkları, veri düzensizlikleri, veri kayıpları ve en önemlisi sistemin kilitlenmesi ve benzeri durumlar ortaya çıkmaktadır. Bu amaçla harflerin gerçek zamanlı yapıları esnasında her bir kol, dirsek, el ve parmaklar için belirlenen özellik çıkarım değerlerinin alınmasında sapmalar, kaymalar oluşmakta ve buna bağlı anlık sabit ve doğru değerler döndürmemektedir. Bu tür sebeplerden, hareket esnasında alınan verilerdeki olumsuzluk ve farklılıkları azaltmak için cihazdan her 100 milisaniyede bir alınan ve saniyede oluşan 10 görüntü karesinin ortalaması kayıt edilmiştir. Bu kayıtlar, bilgisayara USB üzerinden bağlanan LMC ve Kinect kullanılarak alınmıştır. Yapılan bu işlem ile yukarıda anılan sorunların önüne geçilmesi ve veri kümesinin doğruluk ve güvenilirliğinin artırılması hedeflenmiştir. İşaretlerin denekler

tarafından yapılma aşamasında aydınlatma koşulları değiştirilmeden gün ışığı aydınlatması kullanılmıştır. Denek, LMC ile arası 20 cm ve yerden 90 cm yükseğe konumlandırılmıştır. Kinect algılayıcı ise denekten 180 cm uzağa ve yerden 90 cm yükseğe, yere paralel olacak şekilde 0°'lik açı ile konumlandırılmıştır. Kinect ile LMC cihazı arası 160 cm belirlenmiştir. Her bir deneğin her bir parmak alfabeti ve sözcük için oluşturduğu kayıt ayrı bir dosya olarak kaydedilmiştir. Geliştirilen uygulamalara ait oluşturulan veri kümesini içeren csv uzantılı dosyanın Matlab dosya biçimine dönüştürme, normalisasyon, özellik seçimi, özellik çıkarımı, boyut indirgeme ve sınıflandırma aşamaları Matlab kullanılarak gerçekleştirilmiştir.

## 5.2. LMC İçin Özellik Çıkarımı

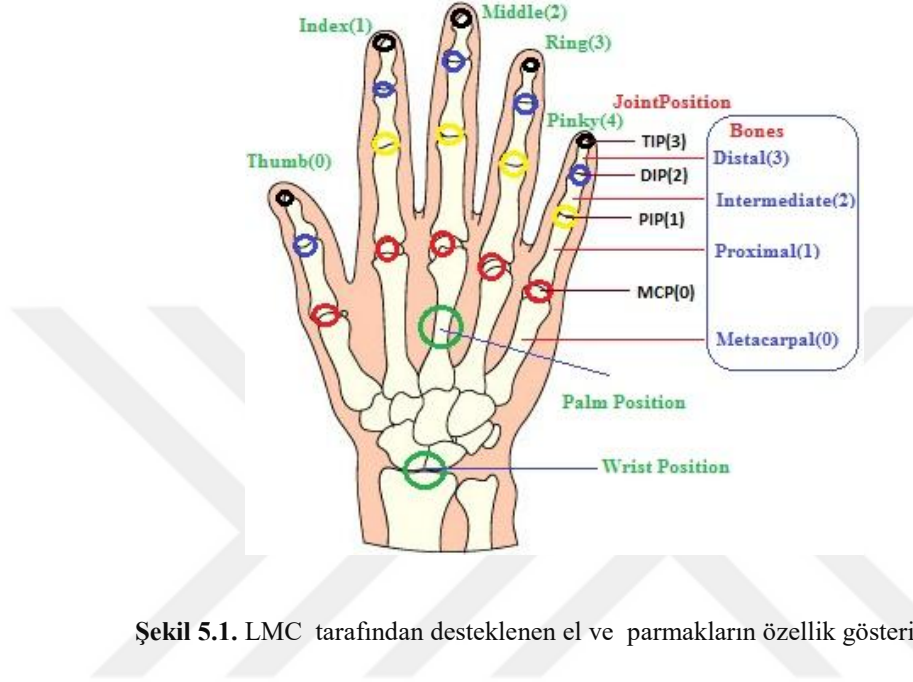
LMC cihazı için 2 çeşit özellik çıkarımı hazırlanmıştır. Birinci özellik çıkarımı ilk iki uygulama için belirlenmiş ve kullanılmıştır. Daha çok parmak ve eklemlere bağlı ve uzaklık mesafesi bilgisi içeren özellikler tercih edilmiştir. İkinci özellik çıkarımı LMC cihazının Kinect ile birlikte kullanıma bağlı belirlenmiş ve kullanılmıştır. Bu kapsamda, ilk özellik çıkarımından farklı olarak eklem ve eklem noktalarına bağlı ve uzaklık mesafesi bilgisi içeren özellikler çıkarılırken parmak hareketlerinin vektör bilgisi içeren özellikler eklenmiştir.

**Tablo 5.1.** LMC tarafından desteklenen önemli el ve parmak özellikleri

<b>LMC Özellikleri</b>			
<b>Eller</b>		<b>Parmaklar</b>	
<b>İsim</b>	<b>Tür</b>	<b>İsim</b>	<b>Tür</b>
Direction	Vector	Distance	mm
PalmNormal	Vector	Angle	Radian
PalmPosition	Vector	IsExtended	Bool
Leftmost	Hand	Direction	Vector
Rightmost	Hand	TipPosition	Vector
Fingers	Fingerlist	Bone(DISTAL)	Vector
Arm.ElbowPosition	Vector	Bone(INTERMEDIATE)	Vector
Arm.WristPosition	Vector	Bone(PROXIMAL)	Vector
Arm.Direction	Vector	Bone(METACARPAL)	Vector
Direction.Pitch	Radian	JointPosition(TIP)	Vector
Direction.Yaw	Radian	JointPosition(DIP)	Vector
PalmNormal.Roll	Radian	JointPosition(PIP)	Vector
		JointPosition(MCP)	Vector

TİD'de statik parmak alfabeti harflerinin ve dinamik sözcüklerin tanınması için uygun özellik çıkarımları yapılmıştır. Özellikler her bir harf ve sözcüğün yapılış durumu ve kol, dirsek, el, parmak ve eklem hareketlerinin gereksinim görülen her noktası dikkate alınarak

belirlenmiştir. LMC API tarafından desteklenen önemli el ve parmak özellikleri Tablo 5.1’de sunulmuştur. Cihazın hassas hareket sensörleri ve kameraları yardımıyla 3 boyutlu uzayda gerçekleşen el ve parmak hareketlerinden geçici zamansal ve uzamsal özellikler çıkarılmıştır. LMC API tarafından desteklenen el ve parmakların özellik gösterimleri Şekil 5.1’de sunulmuştur.



Şekil 5.1. LMC tarafından desteklenen el ve parmakların özellik gösterimi (LMC)

### 5.1.1. Birinci özellik çıkarımı

Birinci özellik çıkarımı sadece LMC kullanılan uygulamaları için el, parmak, eklem ve eklem noktalarının her biri için yön, uzunluk, mesafe, konum, açı, durum gibi özelliklerinden ayırt edici 119 tanesi belirlenmiştir. Bunlar arasında özellikle el ve parmak boyutuna bağlı kalmadan ve benzer harf ve sözcüklerin tanınmasına yönelik açı bilgisi kullanılarak herhangi bir kişinin kullanımı hedeflenmiştir.

Öncelikle 10 parmağın açık - kapalı durum özellik bilgileri çıkarılmıştır. Sağ elin işaret parmağının açık kapalı durum bilgisi alan kod, aşağıda verilmiştir.

*IsExtended\_set={f1,...,f10}*

`frame.Hands.Rightmost.Fingers[1].IsExtended`

Parmak ucu noktaları (TipPosition), avuç içi noktası (PalmPosition) ve parmak eklem noktaları (JointPosition) için uzaklık mesafesi bilgisine başvurulmuştur. Her 3 konum noktaları arası toplam 44 özellik belirlenmiştir. Sağ elin işaret ve orta parmakların TIP

parmak ucu noktaları arası uzaklık mesafesi bilgisini alan kod, aşağıda gösterilmiştir.

*Distance\_set={f11,...f54 }*

```
frame.Hands.Rightmost.Fingers[1].TipPosition.DistanceTo(frame.Hands.  
Rightmost.Fingers[2].TipPosition
```

Bu özelliklerin yanı sıra, açı bilgisi de önem arz etmektedir. Parmak ölçüleri ve büyüklükleri farklılık gösterdiği için parmak yönü (Direction), parmak kemik yönü (Bone) ve el ve avuç içi yönü açı bilgisine ihtiyaç duyulmuştur. Ayrıca birçok alanda kullanılan elin boylamsal, yatay ve dikey eksen yönünün açı bilgisi (pitch, yaw, roll)de eklenmiştir. Her yön arası açı bilgisi için toplam 29 özellik çıkarılmıştır. Sağ elin işaret ve orta parmaklar arası açı bilgisini alan kod, aşağıda gösterilmiştir.

*Angle\_set={f55,.....f83 }*

```
frame.Hands.Rightmost.Fingers[1].Direction.AngleTo(frame.Hands.Right  
most.Fingers[2].Direction)
```

Ayırt ediciliği güçlendirmek için bazı özellik vektörleri de ayrıca kullanılmıştır. Bunlar elin, avuç içi merkezinin (PalmPosition) ve yönünün (PalmNormal) vektör bilgileridir. Bunlarla birlikte kolun dirsek konum (ElbowPosition), bilek konum (WristPosition) ve yön vektör (Direction) bilgileri de kullanılmıştır. Elin ve kolun vektör yön bilgisi için toplam 36 özellik eklenmiştir. Sağ elin kol dirsek boyu koordinat bilgisini alan kod, aşağıda gösterilmiştir.

*Vector\_set={f84,.....f119 }*

```
frame.Hands.Rightmost.Arm.ElbowPosition.x
```

Bu çalışma kapsamında, TİDne ait iki uygulama için belirlenen özellikler yukarıda bahsedildiği üzere 4 kategori ve 16 başlık altında gruplandırılmış ve toplamda 119 özellik sınıflandırma aşamasında kullanılmak üzere belirlenmiştir. Belirlenen 4 kategori ve 16 başlık altındaki gruplandırılmış birinci özellik çıkarımının ölçü birimi, türü, isimleri ve sayıları Tablo 5.2'de sunulmuştur.

**Tablo 5.2.** Birinci özellik çıkarımının ölçü birimi, türü, isimleri ve sayısı

LMC ( 1. özellik çıkarımı )			
Ölçü	Tür	Özellik	adet
IsExtended	bool	Fingers.IsExtended	10
Distance	mm	TipPosition to TipPosition	15
		TipPosition to PalmPosition	8
		TipPosition to JointPosition	13
		JointPosition to JointPosition	7
Angle	radian	Fingers.Direction to Fingers.Direction	6
		Bone.Direction to Bone.Direction	9
		Fingers.Direction to Hands.Direction	4
		Fingers.Direction to Hands.PalmNormal	4
		Hands.Direction.(Pitch,Yaw,Roll)	6
Vector	float	Hands.PalmPositions(X,Y,Z)	6
		Hands.PalmNormal(X,Y,Z)	6
		Hands.Direction(X,Y,Z)	6
		Arm.ElbowPosition(X,Y,Z)	6
		Arm.WristPosition(X,Y,Z)	6
		Arm.Direction(X,Y,Z)	6
Toplam			119

### 5.1.2. İkinci özellik çıkarımı

LMC cihazının Kinect ile birlikte kullanımı uygulaması kapsamında farklı özellik çıkarımları kullanılmıştır. Sözcüklerin farklı uzunluk ve sürelerde gerçekleştirilmesi ve daha çok kol, el ve parmak hareketlerinin yoğunlukta olması daha kapsayıcı özellik çıkarımlarına yönlendirmiştir. Bu kapsamda özellikle eklem ve eklem noktaları için yön, uzunluk, mesafe, konum gibi özellikler çıkarılarak parmak hareketlerinin vektör bilgisini içeren özellikler eklenmiştir. Bu özellik çıkarımında birlikte uzaklık mesafesi bilgisi çok az kullanılmıştır. Bu doğrultuda her uygulama için kol, el, parmak noktalarının her biri için yön, uzunluk, mesafe, konum, açı, durum gibi özelliklerinden ayırt edici 110 tanesi belirlenmiştir.

Öncelikle sadece parmak ucu noktalarının (TipPosition), avuç içi noktası (PalmPosition) için uzaklık mesafesi bilgisine başvurulmuştur. Bu özellik çıkarımı kapsamında uzaklık bilgisi çok tercih edilmemiştir. Her parmak noktası için toplam 10 özellik belirlenmiştir. Sağ elin işaret parmağının TIP parmak ucu noktasının, avuç içi noktasına (PalmPosition) uzaklık mesafesi bilgisi alan kod, aşağıda gösterilmiştir.

*Distance\_set={f1,...f10 }*

```
frame.Hands.Rightmost.Fingers[1].TipPosition.DistanceTo(frame.Hands.  
Rightmost.PalmPosition)
```

Bunun yanı sıra, açı bilgisi de önem arz etmektedir. Parmak ölçüleri ve büyüklükleri farklılık gösterdiği için parmak yönü (Direction) ve el ve avuç içi yönü açı bilgisine ihtiyaç

duyulmuştur. Ayrıca birçok alanda kullanılan elin boylamsal, yatay ve dikey eksen yönünün açı bilgisi (pitch, yaw, roll)de eklenmiştir. Her yön arası açı bilgisi için toplam 34 özellik çıkarılmıştır. Sağ elin işaret ve orta parmaklar arası açı bilgisini alan kod aşağıda gösterilmiştir.  $Angle\_set=\{f11, \dots, f45\}$

```
frame.Hands.Rightmost.Fingers[1].Direction.AngleTo(frame.Hands.Rightmost.Fingers[2].Direction)
```

Ayırt ediciliği güçlendirmek için bazı özellik vektörleri de ayrıca kullanılmıştır. Bunlar elin, avuç içi merkezinin (PalmPosition) ve yönünün (PalmNormal) vektör bilgileridir. Bunlarla birlikte kolun dirsek konum (ElbowPosition), bilek konum (WristPosition) ve yön vektör (Direction) bilgileri de kullanılmıştır. Elin ve kolun vektör yön bilgisi için toplam 66 özellik eklenmiştir. Sağ elin kol dirsek boyu koordinat bilgisini alan kod, aşağıda gösterilmiştir.  $Vector\_set=\{f46, \dots, f110\}$

```
frame.Hands.Rightmost.Arm.ElbowPosition.x
```

Bu çalışma kapsamında, TİDne ait LMC+Kinect uygulaması için belirlenen özellikler yukarıda bahsedildiği üzere 3 kategori ve 12 başlık altında gruplandırılmış ve toplamda 110 özellik sınıflandırma aşamasında kullanılmak üzere belirlenmiştir. Belirlenen 3 kategori ve 12 başlık altındaki gruplandırılmış ikinci özellik çıkarımının ölçü birimi, türü, isimleri ve sayıları Tablo 5.3’de sunulmuştur.

**Tablo 5.3.** İkinci özellik çıkarımının ölçü birimi, türü, isim ve adetleri

LMC ( 2. özellik çıkarımı )				
Ölçü	Tür	Özellik		Adet
Distance	mm	Fingers.TipPosition to PalmPosition	10X1	10
Angle	radian	Fingers.Direction to Fingers.Direction	8X1	8
		Fingers.Direction to Hands.Direction	10X1	10
		Fingers.Direction to PalmNormal	10X1	10
		Hands.Direction(Pitch, Yaw, Roll)	2X3	6
Vector	float	Fingers.TipPosition(X, Y, Z)	10X3	30
		Hands.PalmPositions(X, Y, Z)	2X3	6
		Hands.Direction.PalmNormal(X, Y, Z)	2X3	6
		Hands.Direction.Direction(X, Y, Z)	2X3	6
		Arm.ElbowPosition(X, Y, Z)	2X3	6
		Arm.WristPosition(X, Y, Z)	2X3	6
		Arm.Direction(X, Y, Z)	2X3	6
			Toplam	110

Belirlenen özellik çıkarımlarından uygulama aşamasında denekler tarafından yapılan sözcükler içinden **börek** sözcüğünün 6 adet sağ ve sol ele ait avuç içi noktası

(Hands.PalmPositions(X,Y,Z)) vektör bilgisine ait örnek bir kayıt Tablo 5.4’de gösterimi sunulmuştur.

**Tablo 5.4.** Börek sözcüğüne ait sağ ve sol el avuç içi noktası vektör bilgileri

Sol el avuç içi vektör bilgileri			Sağ el avuç içi vektör bilgileri		
X	Y	Z	X	Y	Z
74,55	383,57	34,64	-144,28	375,3	10,56
74,2	387,04	24,1	-140,6	377,55	-0,6
73,67	386,27	7,59	-137,51	380,93	-18,74
74,84	387,65	-7,79	-136,07	382,97	-35,68
74,19	389	-14,05	-133,93	382,56	-41,18
72,6	394,77	-12,72	-132,6	389,08	-35,49
72,76	398,86	-4,71	-132,87	389,51	-23,73
73,64	406,66	6,36	-134,2	392,51	-7,37
73,77	410,84	23,34	-135,12	397,73	11,29
75,94	416,23	36,12	-132,63	400,23	25,65
76,96	417,52	37,83	-132,04	403,95	28,47
77,21	411,61	33,68	-131,55	399,48	21,67
77,13	404,61	25,14	-132,13	391,6	10,31
74,79	396,68	7,39	-130,21	384,46	-9,76
72,73	394,03	-4,57	-128,41	384,36	-24,61
69,61	391,82	-17,39	-127,19	388,77	-39,06
68,23	394,03	-26,76	-123,31	383,85	-47,79
67,83	395,58	-33,3	-121,05	382,01	-53,66
68,03	396,83	-37,29	-120,6	384,78	-55,84
69,61	397,51	-39,15	-122,38	382,01	-54,87
70,76	394,36	-38,7	-125,81	385,25	-56,5

### 5.3. Kinect Cihazından Özellik Çıkarımı

TİD’de dinamik sözcüklerin tanınması için uygun özellik çıkarımları yapılmıştır. Bu çalışmada, işaret dilini tanımlamak için kullanılan özellikler, Kinect tarafından yakalanan iskelet eklemlerinden alınmıştır. Şekil 4.8’de gösterildiği gibi Kinect tarafından yakalanan 25 adet iskelet eklem noktası gösterimi sunulmuştur. Bu kapsamda iskelet eklem noktası (vektör) ve yön bilgisine (orientation) ait iskelet hareketlerinin vektör bilgisini içeren özellikler seçilmiştir. Bunlara ek olarak el ve eklem bilgilerine bağlı kalmadan sözcüklerin tanınmasına yönelik açı bilgisi kullanılmıştır. Bu doğrultuda el ve kol iskelet eklem noktalarının yön, konum, açı gibi özelliklerden ayırt edici 110 tane belirlenmiştir.

Öncelikle 25 iskelet eklem noktasından gerekli olan eklem noktası koordinat vektör bilgisine başvurulmuştur. Bunlardan 16 adet iskelet eklem noktası bilgisi kullanılarak 48

özellik belirlenmiştir. Sol omuz eklem noktasına ait koordinat vektör bilgisi alan kod, aşağıda gösterilmiştir.

*vektor\_set={ f1,...,f48 }*

```
Body[frame.BodyFrameSource.BodyCount].Joints[JointType.  
ShoulderLeft].Position.X;
```

Bununla birlikte 25 iskelet eklem noktasından gerekli olan eklem noktasına ait yön bilgisi alınmıştır. Bunların içinden 16 adet iskelet eklem noktası yön bilgisi ait 64 özellik belirlenmiştir. Sol bilek eklem noktasına ait yön bilgisi alan kod, aşağıda gösterilmiştir.

*yon\_set={ f49,...,f112 }*

```
Body[frame.BodyFrameSource.BodyCount].JointOrientations[JointType.Wr  
istLeft].Orientation.X;
```

Bunun yanı sıra, ek olarak açı bilgisi de kullanılmıştır. İskelet eklem ölçüleri ve büyüklükleri farklılık gösterdiği için eklem noktaları açı bilgisine ihtiyaç duyulmuştur. 25 iskelet eklem noktasından 8 eklem noktasına ait açı bilgisi çıkarılmıştır. Sol omuz, sol dirsek ve sol bilek arası açı bilgisini alan kod aşağıda gösterilmiştir.

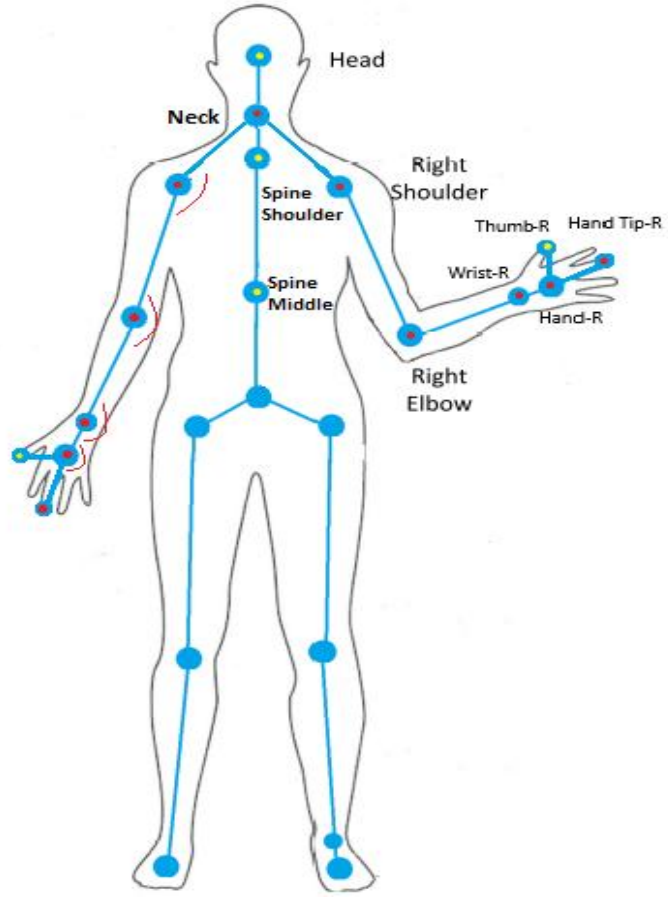
*Angle\_set={ f113,...,f120 }*

```
Vector.AngleBetween(((Joints[ShoulderLeft].Position.X-  
Joints[ElbowLeft].Position.X),(Joints[ShoulderLeft].Position.Y-  
Joints[ElbowLeft].Position.Y)),((Joints[WristLeft].Position.X-  
Joints[ElbowLeft].Position.X),(Joints[WristLeft].Position.Y-  
Joints[ElbowLeft].Position.Y)));
```

Belirlenen özellikler 3 başlık altında gruplandırılmış iskelet eklemlerinin özellik çıkarım bilgileri Tablo 5.5’de sunulmuştur. Belirlenen özellik çıkarımlarına ait kırmızı ve sarı olarak belirtilen iskelet eklem noktaları ve açı bilgilerine ait gösterim Şekil 5.2’de sunulmuştur. Bu kapsamda, dinamik çift elli sözcükler için belirlenen toplamda 120 özellik sınıflandırma aşamasında kullanılmak üzere hazırlanmıştır.

**Tablo 5.5.** Kinect iskelet eklemlerinin özellik çıkarım bilgisi

Kinect V2			
sn	Vector3 (X, Y, Z)	Yön (Orientation) (X, Y, Z, W)	Angle
1	Head	Head	Neck-ShoulderRight-ElbowRight
2	Neck	Neck	ShoulderRight-ElbowRight-WristRight
3	SpineShoulder	SpineShoulder	ElbowRight-WristRight-HandRight
4	SpineMid	SpineMid	WristRight-HandRight-HandTipRight
5	ShoulderLeft	ShoulderLeft	Neck-ShoulderLeft-ElbowLeft
6	ElbowLeft	ElbowLeft	ShoulderLeft-ElbowLeft-WristLeft
7	ThumbLeft	ThumbLeft	ElbowLeft-WristLeft-HandLeft
8	WristLeft	WristLeft	WristLeft-HandLeft-HandTipLeft
9	HandTipLeft	HandTipLeft	
10	HandLeft	HandLeft	
11	ShoulderRight	ShoulderRight	
12	ElbowRight	ElbowRight	
13	ThumbRight	ThumbRight	
14	WristRight	WristRight	
15	HandRight	HandRight	
16	HandTipRight	HandTipRight	
	48	64	8
Toplam:			120



Şekil 5.2. Belirlenen özellik çıkarımlarına ait eklem noktaları ve açı bilgileri

Belirlenen özellik çıkarımlarından uygulama aşamasında denekler tarafından yapılan sözcükler içinden **börek** sözcüğünün 8 adet kollara ait iskelet eklem açı bilgisine ait örnek bir kayıt Tablo 5.6’da gösterimi sunulmuştur.

**Tablo 5.6.** Börek sözcüğünün 8 adet kollara ait iskelet eklem açısı bilgileri

Sol el açısı bilgileri				Sağ el açısı bilgileri			
El	Bilek	Dirsek	Omuz	El	Bilek	Dirsek	Omuz
151,81	41,98	177,05	176,38	-155,85	-56,99	-157,84	152,44
161,33	38,15	170,89	170,98	-154,05	-61,62	-158,32	176,14
149,88	52,67	156,1	-167,42	-156,45	-37,08	-178,75	-173,17
148,23	48,02	175,42	145,92	-147,79	-53,81	-145,5	168,35
147,08	47,33	175,1	161,68	-157,23	-41,79	-163,11	175,6
147,47	49,48	147,31	-168,76	-158,21	-40,28	-165,44	177,6
148,42	50,28	142,31	-163,9	-148,52	-49,61	-154,99	173,72
149,81	38,37	174,53	-178,03	-149,28	-40,02	178,67	150,45
152,43	35,23	164	-154,35	-151,06	-39,12	-144,57	111,82
153,97	27,82	179,94	-161,32	-150,96	-39,84	-160,89	134,74
154,65	28,48	171,29	-154,87	-152,22	-37,02	-176,47	145,02
154,63	27,87	-178,74	-167,31	-152,41	-39,07	-172,16	139,7
154,48	28,56	-175,91	-175,23	-151,4	-40,39	-179,78	159,43
151,82	35,27	-175,69	173,43	-148,53	-56,41	-150,01	143,96
149,79	39,07	-177,23	168,59	-146,07	-51,8	-166,38	139,84
146,56	44,36	-179,52	154,39	-153,77	-44,76	-171,86	-167,63
143,13	54,47	41,37	-101,69	-153,12	-43,28	-171,32	-160,2
141,98	54,62	172,43	143,75	-146,85	-47,79	-172,04	-162,93
141,04	61,46	12,68	-116,64	-146,36	-47,9	-175	-157,65
141,21	52,42	119,12	148,97	-139,65	-57,6	-173,6	-161,04
141,97	51,65	90,23	-146	-139,2	-58,38	-169,33	-127,28

#### 5.4. Özellik Seçimi Algoritması

İşaret dili tanıma hareket tabanlı olduğundan, farklı ve benzer çeşitli hareketleri ayırt etmemizi sağlayacak özellik seçimleri için 5 sinyal işleme tekniği kullanılmıştır. Bunlar hızlı Fourier dönüşümü (FFT), ayrık dalgacık dönüşümü (DWT), karesel ortalamaların karekökü (RMS), standart sapma (STD) ve ortalama (AVG) olarak belirlenmiştir. Buradaki amacımız, bu sinyal işleme tekniklerini kullanarak kritik özellikleri çıkarmaktır. FFT, verilerde en yaygın değer oluşumlarını ayıklamak ve bir hareketi temsil eden özellik çıkarımlarını seçmek için kullanılır. FFT sadece frekans bilgisi verir, ancak tüm zamansal verileri kaybederiz (Coşkun ve İstanbullu, 2012). DWT FFT gibi geçici verileri kaybetmez. Verilerimizin hem frekans hem de zaman bilgilerinin sezgisini verir. Bu olayın ne zaman gerçekleştiği bilgisiyle birlikte en sık karşılaşılan olaylar hakkında bize bir fikir vermektedir (Übeyli ve Güler, 2004). RMS, zaman serisi verilerinin değişim büyüklüğünü ölçerek tek bir değer verir, veri analizlerinde verilerin arasında daha iyi ayırım yapılmasına olanak sağlamaktadır. STD ve

AVG, zaman serisi verilerinin tek bir değere sıkıştırılabildiği ve bir özellik olarak kullanılabilirdiği için RMSye benzer.

Özellik alt kümesi seçimi ve özellik matrisi oluşturulması aşamaları öncesinde veri kümesi ön işlem aşamasından geçirilir. Her denek tarafından hazırlanan her bir sözcük örneğinin transpozu alınır. Transpozu alınan her bir sözcük örneği birleştirilerek öncelikle matrise sonra denek sayısına bağlı tekrar birleştirilerek bir veri kümesine dönüştürülerek kullanıma hazır edilir. Özellik alt kümesini seçmek için **Algoritma 1**'de sözde kodu verilen algoritma kullanılmıştır. Algoritmada elde edilen çıktılar, her teknik için özneliklerin bir alt kümesidir. Her bir özellik içeriğinin her bir teknik kullanılarak hesaplaması yapıldıktan sonra *ortalaması* alınarak birleştirilir. Algoritmada, hangi özellik setinin hareketleri daha iyi ayırt ettiğinin belirlenmesinde FFT ve DWT için korelasyon katsayısı kullanılmıştır. STD, RMS ve AVG teknikleri için ise, vektör çıktısı yerine skaler değer döndürülür. Her bir özellik için oluşturulan özellik vektörü için tüm kombinasyonlarının benzerlik ölçütü hesaplaması yapılarak toplanır. Belirlenen özellik çıkarımlarının her bir teknik için elde edilen sonuçlar içerisinde sıralama yapılarak ilk 15, 10 ve 5 özellik alt kümesi seçilerek özellik matrisi oluşturulması aşamasında kullanılmıştır.

**Algoritma 1.** Özellik alt kümesi seçim algoritması sözde kodu

---

```
fname='f1,f2,...f119'
numoffeature=119
For each technique (t=1:5)
    For each feature (f=1:119)
        For each gesture word (g=1:50)
            For each sample (s=1:160)
                r = numoffeature * s + f
                case 'FFT' : x1 = fft(content(r:end))
                case 'DWT' : x1 = dwt(content(r:end))
                case 'RMS' : x1 = rms(content(r:end))
                case 'STD' : x1 = std(content(r:end))
                case 'AVG' : x1 = avg(content(r:end))
            end
            x2=cat(1,x2,x1)
        end
        x3=mean(x2)
        x4 =cat(1,x4,x3)
    end
    [row,col] = size(x4)
    for each row ( i=1:x4-1)
        for each row+1 (j=i+1 : x4 )
            if FFT or DWT
                coff=corrcoef(x4(i:end),x4(j:end))
                sum=sum+coff(1,2)
            elseif RMS or STD or AVG
                mnmx=min(x4(i:end),x4(j:end)) / max(x4(i:end),x4(j:end))
                sum=sum+mnmx(1,2)
            end,end,end

            corrCoffSum(f)=sum
        end
    sortedCoffSum = sort(corrCoffSum,DESC)
    for n=1:15
        csvwrite (fname(find(corrCoffSum == sortedCoffSum(n))))
    end
end
```

---

İkinci aşamada ise öncelikle seçilen özellik alt kümesi setine **Algoritma 2**'de sözde kodu verilen algoritma uygulanarak tepe değerler kümesi belirlenir. Burada sözcük işaretlerinin seçilen özellik alt kümelerine ait dizilere FFT ve DWT teknikleri uygulanarak elde edilen yeni dizilere sıralama işlemi yapılır. Yapılan bu sıralama işlemi sonunda içinden ilk dokuz en büyük değer alınır. Diğer RMS, STD ve AVG teknikler uygulanarak ise sözcük işaretlerinin seçilen özellik alt kümelerine ait bir değer alınır. Bütün bu 5 teknik için elde edilen değerler birleştirilerek tepe değerler kümesi oluşturulur. Bu işlemler sonunda elde edilen değerlerin birleşimi sonucu, her hareket için aynı hareketin birden çok tekrarı ile oluşan satırları ve kullanılan teknikler tarafından çıkarılan farklı özellikleri belirten sütunları içeren bir özellik matrisi oluşturulur. Son aşama sonunda oluşturulan bu özellik matrisi veri kümeleri boyut indirgeme ve sınıflandırma aşamalarında kullanılır.

**Algoritma 2.** Tepe değerler kümesinin sözde kodu ve özellik matrisi algoritması

---

```

numberOfPeakValues=9; numoffeature=119
For each gesture word (g=1:50)
    For each technique (t=1:5)
        For each select feature (f=1:15)
            For each sample (s=1:160)
                r = numoffeature * s + f
                case FFT
                    y1=fft (content(r:end));y1=sort(y1, DESC)
                    y1=y1(:1,numberOfPeakValues)
                case DWT
                    y1=dwt (content(r:end));y1=sort(y1, DESC)
                    y1=y1(:1,numberOfPeakValues)
                case RMS :y1=rms(content(r:end))
                case STD :y1=std(content(r:end))
                case AVG :y1=avg(content(r:end))
            end
            y2=cat(1,y2,y1)
        end
        y3=cat(2,y3,y2)
    end
end
csvwrite(y3)
end

```

---

## 5.5. Histogram Tabanlı Özellik Çıkarımı

Histogram sinyalin içerisinde bulunan verilerin nasıl bir dağılıma sahip olduğunu gösterir. Bir histogram, ayrık bir olasılık yoğunluk fonksiyonu olarak da düşünülebilir. Histogram tabanlı özellik çıkarımı için öncelikle sinyalin histogramı elde edilir. Her histogram sütunu, belirli özellik vektörlerinin bir özelliğine karşılık gelir. Histogram elde edildikten sonra istenilen özellikler çıkarılır. Özellik histogramı, tüm farklı uzunluk ve sürelerle sahip sözcükler için elde edilen veri kümeleri boyutlarını eşitleyerek sınıflandırıcıya girişi kolaylaştırır ve aynı zamanda verilerin boyutunun küçültülmesini sağlar (Öz ve Leu, 2007) (Öz ve Leu, 2011).

Temel olarak, olası değerler aralığı alt aralıklara bölünür ve sonra her bir alt aralıktaki örneklerin sayısı sayılır.  $d$  aralığı bölmek istediğimiz bölümlerin sayısı olsun (sıra bilgisi) ve  $h_i$ ,  $0 \leq i \leq d$  ise, özellik vektörlerinin her biri için histogramın sıra bilgisini içerir. Aşağıdaki denklemler, bir özellik vektörü için bir histogramın nasıl hesaplanabileceğini gösterir:

$$h_i = \sum_{j=0}^n \frac{1}{n} r_i(x_j), \forall i, 0 \leq i \leq d \quad (1)$$

$$r_i(x_j) = \begin{cases} 1, & \text{if } \frac{i(x_{max} - x_{min})}{d} \leq x_j < \frac{(i+1)(x_{max} - x_{min})}{d} \\ 0, & \text{diğer} \end{cases} \quad (2)$$

Etkin bir şekilde, bu işlem yapılarak sıra ve sıra bileşenini normalleştirmiş oluyoruz. Birinci sıra bileşeni dizinin uzunluk bilgisidir. Yukarıdaki toplam ifadesi bir  $1/n$  terimi içerdiğinden ve her  $x_j$ 'nin tam olarak bir  $h_i$  sütununa düşmesi gerektiğinden, net etki şudur:

$$\sum_{i=0}^{d-1} h_i = 1 \quad (3)$$

İkinci sıra bileşeni, özellik vektör verilerinin aralığındadır. Sıra bölümleri özellik vektör verilerinin aralığına göredir; bu nedenle  $h_i$ ,  $0 \leq i \leq d$  aralığında değerler çoğu sıfır olmayacaktır. Bu istenen bir durumdur, çünkü bir işaretin boyutu ve küçük işaretlerde düşük çözünürlük sorunlarını ortadan kaldırarak birçok boş sıra bırakır. Aynı zamanda, işaretin kendisindeki bilgilerle hiçbir yerde bu kadar yakından ilişkili olmayacak mutlak konumlara sahip olduğunu gösterir. İşaretlere açı bilgisi yöntemi uygulanırsa işaretler -180 ile 180 arasındaki değerlere dönüşmektedir. Her açı değerinin frekansı bir açı örüntüsü olarak ele

alınmaktadır. Bu işlem yeni oluşan aç ı işaretlerine ait histogram öznitelik vektörü olarak ele alınmaktadır.

## 5.6. Zaman Tabanlı Özellik Çıkarımı

Literatürde işaret dillerinin sınıflandırılması için birçok özellik çıkarma yöntemi kullanılmıştır. Özellik çıkarma yöntemleri farklı problemler için farklı sınıflandırma başarımına sahip olabilmektedir. Etkili bir biçimde özellik çıkarma işaretlerin yüksek doğruluk oranıyla sınıflandırılması için son derece önemlidir. Bu nedenle hangi özellik çıkarma yönteminin hangi problem için uygun olduğuna karar vermek son derece güçtür. Ayrıca mevcut özellik çıkarma yöntemlerinin bazıları iyi sınıflandırma başarımına sahip olmasına rağmen çok karmaşık ve zaman alıcı olabilmektedir. Bu kapsamda zaman tabanlı özellik çıkarımları olan betimsel istatistik ölçütleri tercih edilmiştir (Çalışkan, 2019). Bu zamansal özellik çıkarımları tüm farklı uzunluk ve sürelerle sahip işaretler için elde edilen veri kümesi boyutlarını eşitleyerek sınıflandırıcıya girişi kolaylaştırır ve aynı zamanda verilerin boyutunun küçültülmesini sağlar. Zamansal özellik çıkarımı için kullanılan çok sayıda betimsel istatistik ölçütleri vardır. Bunlar içerisinde standart sapma, medyan, varyans, karesel ortalamaların karekökü ve ortalama olmak üzere 5 adet istatistiksel ölçüt kullanılmıştır.

### 5.6.1. Standart sapma (STD)

Varyans değerinin karekökü standart sapma değerini verir. Standart sapma genel olarak uzunluk, mesafe gibi niceliksel ölçekli sayılar için en çok kullanılan verilerin ortalamaya göre yayılımını gösteren istatistiksel bir ölçüdür. Eğer birçok veri ortalamaya yakın ise, standart sapma değeri küçüktür; eğer birçok veri ortalamadan uzakta yayılmışlarsa standart sapma değeri büyük olur. STD hesabı eşitlik (4)'de verilmiştir. Burada  $n$  alınan örneklem büyüklüğü sayısıdır.

$$STD = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (4)$$

### 5.6.2. Medyan (M)

Verilerin dağılımda uç noktalarının etkisini dikkate almak için medyan hesaplanır. Küçükten büyüğe sıraladıktan sonra tam ortada kalan değerdir. Veriler  $\{x_1, x_2, x_3, \dots, x_n\}$  ve  $n$  veri sayısı olmak üzere  $M$  eşitlik (5)'deki gibi hesaplanır.

$$M = \begin{cases} \frac{\frac{x_n}{2} + \frac{x_n}{2} + 1}{2}, & n \text{ çift ise} \\ \frac{x_n + 1}{2}, & n \text{ tek ise} \end{cases} \quad (5)$$

### 5.6.3. Varyans (V)

Bir veri kümesinin nasıl dağıldığının ölçüsü olarak tanımlanır. Varyans bir rassal değişken, bir olasılık dağılımı veya örneklem için istatistiksel yayılımın, mümkün bütün değerlerin beklenen değer veya ortalamadan uzaklıklarının karelerinin ortalaması şeklinde eşitlik (6)'da verildiği gibi bulunan bir ölçüdür. Düşük varyans, verilerin uyumlu olduğunun bir göstergesidir. Burada n alınan örneklem büyüklüğü sayısıdır.

$$V = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (6)$$

### 5.6.4. Karesel ortalamaların karekökü (RMS)

Karesel ortalamaların karekökü zaman içerisinde işaret uzunluklarının büyüklüğü olarak da ifade edilebilir. Değişen değerlerin büyüklüğünün ölçülmesinde kullanılan istatistik bir ölçüttür. Sürekli olarak değişen bir fonksiyonun sürekli olmayan değer serisi için hesaplanabilir. RMS değeri, eşitlik (7)'de verildiği gibi, anlık değerlerin karesi alınmış fonksiyonunun ortalama değerinin kareköküdür.

$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^n |x_i|^2} \quad (7)$$

### 5.6.5. Ortalama (AVG)

Ortalama bir sayısal veri dizisinin merkezsiz konumunu temsil etmek için seçilen tek bir sayı halinde gösterimidir. En çok kullanılan ortalama türü eşitlik (8)'de tanımlı aritmetik ortalamadır, bunun yanı sıra mod, medyan, geometrik ortalama ve diğer ortalama türleri de vardır.

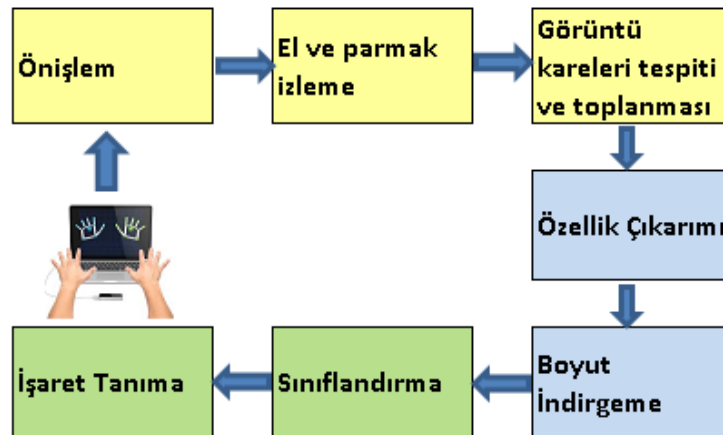
$$AVG = \frac{1}{n} \sum_{i=1}^n x_i \quad (8)$$

## 6. ALFABE VE SÖZCÜK TANIMA UYGULAMALARI

Bu çalışmada parmak alfabesi ve sözcük tanıma üzerine 4 farklı uygulama geliştirilmiştir. Birinci uygulamada LMC kullanılarak parmak alfabesi tanıma üzerine çalışma yapılmıştır. İkinci uygulama LMC tabanlı dinamik sözcük tanıma uygulamasıdır. Üçüncü uygulama Kinect cihazı kullanılarak geliştirilen dinamik sözcük tanıma çalışmasıdır. Dördüncü uygulama ise LMC ve Kinect cihazlarının birlikte kullanımı ile yapılan dinamik sözcük tanıma çalışmasıdır.

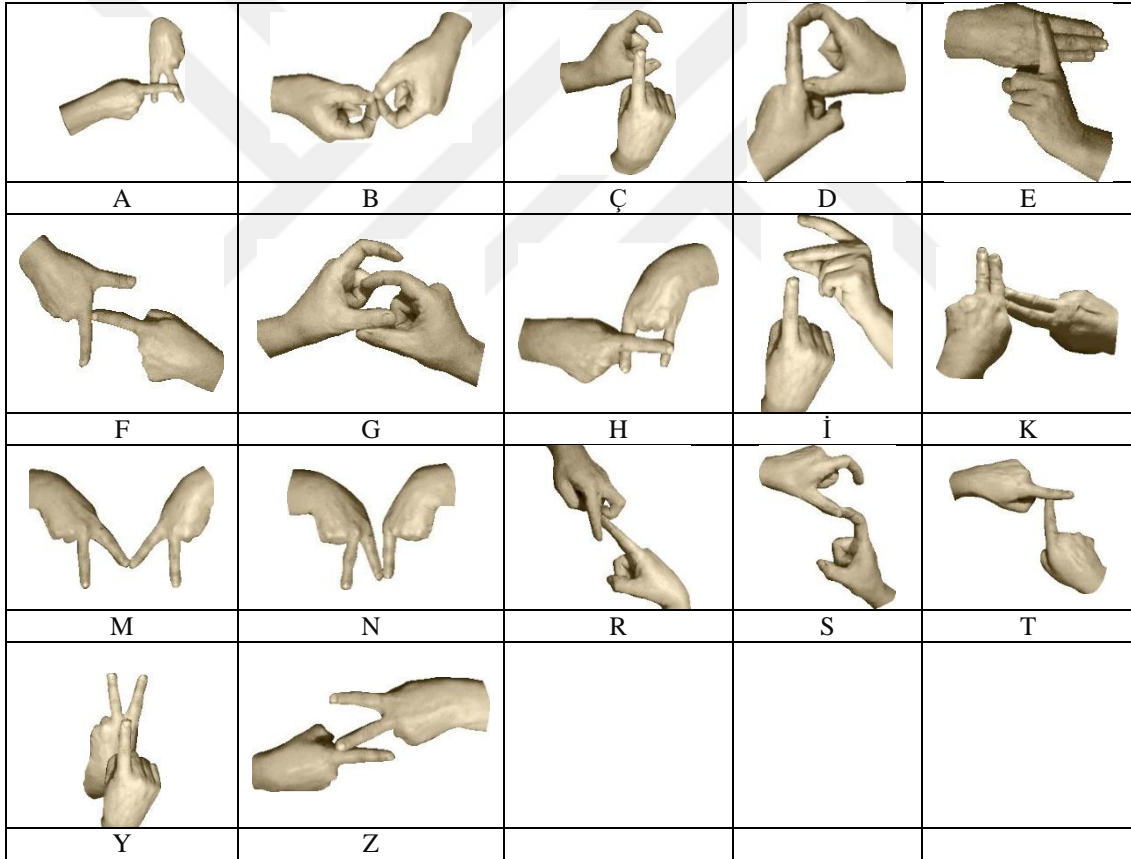
### 6.1. LMC Tabanlı Parmak Alfabeti Tanıma Uygulaması

Önerilen tanıma modelinin genel iş akışı, Şekil 6.1'de gösterildiği gibi 3 kısım ve 7 adımdan oluşmaktadır. Şekilde sarı renk ile gösterilmiş LMC kısmı ön işlem, izleme ve görüntü karelerinin tespiti ve toplanması aşamalarından oluşur. Mavi renk ile işaretli veri kümesi kısmı özellik çıkarımı ve boyut indirgeme adımlarından oluşur. Uygulama kısmı ise eğitim ve test aşamasından oluşmaktadır. İlk adım, LMC hizmetini başlatan ön işlem adımındır. İkinci adım, izleme adımındır, bu adımda izleme katmanı, el ve parmakların izleme bilgilerini çıkararak verileri eşleştirir. Üçüncü adım, LMC API özelliklerini kullanarak eşleştirilen verilerden görüntü karelerinin tespiti ve toplanmasıdır. Dördüncü adım, LMC cihazından toplanan görüntü karelerinden belirlenen özellikler çıkarılarak veri kümesi oluşturulur. Beşinci adımda ise özelliklerin indirgeme yöntemiyle boyutları düşürülür. Altıncı adım ise bu özelliklerin sınıflandırıcıya bir vektör olarak sunulduğu ve harflerin tanımlanması için eğitimin yapıldığı aşamadır. Sınıflandırma adımında makine öğrenmesinde kullanılan geleneksel sınıflandırma algoritmaları kullanılmıştır. Son adımda eğitilmiş sınıflandırıcı kullanıcıların hareketini tanımak için kullanılır.



Şekil 6.1. Statik parmak alfabesi için önerilen tanıma modelinin iş akışı

TİD alfabesinde çift elle gerçekleştirilen 22 harften 17'si statik hareketlerden oluşmaktadır. Bu çalışmada, öncelikle TİD'de çift elle gerçekleştirilen 17 statik parmak alfabesi için biri yetişkin erkek, diğeri yetişkin kadın olmak üzere 2 denek kullanılarak veri toplanmış ve özellik çıkarımları yapılmıştır. Belirlenen özellik çıkarımları ile birlikte her bir harf için 54 farklı yön ve her bir yön için tek çerçeveli (frame) 2 örnek içeren 2 denekten toplam 3672 kayıtlı bir veri kümesi oluşturulmuştur. Bu uygulamamızda cihazın donanımsal yapısından dolayı cihazın avuç içi hız özelliği durağan durumda ayarlanmış iken belirlenen değer aralığında cihazdan veriler alınmıştır. Harfler için çoğu ihtimal göz önüne alınarak belirlenen 54 farklı konum şekli her harf için uygulanarak harfler arası farkın oluşturulması sağlanmıştır. Kullanılan çift elle gerçekleştirilen 17 statik parmak alfabesi Şekil 6.2'de sunulmuştur.



Şekil 6.2. Çift elle gerçekleştirilen 17 statik TİD parmak alfabesi

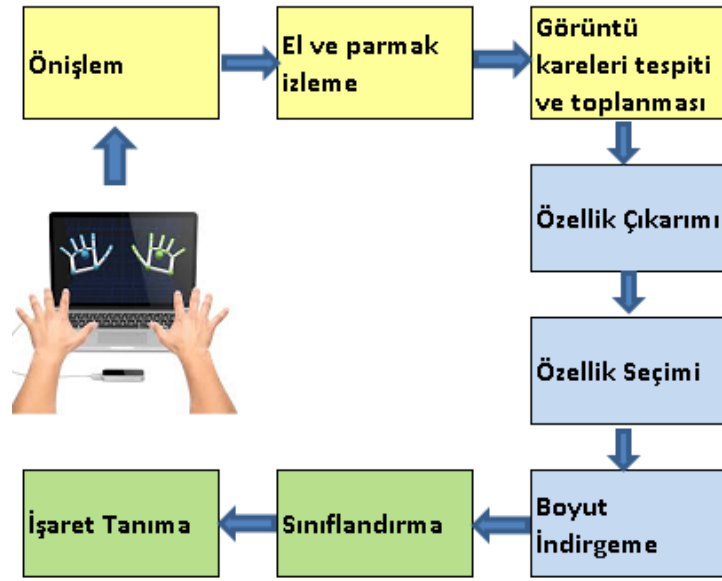
Bu çalışma için oluşturulan veri kümesine öncelikle boyut indirgeme yöntemleri uygulanarak veri kümesi boyutu küçültülmüştür. Geleneksel tabanlı algoritmalar kullanılarak orijinal veri kümesi ile boyutu indirgenmiş veri kümelerinin sınıflandırma başarımları karşılaştırılmıştır. LMC parmak alfabesi tanıma uygulaması esnasında bir denek tarafından yapılan çekimlerden örnek gösterimi Şekil 6.3'de sunulmuştur.



Şekil 6.3. LMC parmak alfabesi tanıma uygulaması örnek gösterimi

## 6.2. LMC Tabanlı Dinamik Sözcük Tanıma Uygulaması

Önerilen bu dinamik sözcük tanıma modelin genel iş akışı, Şekil 6.4'de gösterildiği gibi 3 kısım ve 8 adımdan oluşmaktadır. LMC tabanlı parmak alfabesi tanıma uygulamasından farklı olarak özellik seçimi algoritması aşaması eklenmiş ve kullanılan algoritmalar değiştirilmiştir. Şekilde sarı renk ile gösterilmiş LMC kısmı önileme, izleme ve görüntü karelerinin tespiti ve toplanması adımlarından oluşur. Mavi renk ile işaretli veri kümesi kısmı özellik çıkarımı, özellik seçimi ve boyut indirgeme adımlarından oluşur. Uygulama kısmı ise eğitim ve test aşamasından oluşmaktadır. Birinci uygulamadan farklı olarak beşinci adımda, çıkarılan özelliklere özellik seçim algoritması uygulanarak özellik seçimi yapılır. Sınıflandırma adımında ise makine öğrenmesinde yer alan aşırı öğrenme (ELM) tabanlı sınıflandırıcılar kullanılmıştır.



**Şekil 6.4.** LMC tabanlı dinamik sözcükler için önerilen tanıma modelinin iş akışı

Bu çalışma kapsamında, 50 dinamik sözcük kendi arasında benzerlik ve farklılıkları dikkate alınarak çalışmada kullanılmak üzere hazırlanmıştır. Bu sözcükler, 20 farklı kategori altında ve yapılışı itibari ile birbirine benzeyen sözcüklerden oluşmaktadır. Sözcük seçiminde yapılış süreleri açısından birbirine yakın sürede yapılan sözcükler tercih edilmiştir. Çalışmamızda kullanılan sözcükler Tablo 6.1’de sunulmuştur. Yapılan bu seçim ve çalışmalar ile sözcüklerin sınıflandırma başarımlarının oranı doğruluğunun ve güvenilirliğinin artırılması hedeflenmiştir.

**Tablo 6.1.** Yapılışı itibarıyla birbirine benzer dinamik sözcükler

	Sözcükler		Sözcükler
1	Balta, patlıcan, yarım	11	Suç, yumurta
2	Açık, baklava, yarı final	12	Felsefe, trafik
3	Farklı, final, sınav, takım	13	Matkap, pergel, tornavida
4	Fırın, gemi	14	Çatal, fiş
5	Avustralya, Avusturya	15	Kibar, kibrit
6	Bıçak, çabuk	16	Akraba, dama
7	Damga, köfte	17	Sosis, tel
8	Muhasebe, terzi	18	Çırak, eş
9	Cüzdan, incir	19	Pasta, serseri, silecek, yönetmek
10	Kutu, masa, rekor	20	Coğrafya, deprem, kilim, yağmur

Bu çalışmada, öncelikle TİD’de çift elle gerçekleştirilen 50 dinamik sözcük için iki yetişkin erkek, iki yetişkin kadın olmak üzere 4 denek kullanılarak veri toplanmış ve özellik

çıkarımları yapılmıştır. Her bir sözcük için tek çerçeveli (frame) 40 örnek içeren 4 denekten toplam 8000 kayıttan oluşan bir veri kümesi oluşturulmuştur. Veri kümesi, her bir sözcük için 12 saniye süre içerisinde alınan 480 adet görüntü karelerinin birleşiminden oluşmuştur. Kullanılan iki dinamik benzer sözcüğün yapılaş görüntü dizisi Şekil 6.5’de sunulmuştur.



Şekil 6.5. "Fırın" (üst sıra) ve "Gemi" (alt sıra) sözcüklerin yapılaş görüntü dizileri (TID)

Özellik seçimi algoritması uygulama aşaması öncesinde veri kümesi ön işlem aşamasından geçirilir. Her denek tarafından hazırlanan her bir sözcük örneğinin ( $12 \times 119$ ) transpozu alınır. Transpozu alınan her bir sözcük örnekleri birleştirilerek ( $4760 \times 12$ ) boyutlu matrise dönüştürülür. 4 denek için ( $19040 \times 12$ ) boyutlu bir veri kümesine dönüştürülerek kullanıma hazır hale getirilir. Özellik alt kümesini seçmek için **Algoritma 1**'de sözde kodu verilen birinci algoritma kullanılmıştır. Algoritmadan elde edilen çıktılar, her teknik için özneliklerin bir alt kümesidir. Öncelikle her bir özellik için her bir teknik kullanılarak 4 deneğin 40 örneği için ( $160 \times 12$ ) hesaplama yapılır. Sonra ortalaması ( $1 \times 12$ ) alınarak 50 sözcük için ( $50 \times 12$ ) matrisler birleştirilir. Algoritmada, hangi özellik setinin hareketleri daha iyi ayırt ettiğinin belirlenmesinde FFT ve DWT için korelasyon katsayısı kullanılmıştır. STD, RMS ve AVG teknikleri için ise, vektör çıktısı yerine skaler değer döndürülür. Her bir özellik için oluşturulan özellik vektörünün ( $50 \times 12$ ) tüm kombinasyonlarının benzerlik ölçütü hesaplaması yapılarak toplanır. 119 özellik için edilen sonuçlar içerisinde sıralama yapılarak ilk 15, 10 ve 5 özellik alt kümesi seçilerek özellik matrisi oluşturulması aşamasında kullanılmıştır.

İkinci aşamada ise Algoritma 1 ile seçilen özellik alt kümesi setine **Algoritma 2**'de sözde kodu verilen ikinci algoritma uygulanarak en büyük değerler kümesi belirlenir. Algoritmada FFT ve DWT teknikleri için seçilen özellik alt kümelerinin her birinden yapılan sıralama sonunda dokuz en büyük değer, diğer teknikler için ise bir değer belirlenir. Çıktı, her

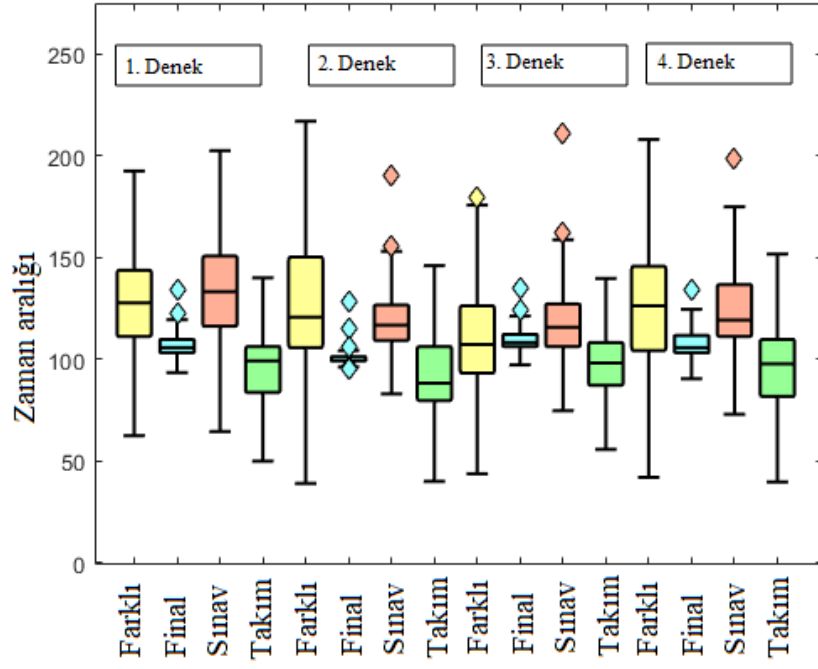
hareket için aynı hareketin birden çok eylemini belirten satırlar ve kullanılan teknikler tarafından çıkarılan farklı özellikleri belirten sütunlar içeren bir özellik matrisidir. 15 özellik alt kümesi seçimi için en büyük değerler kümesi ve özellik matrisi Tablo 6.2'de sunulmuştur. Oluşturulan özellik matrisi veri kümeleri boyut indirgeme ve sınıflandırma aşamalarında kullanılır.

**Tablo 6.2.** 15 özellik alt kümesi seçimi için tepe değeri ve özellik matrisi sayısı

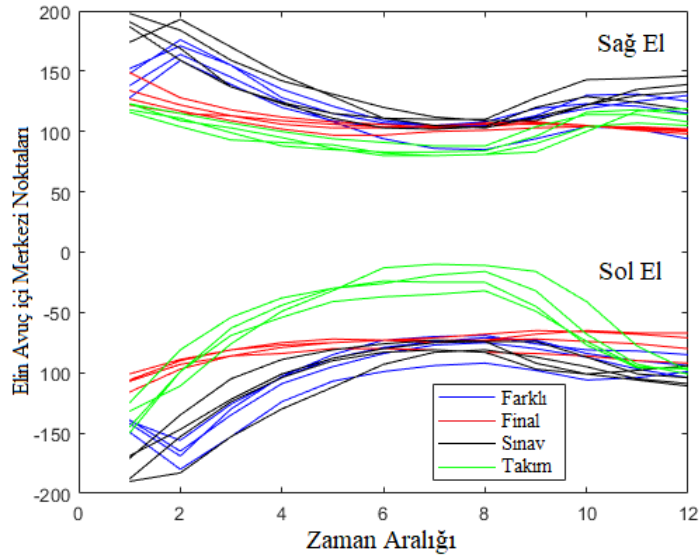
Teknik	Satır sayısı		Sütun sayısı		Özellik matrisi
	Denek sayısı	Örnek sayısı	Tepe değer sayısı	Özellik sayısı	
FFT	4	40	9	15	160 × 135
DWT	4	40	9	15	160 × 135
RMS	4	40	1	15	160 × 15
STD	4	40	1	15	160 × 15
AVG	4	40	1	15	160 × 15
Toplam özellik seçimi ve matris sayısı				75	160 × 315

Bu çalışma için oluşturulan orijinal veri kümesi öncelikle özellik seçim algoritması uygulanarak özellik sayısı kademeli olarak düşürülmüştür. İkinci aşamada ise boyut indirgeme yöntemleri kullanılarak veri kümesi boyutu ikinci kez küçültülmüştür. ELM öğrenmesi yöntemleri kullanılarak orijinal veri kümesi ile boyutu indirgenmiş veri kümelerinin sınıflandırma başarımları karşılaştırılmıştır.

Verilerdeki büyük çeşitlilik, kutu grafiği (Box-Plot) olarak bilinen istatistiksel bir analiz tekniği kullanılarak ölçülür. Verilerin dağılım düzeyini, yayılmasını ve simetrisini iletmek için medyan, ilk/üçüncü çeyrek, en düşük/en yüksek olmak üzere 5 veri noktasını kullanır. 4 denek için 4 benzer sözcüğünün el hareketleri için örnek bir kutu grafiği gösterimi Şekil 6.6'da sunulmuştur. Bu hareketler LMC kullanılarak çekilmiştir. Ek olarak, bir işaret hareketi dizisi havada gerçekleştirildiğinden, 4 denek tarafından yapıldığında benzer sözcüklerin el hareketlerindeki varyasyonu gösteren 2D çizim kullanılarak görselleştirilebilir. Şekil 6.6, tüm dizi boyunca avuç içi merkez noktaları (Palm Position) çizilerek elde edilmiştir. Çizimdeki her renk bir denekten elde edilen değişimi diğerlerinden ayırt edilmesi için kullanılmıştır. 4 deneğe ait 4 benzer sözcüğün el hareketleri için örnek bir 2D çizim Şekil 6.7'de sunulmuştur.



Şekil 6.6. Benzer sözcüklerin işaret hareketleri için kutu grafiği.

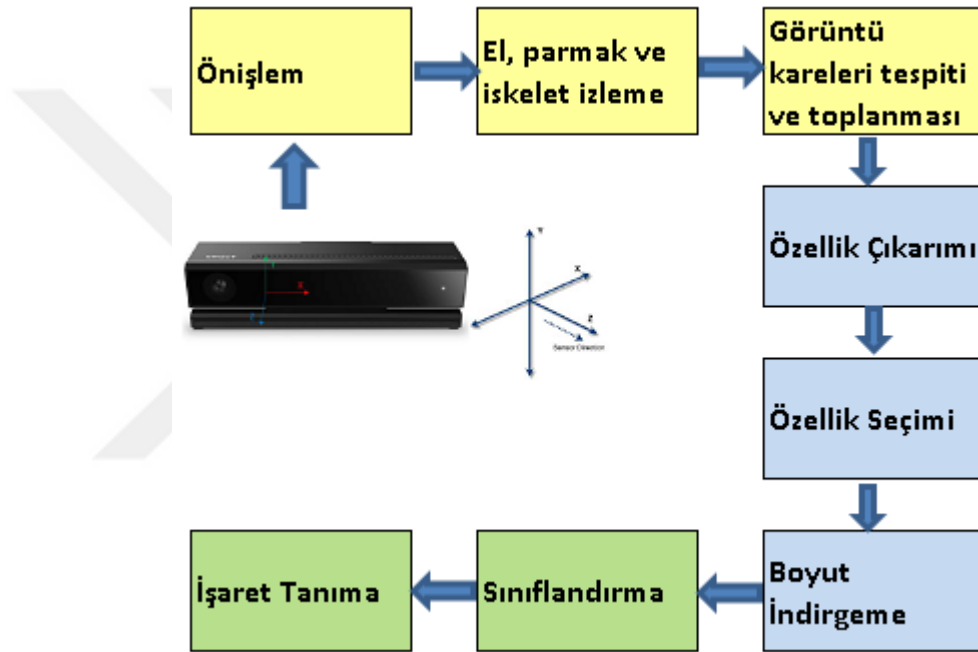


Şekil 6.7. Benzer sözcüklerin işaret hareketleri için 2D çizim.

### 6.3. Kinect Tabanlı Dinamik Sözcük Tanıma Uygulaması

Önerilen Kinect tabanlı dinamik sözcük tanıma modelinin genel iş akışı, Şekil 6.8'de gösterildiği gibi 3 kısım ve 8 adımdan oluşmaktadır. Şekilde sarı renk ile gösterilmiş Kinect kısmı önışlem, izleme ve görüntü karelerinin tespiti ve toplanması adımlarından oluşur. Mavi renk ile işaretli veri kümesi kısmı özellik çıkarımı, özellik seçimi ve boyut indirgeme

adımlarından oluşur. Uygulama kısmı ise eğitim ve test aşamasından oluşmaktadır. İlk adım, Kinect hizmetini başlatan ön işleme aşamasıdır. İkinci adım, izleme aşamasıdır, bu aşamada izleme katmanı, iskelet yapısı izleme bilgilerini çıkararak verileri eşleştirir. Üçüncü adım, Kinect *skeleton* kütüphanesi özelliklerini kullanarak eşleştirilen verilerden görüntü karelerinin toplanmasıdır. Dördüncü adımda, Kinect cihazından toplanan görüntü karelerinden belirlenen özellikler çıkarılarak veri kümesi oluşturulur. Beşinci ve sonraki adımlar ise LMC tabanlı dinamik sözcük tanıma uygulamasındaki adımları ile aynı şekilde yürütülmüştür. Sınıflandırma adımında ise makine öğrenmesinde yer alan sinir ağı tabanlı ve aşırı öğrenme tabanlı (ELM ) algoritmalar kullanılmıştır.



**Şekil 6.8.** Kinect tabanlı dinamik sözcükler için önerilen tanıma modelinin iş akışı

Bu çalışma kapsamında, 20 dinamik sözcük kullanılmıştır. Çalışmamızda kullanılan sözcükler Tablo 6.3’de sunulmuştur. Sözcük seçiminde yapılış süreleri açısından birbirine yakın sürede yapılan sözcükler tercih edilmiştir.

**Tablo 6.3.** Kinect cihazı için kullanılan dinamik sözcükler

Sözcükler	
Ağaç	Gündüz
Akşam	Hademe
Asistan	Hemşire
Aslan	Kahraman
Bluz	Kapalı
Boğa	Kıble
Bulut	Mutlu
Çapraz	Tavşan
Elbise	Timsah
Gözlük	Uzun

Bu çalışmada, öncelikle TİD’de çift elle gerçekleştirilen 20 dinamik sözcük için 1 denek kullanılarak veri toplanmış ve özellik çıkarımları yapılmıştır. Bu kapsamda iskelet yapısına ait kol, el ve parmak noktalarının yön, konum, açı, durum gibi özelliklerden ayırt edici 120 özellik belirlenmiştir. Bu özellikler arasında sadece iskelet boyutuna bağlı kalmadan benzer sözcüklerin tanınmasında ayırt edici olacağı düşünülerek açı bilgisi de vardır. Her bir sözcük için tek çerçeveli (frame) 25 örnek içeren 1 denekten toplam 500 kayıttan oluşan bir veri kümesi oluşturulmuştur. Veri kümesi, her bir sözcük için 10 saniye süre içerisinde alınan 250 adet görüntü karelerinin birleşiminden oluşmuştur. Kinect dinamik sözcük tanıma uygulaması esnasında bir denek tarafından yapılan çekimlerden örnek bir gösterim Şekil 6.9’de sunulmuştur.

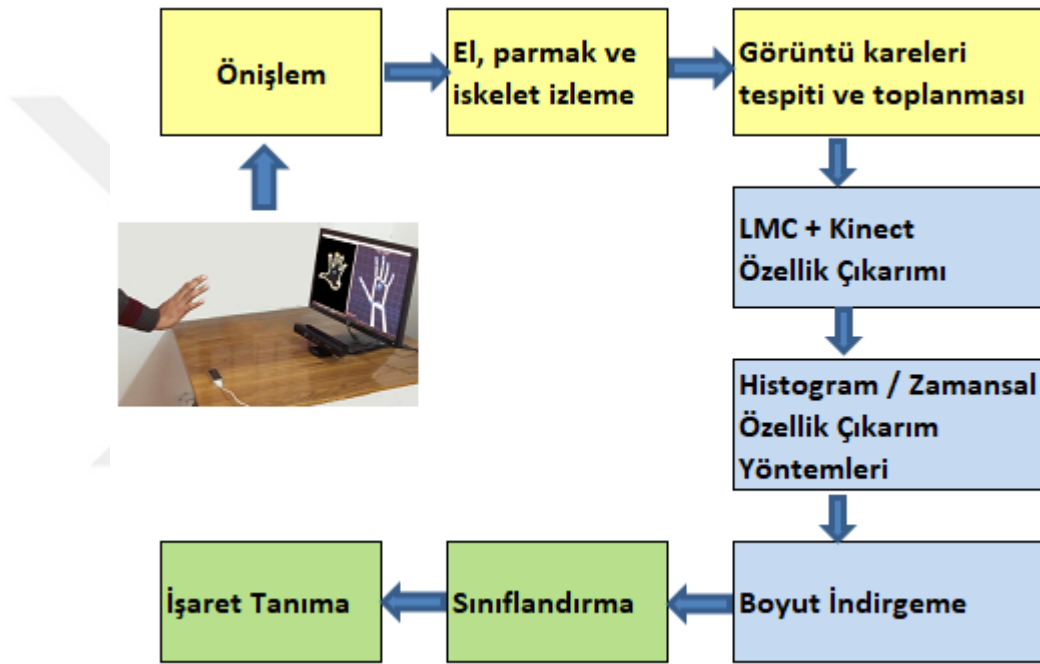


Şekil 6.9. Kinect dinamik sözcük uygulaması örnek gösterimi

#### 6.4. LMC + Kinect Tabanlı Dinamik Sözcük Tanıma Uygulaması

Önerilen modelin genel iş akışı, Şekil 6.10'da gösterildiği gibi 3 kısım ve 8 adımdan oluşmaktadır. Şekilde sarı renk ile gösterilmiş LMC ve Kinect kısmı önileme, izleme ve görüntü karelerinin tespiti ve toplanması aşamalarından oluşur. Mavi renk ile işaretli veri kümesi kısmı cihazlardan özellik çıkarımı, özellik çıkarım yöntemleri ve boyut indirgeme adımlarından oluşur. Uygulama kısmı ise eğitim ve test aşamasından oluşmaktadır. İlk adım LMC ve Kinect hizmetini başlatan ön işleme aşamasıdır. İkinci adım, izleme aşamasıdır. Bu aşamada izleme katmanı, LMC cihazı el ve parmak izleme bilgilerini ve Kinect cihazı iskelet yapısı izleme bilgilerini çıkararak verileri eşleştirir. Üçüncü adım, LMC SDK kütüphanesi ve Kinect *skeleton* kütüphanesi özellikleri kullanarak eşleştirilen verilerden görüntü karelerinin toplanmasıdır. Dördüncü adım, LMC ve Kinect cihazından toplanan görüntü karelerinden belirlenen özellikler çıkarılarak veri kümesi oluşturulur. Beşinci adımda, önceki iki

uygulamada kullanılan özellik seçimi algoritması kullanılmamıştır. Bu algoritma yerine iki yeni özellik çıkarım yöntemi kullanılmıştır. Bu adımda farklı uzunluk ve sürelerle sahip özellik kümeleri histogram ve zamansal özellik çıkarım yöntemleri uygulanarak uzunlukları ve girişleri eşitlenir. Altıncı adımda ise özellik çıkarımı ve uzunlukları eşitlenen özelliklerin indirgeme yöntemiyle boyutları düşürülür. Yedinci adım ise bu özelliklerin sınıflandırıcıya bir vektör olarak sunulduğu ve sözcüklerin tanımlanması için eğitimin yapıldığı aşamadır. Son adımda eğitilmiş sınıflandırıcı kullanıcıların hareketini tanımak için kullanılır. Sınıflandırma adımında ise makine öğrenmesinde yer alan aşırı öğrenme (ELM) tabanlı sınıflandırıcılar kullanılmıştır.



Şekil 6.10. LMC+Kinect tabanlı dinamik sözcükler için önerilen tanıma modelinin iş akışı

Bu çalışma kapsamında, 21 dinamik sözcük kullanılmıştır. Çalışmamızda kullanılan dinamik sözcükler Tablo 6.4'de sunulmuştur. Diğer uygulamalardan farklı olarak yapılış süreleri açısından birbirinden farklı uzunluklara sahip sözcükler tercih edilmiştir. Aynı zamanda sözcüklerin yapılışları yavaş, orta ve hızlı olmak üzere farklı sürelerde yapılarak sistemin doğruluğunun artırılması hedeflenmiştir.

**Tablo 6.4.** LMC ve Kinect cihazı için kullanılan dinamik sözcükler

Sözcükler	
Ağaç	El
Akşam	Ev
Asistan	Küçük
Aslan	Lastik
Bilgisayar	Satranç
Börek	Sürücü
Bulut	Tablo
Büyük	Tahlil
Çatı	Uygun
Ekmek	Uzak
	Vazo

Bu çalışmada, öncelikle TİD’de çift elle gerçekleştirilen 21 dinamik sözcük için 2 denek kullanılarak veri toplanmış ve özellik çıkarımları yapılmıştır. LMC kullanılarak dirsek, el, parmak noktaları için yön, uzunluk, mesafe, konum, açı, durum gibi özelliklerden ayırt edici 110 özellik belirlenmiştir. Kinect kullanılarak iskelet yapısına ait 16 eklem ve eklem noktalarına ait yön, konum ve açı özelliklerden ayırt edici 120 özellik belirlenmiştir. Bu özellikler arasında benzer harflerin tanınmasında ayırt edici olacağı düşünülerek açı bilgileri de vardır. Kinect için belirlenen özellik çıkarımları için kullanılan iskelet eklem noktalarına ve açı bilgilerine ait gösterim Şekil 5.2’de kırmızı ve sarı renk ile imlenerek sunulmuştur. LMC ve Kinect için belirlenen özellik çıkarımları birleştirilerek 5 kategori ve 15 başlık altında gruplandırılmış özelliklerin ölçü birimi, türü, isimleri ve adetleri Tablo 6.5’de sunulmuştur. LMC ve Kinect cihazlarından toplam olarak 230 özellik çıkarımı belirlenmiştir. Her bir sözcük için tek çerçeveli (frame) 20 örnek içeren 2 denekten toplam 840 kayıttan oluşan bir veri kümesi oluşturulmuştur. Veri kümesi, farklı uzunluklara sahip sözcüklerin farklı süreler içerisinde alınan farklı görüntü karelerinin birleşiminden oluşmuştur.

**Tablo 6.5.** Belirlenen özelliklerin cihaz, ölçü birimi, türü, isim ve adetleri

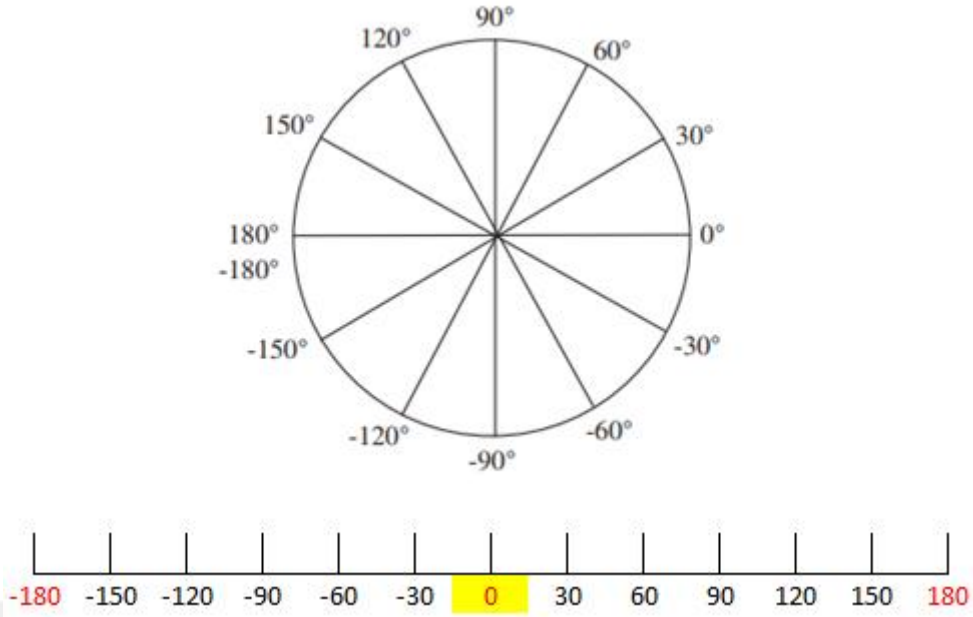
LMC+Kinect (Özellik Çıkarımı)				
Cihaz	Ölçü	Özellik		Adet
LMC	Distance	Fingers.TipPosition to PalmPosition	10X1	10
	Angle	Fingers.Direction to Fingers.Direction	8X1	8
		Fingers.Direction to Hands.Direction	10X1	10
		Fingers.Direction to PalmNormal	10X1	10
		Hands.Direction.Pitch,Yaw,Roll	2X3	6
		Fingers.TipPosition(X,Y,Z)	10X3	30
	Vector	Hands.Direction.PalmPositions(X,Y,Z)	2X3	6
		Hands.Direction.PalmNormal(X,Y,Z)	2X3	6
		Hands.Direction.Direction(X,Y,Z)	2X3	6
		Arm.ElbowPosition(X,Y,Z)	2X3	6
Arm.WristPosition(X,Y,Z)		2X3	6	
Arm.Direction(X,Y,Z)		2X3	6	
Kinect	Vector	Joints[JointType].Position(X,Y,Z)	16x3	48
		JointOrientations[JointType].Orientation(X,Y,Z,W)	16x4	64
	Angle	Joints([JointType1],[JointType2],[JointType3]).Position	2x4	8
Toplam				230

Bu uygulama kapsamında farklı uzunluk ve sürelerle sahip dinamik sözcükler için oluşturulan veri kümelerinin boyutlarının küçültülmesi ve sınıflandırıcılara veri girişlerinin eşitlenmesinin sağlanmasında 2 tür özellik çıkarımı yöntemi uygulanmıştır. Bunlar histogram ve zamansal tabanlı özellik çıkarımlarıdır. Histogram ve yeni veri kümesi oluşturma işlemleri 5 aşamadan oluşmaktadır. Birinci aşama özellik çıkarımlarının özellik ve türlerine göre gruplandırılmasıdır. Öncelikle LMC ve Kinect için belirlenen 230 adet özellik çıkarımı kendi içinde gruplandırılmıştır. Gruplandırma işlemi sırasında LMC ve Kinect cihazı özellik çıkarımları ayrı ayrı değerlendirilmiştir. Bu aşamada her cihaz için aynı özellik ya da benzer özellik taşıyan özellik çıkarımları ölçü birimlerine göre ayrılarak gruplandırılmıştır. Her iki cihaza ait toplamda 12 ayrı özellik çıkarım grubu oluşturulmuştur. İkinci aşamada her özellik ya da benzer özellik taşıyan özellik çıkarımları özelliklerin yön, mesafe, konum, durum, açı gibi çeşidine göre ve geriye döndürdüğü değer ölçü birimine göre belirli bir aralık içerisinde değerler üretmektedir. Belirlenen 12 özellik çıkarım grubunun her biri için üretilen değerlerin aralığı (aralık kümesi) tespit edilmiştir. Örneğin bu özellik çıkarım grubundan iskelet eklem açıları açı özelliğine sahip olması ve ölçü birimine göre -180 ile 180 arasında değerler üretmektedir. Üçüncü aşamada ise her bir özellik çıkarım grubunun aralık kümesi büyüklüğüne göre eşit uzunluklarda alt aralık sayıları belirlenmiştir. Özellik çıkarım grubundan iskelet eklem açı bilgisine ait eşit aralığa bölünmüş aralık kümesi bölünmüş gösterimi Şekil 6.11’de sunulmuştur. Her bir özellik çıkarım grubunun özelliğin türe ve ölçü birimine göre ürettiği aralık kümesinin büyüklüğüne göre aralık sayısı farklı belirlenmiştir. Histogram için belirlenen 5 kategori ve 12 başlık altında gruplandırılan özellik çıkarımlarına ait bilgiler ve alt aralık sayıları Tablo 6.7’de sunulmuştur. Dördüncü aşamada aralık kümesi

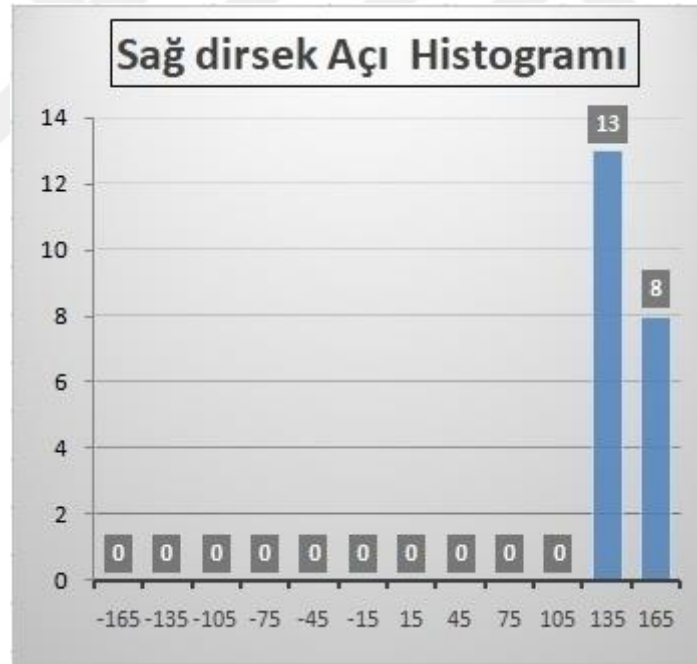
ve alt aralık sayısı belirlenen histogram için her bir işaret örneğine ait veri girişleri alınarak histogram grafiği üzerinde uygun alt aralıklara dağıtımları yapılır. **Börek** sözcüğüne ait sağ dirsek iskelet eklem açısı bilgisine ait aralık kümesine dağıtım yapılan veri dizisinin dağılımları sonucu elde edilen alt aralıklara ait değerler ve histogram grafiği Şekil 6.12’de sunulmuştur. Bu histogram grafiğinin hazırlanmasında Tablo 5.6’da veri değerleri sunulan **börek** sözcüğünün sağ dirsek <Neck-ShoulderRight-ElbowRight> iskelet eklem açısı bilgisi içeren veri dizisi kullanılmıştır. Son aşamada histogram grafiği üzerinde her bir alt aralığa dağıtım yapılan değerler sayılır ve veri girişi uzunluğuna bölünerek her bir alt aralığa ait son değerler bulunur. Her bir aralık için çıkan sonuçlar sıfır ile bir aralığında değerler olup toplamı bir eşiğ olur. Bu işlem ile aynı zamanda veri kümesi üzerinde normalleştirme yapılmış olur. LMC ve Kinect cihazları için belirlenen toplam 230 özelliğe ait 12 özellik çıkarım grubunun her biri için bu işlemler yapılarak elde edilen değerler her bir örnek için ayrı ayrı birleştirilerek yeni veri kümesi oluşturulur. Her bir işaret örneğine ait farklı uzunluklara sahip veri girişleri olsa bile çıkan değerler belirlenen aralık sayısı kadar olacaktır. Histogram özellik çıkarımı sayesinde veri kümesi boyutunun eşitlenmesi, küçültülmesi ve aynı zamanda normalleştirilmesi gerçekleştirilmiş olur. Oluşturulan bu yeni veri kümesi boyut indirgeme ve sınıflandırma aşamasında kullanılır.

**Tablo 6.7.** Histogram için belirlenen özellik çıkarımlarına ait bilgiler ve aralık sayıları

Cihaz	Ölçü	Özellikler	aralıkXözellik	Adet
Leap Motion	Distance	Fingers.TipPosition to PalmPosition	6X10	60
		Fingers.Direction to Fingers.Direction	5X8	40
	Angle	Fingers.Direction to Hands.Direction	7X20	140
		Fingers.Direction to Hands.PalmNormal		
		Hands.Direction.Pitch,Yaw,Roll	7X6	
	Vector	Fingers.TipPosition(X,Y,Z)	11X30	330
		Hands.Direction.PalmPositions(X,Y,Z)	9X6	54
		Hands.Direction.PalmNormal(X,Y,Z)	5X12	60
		Hands.Direction.Direction(X,Y,Z)		
		Arm.ElbowPosition(X,Y,Z)	10X12	120
Arm.WristPosition(X,Y,Z)				
Arm.Direction(X,Y,Z)	5X6	30		
Kinect	Vector	Joints[JointType].Position(X,Y,Z)	7X48	336
		JointOrientations[JointType].Orientation(X,Y,Z,W)	6X44	264
	Angle	Joints([JointType1],[JointType2],[JointType3]).Position	12X8	96
Toplam				1572



Şekil 6.11. Açık bilgisine ait aralık kümesinin 12 eşit aralığa bölünmüş gösterimi



Şekil 6.12. Börek sözcüğüne ait sağ dirsek açı histogramı

Bu uygulama aşamasında kullanılan ikinci özellik çıkarımı yöntemi zamansal tabanlı özellik çıkarımıdır. Zamansal özellik çıkarımlarından bilinen ve en sık kullanılan standart sapma, medyan, varyans, karesel ortalamaların karekökü ve ortalama olmak üzere 5 adet istatistiksel ölçüt tercih edilmiştir. İlk aşamada her bir işaret örneğinin özellik çıkarımına bu istatistiksel ölçütler ayrı ayrı uygulanarak geriye her biri için birer değer üretilir. Her bir işaret veri

kümesi örneğine ait farklı uzunluklara sahip veri girişleri olsa bile çıkın değerler kullanılan istatistiksel ölçüt sayısı kadar olacaktır. Histogram özellik çıkarımına benzer şekilde farklı uzunluklara sahip girişler olsa bile üretilen sonuçlar istenilen aralık ya da istatistiksel ölçüt sayısı kadar olacaktır. İkinci aşamada ise LMC ve Kinect cihazları için belirlenen toplam 230 özellik çıkarımına her bir istatistiksel ölçüt için bu işlem yapılarak elde edilen 5 adet değer her bir örnek için ayrı ayrı birleştirilerek yeni veri kümesi oluşturulur. Aynı zamanda bu işlem sayesinde farklı uzunluklara sahip veri kümesi boyutunun eşitlenmesi ve küçültülmesi gerçekleştirilmiş olur. Oluşturulan bu yeni veri kümesi boyut indirgeme ve sınıflandırma aşamasında kullanılır. LMC+Kinect dinamik sözcük tanıma uygulaması esnasında bir denek tarafından yapılan çekimlerden örnek gösterim Şekil 6.13’de sunulmuştur.



Şekil 6.13. LMC+Kinect dinamik sözcük uygulaması örnek bir gösterimi

## 7. SINIFLANDIRICILAR VE BOYUT İNDİRGEME YÖNTEMLERİ

Makine öğrenmesinde kullanılan sınıflandırıcılar, otomatik karar verme sistemi için çok önemli bir yere sahiptir. Gelen verinin hangi sınıfa ait olduğunu belirlememizi sağlarlar. Sınıflandırma işlemini yapan birçok algoritma vardır. Bunlar, farklı veri kümeleri için farklı sonuçlar verebilir. Bunun için, elimizdeki veriye en uygun sınıflandırıcıyı kullanmamız, sağlıklı karar vermek için önemlidir. Makine öğrenmesinde kullanılan sınıflandırma algoritmaları 3 kategoride ele alınmıştır. Geleneksel olarak kullanılan genel algoritmalar, sinir ağı tabanlı algoritmalar ve aşırı öğrenme tabanlı algoritmalar olarak kategorize edilmiştir. Sınıflandırıcı aşaması öncesi veri kümesi işlemleri aşamasında kullanılan PCA, LDA boyut indirgeme yöntemleri ele alınmıştır.

Geleneksel sınıflandırıcılar K en yakın komşu algoritması (k-NN), Naive Bayes (NB), destek vektör makinesi (SVM), karar ağaçları (DT) ve rastgele orman (RF) olmak üzere ele alınmıştır. Sinir ağı tabanlı sınıflandırıcılar temel yapay sinir ağı (YSA), çok katmanlı yapay sinir ağı (ÇKYSA), ileri beslemeli sinir ağı (FFNN), geri beslemeli sinir ağı (RNN), ileri kademeli sinir ağı (CFNN), olasılıksal sinir ağı (PNN) ve genel regresyon sinir ağı (GRNN) seçilmiştir. Bu çalışmada ayrıca yeni bir YSA modeli olan ve bu alanda kullanılmamış aşırı öğrenme tabanlı algoritmalar kullanılmıştır. Bu sınıflandırıcılar ise temel aşırı öğrenme makinesi (ELM), çekirdek tabanlı aşırı öğrenme makinesi (KELM), çok katmanlı algılayıcı aşırı öğrenme makinesi (MLP-ELM), çok katmanlı aşırı öğrenme makinesi (ML-ELM) ve çok katmanlı çekirdek tabanlı aşırı öğrenme makinesi (ML-KELM) olmak üzere beş farklı mimari olarak ele alınmıştır.

### 7.1. Geleneksel Sınıflandırıcılar

K en yakın komşu (k-NN) algoritması sınıfı bilinmeyen bir örneği, kendisine en yakın olan komşu sınıfa dahil etme yöntemidir. Bunun için iki nokta arasındaki uzaklığı hesaplayan yöntemler kullanılır. k-NN algoritması ile sınıflandırma, önceden belirlenmiş k değerine göre uzaklıkları hesaplanmış eğitim verileri içerisinde en yakınındaki k verisi çoğunlukla hangi sınıfa aitse test verisinin de o sınıfa ait olduğu belirleme işlemidir (Dasarathy, 1991) (Han vd., 2006). Bütün eğitim verileri ile test verilerinin uzaklıkları tek tek hesaplanır ve belirlenecek k değerine göre sınıflandırma sonucuna karar verilir. Burada komşu terimi eğitime kümesindeki verileri, yakınlık ise öklid, manhattan, minkowski, mahalnobis gibi bir uzaklık ölçütünü belirtir. k-NN algoritması kolay anlaşılabilir bir algoritma olduğu için uygulaması basittir.

Naive Bayes (NB); Bayes teoremine dayanan makine öğrenmesi yöntemlerinde kullanılan öngörmeli bir sınıflandırma tekniğidir. Bayes teoreminin bağımsız önermeler ile basitleştirilmiş halidir. Bu yöntemde, her öznelik vektörünün her bir sınıfta bulunma olasılığını hesaplayarak bu olasılıkların çarpımından tüm öznelikleri taşıyan örneğin hangi sınıfta olduğunu sayısal bir değerle ifade edilmektedir (Friedman vd., 1997). Naive Bayes sınıflandırması olasılık ilkelerine göre tanımlanmış bir dizi hesaplama ile sisteme sunulan verilerin sınıfını yani kategorisini tespit etmeyi amaçlar. Sınıflandırma yaparken özellikleri birbirinden bağımsız olarak hesaba katar.

Destek vektör makinesi (SVM) sınıflandırma için kullanılan en önemli yöntemlerden biridir. Sınıflandırma için bir düzlemde bulunan iki grup arasında bir sınır çizilerek iki grubu ayırmak mümkündür. Bu sınırın çizileceği yer ise iki grubun da üyelerine en uzak olan yer olmalıdır. SVM bu sınırın nasıl çizileceğini belirler. Bu işlemin yapılması için iki gruba da yakın ve birbirine paralel iki sınır çizgisi çizilir ve bu sınır çizgileri birbirine yaklaştırılarak ortak sınır çizgisi üretilir (Huang vd., 2007).

Karar ağaçları (DT) algoritması, makine öğrenmesi yöntemlerinden birisidir ve aynı zamanda en çok kullanılan gözetimli öğrenme algoritmalarındandır. Verinin özelliklerinden basit kurallar çıkarıp bu kuralları öğrenerek bir değişkenin değerini tahmin eden model oluşturur (Albright vd., 2006). DT öğrenmesinde, bir ağaç yapısı oluşturularak ağacın yaprakları seviyesinde sınıf etiketleri ve bu yapraklara giden ve başlangıçtan çıkan kollar ile de özellikler üzerindeki işlemler ifade edilmektedir. Yapıları itibarıyla en tepeden en aşağı inen bir strateji sunmaktadırlar. DT algoritmasının birçok farklı türevleri bulunmaktadır. Başlıca bilinenler ID3, C4.5, chaid, cart, mars, rastgele orman, yükseltilmiş ağaçlar, rotasyon orman olarak söyleyebiliriz. Bu türevler amaç olarak aynı işlemi yapmaktalar ancak hesaplama formülleri olarak farklılık gösterirler.

Rastgele orman (RF) algoritması denetimli bir sınıflandırma algoritmasıdır. Algoritma rastgele olarak bir orman oluşturmaktadır. Algoritmadaki ağaç sayısı ve elde edebileceği sonuç arasında doğrudan bir ilişki bulunmaktadır. Ağaç sayısı arttıkça kesin bir sonuç elde edilir. Çalışma mantığı sınıflandırma işlemi sırasında birden fazla karar ağacı üreterek sınıflandırma oranını yükselten algoritmadır. Bir sonuç üreteceği zaman bu karar ağaçlarındaki ortalama değer alınır ve sonuç üretilir (Loupe, 2015).

## 7.2. Sinir Ağı Tabanlı Sınıflandırıcılar

Yapay sinir ağları (YSA)'nın tahmin, kestirim, sınıflandırma ve optimizasyon gibi bir çok problem çözümünde çok sık kullanılmaktadır. YSA, eğitimle elde edilen bilgileri saklayan ve kullanabilen basit işlem birimlerinden oluşan paralel bir dağıtık işlemcidir (Haykin, 2005). YSA, veri kümesinde var olan örüntüleri ortaya çıkarabilmekte ve başarılı bir şekilde genelleştirebilmektedir. Girdi ve çıktı arasında bağlantıların sahip olduğu ağırlık değerlerini ve eşik değerlerini değiştirmek suretiyle, ilişkileri ortaya çıkarabilmektedir (Haykin, 2005). YSA uygulamalarının çoğu; giriş, gizli ve çıkış olmak üzere en az üç katman gerektirir. Giriş katmanı bağımsız değişkenlerin sayısını, çıkış katmanı ise bağımlı değişkenlerin sayısını içerir. Gizli katman modelin özellik belirleyicisidir. Kullanıcı deneme-yanılma yöntemiyle en uygun gizli katman sayısını ve gizli katmandaki nöron sayısını seçmelidir. Aktivasyon fonksiyonları YSA'nın çıktısını şekillendiren bir fonksiyondur (Haykin, 2005). YSA hangi tip problemin çözümünde kullanılacaksa buna uygun aktivasyon fonksiyonunun seçilmesi gerekmektedir. Literatürde, çok katmanlı algılayıcı, ileri beslemeli, geri beslemeli, ileri kademeli, genel regresyon, radyal temelli sinir ağları gibi birçok YSA çeşidi bulunmaktadır.

Çok katmanlı yapay sinir ağı (ÇKYSA) doğrusal olmayan problemlerin çözümünde en sık kullanılan bir YSA modelidir. ÇKYSA modeli çok sayıda ara katman içeren YSA'dır. ÇKYSA'nda sinirler ayrı paralel olarak oluşturulan katmanlara yerleştirilirler. Teorik olarak bu modelde çok sayıda ara katman bulunabilir. Fakat bu eğitilen ağı karmaşıklığını önemli derecede artıran faktördür. Bir veya bir kaç ara katmanlı ağlarda sağlamlık, doğruluk ve genelleştirme oranı çoğu durumda sağlanmaktadır. Ara katman sayısının ve ara katman nöron sayılarının belirlenmesinde genel olarak bir kural bulunmamaktadır. Belirlenme işlemi aşamasında yapılan eşlemenin karmaşıklığına bağlı olarak değişebilir. Giriş ve çıkış katman sayılarının belirlenmesi tamamen tanımlanan probleme özgüdür.

İleri beslemeli sinir ağlar (FFNN), YSA'nın en basit ve belleksiz yapısıdır. Bu ağda bilgi sadece ileri yönde gizli ve çıkış katmanına doğru hareket eder (Rojas, 1996). İleri beslemeli bağlantıların ağırlıkları eğitim sırasında değiştirilebilir İki katmanlı ileri beslemeli ağlar neredeyse tüm girdi çıktı ilişkilerini potansiyel olarak öğrenebilirken, daha fazla katman içeren ileri beslemeli ağlar karmaşık ilişkileri daha hızlı öğrenebilir.

Geri beslemeli sinir ağı (RNN) çıkış ve ara katlardaki çıkışlardan giriş birimlerine veya önceki ara katmanlara geri beslenmenin yapıldığı bir ağ yapısıdır (Bose, 2002). Bu çeşit

sinir ağlarının dinamik hafızaları vardır. Bu yapıdaki nöronların çıkışı sadece o anki giriş değerlerine bağlı değildir ayrıca önceki giriş değerlerine de bağlıdır. Bundan dolayı, bu ağ yapısı özellikle tahmin uygulamaları için uygundur.

İleri kademeli sinir ağı (CFNN), ileri beslemeli sinir ağına benzer şekilde çalışmaktadır. Bu sinir ağındaki giriş değerleri ileri beslemeliden farklı olarak tüm katmanlarla bağlantılıdır (Goyal ve Goyal, 2011). Aynı ileri beslemelide olduğu gibi öğrenme özelliğine sahiptir. İki veya daha fazla katman kullanılarak ve yeterli nöron sayısı seçilmesiyle giriş ile çıkış arasındaki ilişki sağlanabilir.

Olasılıksal sinir ağı (PNN) radyal temelli bir sinir ağıdır. Danışmanlı öğrenmeye dayalıdır. PNN, Bayes teoremi kullanarak karar veren bir sinir ağıdır. PNN, diğer ağlara göre daha fazla hesaplama karmaşıklığı içerdiğinden daha yavaş çalışır (Specht, 1990).

Genel regresyon sinir ağı (GRNN), PNN gibi radyal temelli bir yapay sinir ağıdır (Specht, 1990). Nöron yapısında radyal temelli katmanın yanı sıra özel bir doğrusal katman bulunur. Az sayıda eğitim örneğinden bile öğrenebilme yeteneğinden dolayı tercih edilen bir sinir ağıdır. PNN'den farklı olarak sürekli çıkış verebildiğinden regresyon için de kullanılır. Sinir ağı tabanlı sınıflandırıcılarda kullanılan katman sayısı, hücre (nöron) sayısı, döngü (epoch) sayısı ve fonksiyon bilgileri Tablo 7.1'de gösterimi sunulmuştur.

**Tablo 7.1.** Sinir ağı tabanlı sınıflandırıcılarda kullanılan fonksiyon, katman ve hücre sayısı

Sınıflandırıcı	Katman Sayısı	Gizli Katman Nöron Sayısı	Çıktı Katman Nöron Sayısı	Döngü (Epoch) Sayısı	Fonksiyon
YSA	1	10	21	50	sigmoid
ÇKYSA	3	10	21	50	sigmoid
FFNN	1	10	21	50	sigmoid
CFNN	1	10	21	50	sigmoid
RNN	1	10	21	50	sigmoid
GRNN	1		21		sigmoid
PNN	1		21		sigmoid

### 7.3. Aşırı Öğrenme Makinesi (ELM) Tabanlı Sınıflandırıcılar

Son zamanlarda, ELM makine öğrenmesi alanında ilgi duyulan öğrenme yöntemi haline gelmiştir. Huang ve arkadaşları (Guang-BinHuang vd., 2006) yaptıkları çalışma ile aşırı öğrenme makinesi (ELM) olarak adlandırdıkları öğrenme algoritmasını önermişlerdir. ELM giriş ağırlıkları rastgele çıkış ağırlıkları ise analitik olarak hesaplanan tek gizli katmanlı ileri beslemeli (*Single-hidden Layer Feedforward Neural Network (SLFN)*) bir YSA modelin özelleştirilmiş halidir. Bu nedenle aşırı öğrenme makinelerinin çalışma prensibi belirli bir dereceye kadar YSA'nın çalışma prensipleri ile aynıdır. Ancak aşırı öğrenme makinelerinde

gizli katmandaki ağırlıklar rastgele bir şekilde atanmaktadır ve bu değerler eğitimin ileriki aşamasında güncellenmemektedir. Buna karşın gizli katman ile çıktı katmanı arasındaki ağırlıklar tek bir seferde doğrusal bir model yardımıyla analitik ve hızlı bir şekilde belirlenmektedir. Bu modelde giriş katmandaki nöronlara ait ağırlıklar ve gizli katmandaki nöronlara ait eşik değerler rastgele üretilirken gizli katmandaki çıkış ağırlıkları analitik olarak hesaplanmaktadır (Tang vd., 2015). Bu modelin en önemli özelliği öğrenme sürecinin çok hızlı gerçekleşmesidir. ELM'de gizli katmanda sigmoid, sinüs ve Gaussian gibi aktivasyon fonksiyonları kullanılırken çıkış katmanda doğrusal fonksiyon kullanılmaktadır (Huang G. B., 2014). Deneysel sonuçlar bu algoritmanın iyi bir genelleme başarımı sunduğunu göstermiştir. İleri beslemeli sinir ağları için bilinen algoritmalarından binlerce kez hızlı öğrenebildikleri ifade edilmiştir.

Huang ve arkadaşları (Cao vd., 2012) (Cambria, et al., 2013), kernel çekirdek yöntemi ve aşırı öğrenme makinesini bir araya getirerek, çekirdek tabanlı aşırı öğrenme makinesini (KELM) ortaya koydu. Gizli ELM katmanının çıktıları, numunelerin doğrusal olmayan haritalandırılması olarak kabul edilebilir. KELM'in en popüler çekirdek işlevleri Gauss, RBF'dir.

Çok katmanlı aşırı öğrenme makinesi (ML-ELM), Huang ve arkadaşları (Cambria vd., 2013) tarafından önerilen tek gizli katman besleme ileri sinir ağlar (SLFN'ler) için olan aşırı öğrenme makinesi (ELM) temelli çok katmanlı bir ELM mimaridir. Bununla birlikte, ML-ELM, sadece karmaşık bir işlevi yerine getirmekle kalmayıp aynı zamanda eğitim sürecinde yineleme gerektirmeyen bir tür derin sinir ağıdır. Yeni yöntemde, önce her katmandaki parametreler denetimsiz eğitim ile elde edilir, sonra ağa denetimli öğrenme ile ince ayar yapılır. ML-ELM, Kasun ve arkadaşları (Kasun vd., 2013) tarafından önerildi. ML-ELM, her katmandaki parametreleri eğitmek için denetimsiz öğrenmeyi kullanmaktadır. Aynı zamanda, derin ağların aksine, ML-ELM ince ayar gerektirmez. Bu nedenle, diğer derin öğrenme algoritmalarına kıyasla, ML-ELM'in ağ eğitimi için uzun zaman harcamasına gerek yoktur. ML-ELM gizli katman etkinleştirme işlevleri doğrusal ya da doğrusal olmayan herhangi bir fonksiyon olabilir.

Çekirdek öğreniminden esinlenerek, ML-ELM'nin bir çekirdek modeli geliştirilerek, çok katmanlı çekirdek tabanlı ELM (ML-KELM) önerilmiştir (Wong vd., 2018). ML-KELM'in diğer ELM modellerindeki gibi tüm katmanlar için parametreleri ayarlamasına gerek yoktur. ML-KELM, sabit parametreler altında tek seferde en uygun modeli öğrenir. Özetle, önerilen ML-KELM, daha az kullanıcı müdahalesi ve diğer ELM modellerine göre

birçok dezavantajı çözen birden fazla KELM-AE'yi istifleyerek ML-ELM'nin bir çekirdek sürümüdür. ML-KELM ve KELM-AE yöntemleri **Algoritma 3**'de sözde kodu verilen algoritma ile ayrıntılı olarak açıklanmıştır.

---

**Input:** Input matrix  $\mathbf{X}^{(i)}$ ,

Regularization  $C_i$ ,

Kernel parameter  $\sigma_i$ ,

Activation function  $g_i$ ,

Output matrix  $\mathbf{T}$ ,

Number of layers  $N_{\text{Layer}}$

**Output:** Output weight  $\beta$ ,

Kernel matrix  $\Omega$ ,

New data representation  $\mathbf{X}^{(i+1)}$ ,

Transformation matrix  $\tilde{\mathbf{I}}^{(i)}$ ,

Representation matrix  $\mathbf{X}^{\text{final}}$ ,

Two transformation matrix  $\tilde{\mathbf{I}}^{(1)}$  and  $\mathbf{\Gamma}$

#### Algorithm KELM-AE for the $i$ th Layer

Step 1: Calculate  $\Omega_{k,j}^{(i)} \leftarrow K(\mathbf{x}_k, \mathbf{x}_j, \sigma_i)$

Step 2: Calculate  $\tilde{\mathbf{I}}^{(i)} \leftarrow (\mathbf{I} / C_i + \Omega^{(i)})^{-1} \mathbf{X}^{(i)}$

Step 3: Calculate  $\mathbf{X}^{(i+1)} \leftarrow g_i(\mathbf{X}^{(i)} (\tilde{\mathbf{I}}^{(i)})^T)$

**return**  $\mathbf{X}^{(i+1)}$ , and  $\tilde{\mathbf{I}}^{(i)}$

#### Algorithm ML-KELM

Step 1: Initialize  $\mathbf{I}^{\sim} \leftarrow \tilde{\mathbf{I}}^{(1)}$

**for**  $i = 1 : N_{\text{Layer}} - 1$  **do**

Step 2: Calculate  $\mathbf{X}^{(i+1)}, \tilde{\mathbf{I}}^{(i)} \leftarrow \text{KELMAE}(\mathbf{X}^{(i)}, C_i, \sigma_i, g_i)$

Step 3: Update  $\mathbf{I}^{\sim} \leftarrow \tilde{\mathbf{I}}^{(i)} \mathbf{I}^{\sim}$

Step 4:  $i \leftarrow N_{\text{Layer}}; \mathbf{X}^{\text{final}} \leftarrow \mathbf{X}^{(i)}$

Step 5: Calculate  $\Omega_{k,j}^{(i)} \leftarrow K^{(i)}(\mathbf{x}_k^{(i)}, \mathbf{x}_j^{(i)}, \sigma_i)$

Step 6: Calculate  $\beta \leftarrow (\mathbf{I} / C_i + \Omega^{(i)})^{-1} \mathbf{T}$

**return**  $\mathbf{X}^{\text{final}}, \tilde{\mathbf{I}}^{(1)}, \mathbf{\Gamma}$ , and  $\beta$

---

**Algoritma 3.** ML-KELM ve KELM-AE algoritma sözde kodu

Bu çalışmada; önerilen işaret dili tanıma modelinde sınıflandırıcı olarak temel ELM, MLP-ELM (MLP-ELM, 2021), ML-ELM, KELM ve ML-KELM kullanılmıştır. Temel ELM, MLP-ELM ve ML-ELM sınıflandırıcılarında hücre sayısı 100 ve sigmoid aktivasyon fonksiyonu kullanılmıştır. Çok katmanlı sınıflandırıcılarda katman sayısı 3'tür. Çekirdek tabanlı KELM ve ML-KELM sınıflandırıcılarında *RBF\_kernel* fonksiyonu tercih edilmiştir. KELM ve ML-KELM'de hücre sayısı kullanmaya gerek yoktur. ELM tabanlı sınıflandırıcılarda kullanılan fonksiyon, katman ve hücre sayıları Tablo 7.2'de sunulmuştur.

**Tablo 7.2.** ELM tabanlı sınıflandırıcılarda kullanılan fonksiyon, katman ve hücre sayısı

Sınıflandırıcı	Katman Sayısı	Hücre Sayısı	Fonksiyon
ELM	1	100	sigmoid
MLP-ELM	3	100	sigmoid
ML-ELM	3	100	sigmoid
KELM	1	-	RBF_kernel
ML-KELM	3	-	RBF_kernel

#### 7.4. Boyut İndirgeme Yöntemleri

Boyut indirgeme yöntemleri, çok sayıda özelliğe sahip yüksek boyutlu veri kümesinden daha düşük boyutlara sahip bir veri kümesi elde etmek için gereksiz ve bağımlı özellikleri kaldırmak için kullanılır. Bu çalışmada, makine öğrenme uygulamaları için önileme aşamasında boyut azaltmada PCA, LDA ve PCA+LDA yöntemleri kullanılmıştır.

##### 7.4.1. Temel Bileşenler Analizi (PCA)

Temel bileşen analizi (Übeyli ve Güler, 2004) istatistik, makine öğrenmesi, veri madenciliği gibi birçok alanda sıkça kullanılan bir doğrusal boyut azaltma yöntemidir. PCA, veri kümesinin muhtemel bağımlı değişkenlerini, temel bileşenler olarak adlandırılan doğrusal bağımsız değişkenlere dönüştürmek için ortogonal dönüşüm kullanan istatistiksel bir yöntemdir. PCA yönteminde, d boyutlu n örneklili bir veri kümesi  $X = \{x_1, x_2, \dots, x_n\}$  için, kovaryans matrisi eşitlik (9) ile ifade edilir:

$$cov_{ij} = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T \quad (9)$$

Bu hesaplamada, ilk aşamada, veriye ait ortalama değer her elemandan çıkarılarak merkezleme işlemi gerçekleştirilir. Sonraki aşamada kovaryans matrisi, iki veya daha fazla rastgele değişken kümesi arasındaki korelasyon gücünün bir ölçüsünü sağlamak için hesaplanır. Daha sonra ise, öz değer-öz vektör ayrıştırma yöntemi ile kovaryans matrisin öz değerleri ve öz vektörleri bulunur. Devamında bileşenler seçilerek bir özellik vektörü

oluşturulur. Burada en yüksek öz değere sahip öz vektör, veri kümesinin temel bileşeni olarak belirlenir. Son aşamada ise, boyutu değiştirilmiş yeni veri kümesi elde edilmiş olur (Martinez ve Kak, 2001).

#### 7.4.2. Doğrusal Ayırım Analizi (LDA)

Doğrusal ayırım analizi, veri nesnelерinin sınıf bilgisini de kullanarak sınıfların en iyi ayrıştığı uzayda yer alan vektörleri bulmayı amaçlayan doğrusal boyut indirgeme yöntemidir (Übeyli ve Güler, 2004). PCA yöntemi ile birlikte en sık kullanılan yöntemler arasındadır. Bu yöntem, veri içerisinde bulunan farklı sınıflara ait grupların doğrusal ayrılabilirliğini maksimize ederek boyut azaltması yapar. LDA yönteminde boyut indirgeme işleminde sınıflar arası saçılım matrisi  $\mathbf{S}_B$  ve sınıf içi saçılım matrisi  $\mathbf{S}_W$  aşağıdaki şekilde tanımlanır:

$$S_B = \sum_{i=1}^c (\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^T \quad (10)$$

$$S_W = \sum_{i=1}^c \sum_{j=1}^{N_i} (\mathbf{x}_{i,j} - \bar{\mathbf{x}})(\mathbf{x}_{i,j} - \bar{\mathbf{x}})^T \quad (11)$$

Öncelikle, sınıflar arası saçılım matrisi  $\mathbf{S}_B$  olarak adlandırılan farklı sınıfların ortalaması arasındaki mesafe hesaplanmaktadır. Sonraki aşamada, sınıf içi saçılım matrisi  $\mathbf{S}_W$  olarak adlandırılan her sınıfın ortalaması ve örneği arasındaki mesafe hesaplanır. Devamında saçılım matrisleri için öz vektörler ve bunlara karşılık gelen öz değerler hesaplanır. Sonrasında öz değerler sıralanır ve belirlenen en büyük öz değerlere sahip öz vektörler seçilerek yeni bir matris oluşturulur. Son aşamada ise, veri kümesi ile oluşturulan matrisin iç çarpımı alınarak yeni veri kümesi elde edilir.

## 8. DENEYSEL SONUÇLAR

Türk İşaret Dili'ne ait çift elli statik parmak alfabesi ve dinamik sözcük tanıma sistemi olmak üzere iki uygulama üzerinde çalışılmıştır. Statik parmak alfabesi tanıma uygulamasında sadece PCA, LDA ve PCA+LDA boyut indirgeme yöntemleri kullanılırken, dinamik sözcük tanıma uygulamasında boyut indirgeme yöntemleri özellik seçimi algoritması ile birlikte kullanılmıştır. Bunun yanı sıra statik parmak alfabesi tanıma uygulamasında geleneksel algoritmalar kullanılırken, dinamik sözcük tanıma uygulamasında ELM tabanlı algoritmalar tercih edilmiştir.

### 8.1. LMC Tabanlı Parmak Alfabeti Tanıma Uygulaması

Başarım oranlarının değerlendirilmesinde 2 denek tarafından 17 harf için oluşturulan 3672 örnekten oluşan orijinal veri kümesi ve boyut indirgenerek oluşturulan 3 adet veri kümesi kullanılmıştır. İlk olarak makine öğrenmesinde kullanılan geleneksel sınıflandırma algoritmaları kullanılarak orijinal veri kümesinin başarım oranlarına bakılmıştır. Geleneksel sınıflandırma algoritmalarının orijinal veri kümesi için başarım oranları ve süreleri Tablo 8.1'de sunulmuştur. Orijinal veri kümesi kullanılarak 17 harf için gerçekleştirilen sınıflandırma algoritmalarının en yüksek başarım oranı %98,5 ile RF sınıflandırıcısı yöntemi ile elde edilmiştir. Algoritma uygulama aşamasında eğitim ve test süreleri ayrı hesaplanmıştır. Algoritma test sürelerine bakıldığında DT sınıflandırıcı en iyi süreye sahiptir.

**Tablo 8.1.** Orijinal veri kümesi ile sınıflandırma başarım oranları ve süreleri

1.Uygulama (119 özellik)		
Sınıflandırma algoritmaları	Başarım oranı	Test Süresi (sn)
k En Yakın Komşu Algoritması (kNN)	87	0,161
Naive Bayes (NB)	91	14,563
Destek Vektör Makinesi (SVM)	95	2,997
Karar Ağaçları (DT)	76	0,147
Rastgele Orman (RF)	98	1,798
Doğrusal Ayırım Analizi (LDA)	93	0,203

Orijinal veri kümesinin PCA boyut indirgeme yöntemi kullanılarak oluşturulan yeni veri kümesi 42 özelliğe indirgenerek aynı sınıflandırma algoritmalarına tabi tutulmuştur. PCA uygulanmış veri kümesi için sınıflandırma algoritmalarının başarım oranları ve süreleri Tablo 8.2'de sunulmuştur. PCA uygulanmış veri kümesi kullanılarak 17 harf için gerçekleştirilen en yüksek sınıflandırma başarım oranı SVM sınıflandırıcısı ile elde edilmiştir. PCA uygulanarak özellik sayısı düşürülen veri kümesinin orijinal veri kümesi ile başarım oranlarına

bakıldığında başarımlarının düştüğü söylenebilir. Algoritma sürelerine bakıldığında ise genel olarak sürelerin düştüğü gözlemlenmiş ve kNN sınıflandırıcısının en iyi süreye sahip olduğu belirlenmiştir.

**Tablo 8.2.** PCA uygulanmış veri kümesi ile sınıflandırma başarımları ve süreleri

<b>2.Uygulama (PCA) (42 özellik)</b>		
Sınıflandırma algoritmaları	Başarımları	Test Süresi (sn)
k En Yakın Komşu Algoritması (kNN)	85	0,109
Naïve Bayes (NB)	78	5,441
Destek Vektör Makinesi (SVM)	93	2,474
Karar Ağaçları (DT)	48	0,293
Rastgele Orman (RF)	91	3,497
Doğrusal Ayırım Analizi (LDA)	80	0,152

Orijinal veri kümesinin LDA boyut indirgeme yöntemi kullanılarak oluşturulan yeni veri kümesi 11 özelliğe indirgenerek aynı sınıflandırma algoritmalarına tabi tutulmuştur. LDA uygulanmış veri kümesi için sınıflandırma algoritmalarının başarımları ve süreleri Tablo 8.3’de sunulmuştur. LDA uygulanmış veri kümesi kullanılarak 17 harf için gerçekleştirilen en yüksek sınıflandırma başarımları RF sınıflandırıcısı ile elde edilmiştir. LDA uygulanarak özellik sayısı düşürülen veri kümesinin orijinal veri kümesi ile başarımlarına bakıldığında başarımlarının genel olarak arttığı gözlenmiştir. Algoritma sürelerine bakıldığında ise genel olarak tüm sınıflandırıcı tiplerinde sürelerin düştüğü, DT sınıflandırıcısının en hızlı sınıflandırma yaptığı belirlenmiştir.

**Tablo 8.3.** LDA uygulanmış veri kümesi ile sınıflandırma başarımları ve süreleri

<b>3.Uygulama (LDA) (11 özellik)</b>		
Sınıflandırma algoritmaları	Başarımları	Test Süresi (sn)
k En Yakın Komşu Algoritması (kNN)	93	0,086
Naive Bayes (NB)	94	1,512
Destek Vektör Makinesi (SVM)	95	1,776
Karar Ağaçları (DT)	85	0,126
Rastgele Orman (RF)	96	1,550
Doğrusal Ayırım Analizi (LDA)	93	0,135

Orijinal veri kümesi PCA+LDA melez boyut indirgeme yöntemi uygulanarak oluşturulan yeni veri kümesi 9 özelliğe indirgenerek aynı sınıflandırma algoritmalarına tabi tutulmuştur. PCA+LDA uygulanmış veri kümesi için sınıflandırma algoritmalarının başarımları ve süreleri Tablo 8.4’de sunulmuştur. PCA+ LDA uygulanmış veri kümesi

kullanılarak 17 harf için gerçekleştirilen en yüksek sınıflandırma başarımları SVM sınıflandırıcısı ile elde edilmiştir. PCA+LDA uygulanarak özellik sayısı düşürülen veri kümesinin orijinal veri kümesi ile başarımlarına bakıldığında başarımlarının düştüğü gözlemlenmiştir. Algoritma sürelerine bakıldığında ise genel olarak tüm sınıflandırıcı tiplerinde sürelerin düştüğü, DT sınıflandırıcısının yine en hızlı sınıflandırma yaptığı belirlenmiştir.

**Tablo 8.4.** PCA+LDA uygulanmış veri kümesi ile sınıflandırma başarımları ve süreleri

4.Uygulama (PCA+LDA) (9 özellik)		
Sınıflandırma algoritmaları	Başarımları	Test Süresi (sn)
k En Yakın Komşu Algoritması (kNN)	85	0,097
Naive Bayes (NB)	81	1,350
Destek Vektör Makinesi (SVM)	89	1,867
Karar Ağaçları (DT)	70	0,129
Rastgele Orman (RF)	88	1,801
Doğrusal Ayırım Analizi (LDA)	78	0,136

Geleneksel sınıflandırma algoritmalarının orijinal veri kümesi üzerindeki kNN ve DT dışında başarımların yüksek ve farklı veri kümelerinde birbirine yakın sonuçlar elde edildiği görülmektedir. Bunun yanında algoritmalarının yapıları farklı olmasından dolayı ise sürelerinde farklılıklar oluşmuştur. PCA ve LDA uygulanmış veri kümeleri için elde edilen başarımları karşılaştırdığımızda ise, LDA boyut indirgeme yönteminin sınıflandırma algoritmaları üzerinde başarımları ve süre olarak daha başarılı olduğu ve özellik sayısını daha düşük sayıya indirdiği görülmektedir. LDA'nın PCA+LDA birlikte uygulanmış veri kümesinin başarımlarına göre daha iyi sonuçlar verdiğini ve süre olarak da yakın değerler ürettiğini söyleyebiliriz. LDA'nın sakıncası ise yapısı gereği PCA'a göre çalışma süresi yaklaşık iki kat olmasıdır.

## 8.2. LMC Tabanlı Dinamik Sözcük Tanıma Uygulaması

Başarımlarının değerlendirilmesinde 4 denek tarafından oluşturulan 8000 örnekten oluşan orijinal veri kümesi için 3 aşamalı özellik seçimi ve boyut indirgeme işlemi yapılmıştır. Öncelikle birinci özellik seçim algoritması kullanılarak kritik özellikler için 3 ayrı alt küme belirlenmiştir. Kritik özelliklerin seçiminde aşamalı olarak özellik sayıları azaltılmıştır. Birinci özellik seçimi algoritması kullanılarak 119 özellik içerisinde her bir teknik için ilk aşamada 15 özellik, ikinci aşamada 10 özellik ve üçüncü aşamada 5 özellik seçilerek 5 farklı teknik için toplamda sırayla 75, 50 ve 25 özellik seçimi yapılmıştır. İkinci

özellik seçim algoritması ile belirlenen özellik alt kümelerine ait 3 adet veri kümesi oluşturulmuştur. Bu 3 adet veri kümesi PCA, LDA ve PCA+LDA yöntemleri ile boyut indirgenerek 9 adet yeni veri kümesi oluşturulmuştur. Boyutu indirgenmiş veri kümeleri ile birlikte toplam 12 adet veri kümesi sınıflandırma sürecinde kullanılmıştır. Özellik seçimi ve boyut indirgeme yöntemleri ile elde edilen veri kümelerinin başarımlarının karşılaştırılması ve en yüksek başarımla sahip özellik alt kümesinin belirlenmesi hedeflenmiştir. 12 adet veri kümesinin oluşum aşamaları Tablo 8.5'de sunulmuştur.

**Tablo 8.5.** Veri kümelerinin oluşum aşamaları

1.Aşama	2.Aşama	3.Aşama
Birinci Uygulama	Özellik seçimi uygulaması	Boyut küçültme uygulaması
Orijinal Veri kümesi (119 özellik)	(5 teknik x 15 özellik) 75 özellik içeren veri kümesi	PCA (33 özellik)
		LDA (16 özellik)
		PCA+LDA (15 özellik)
	(5 teknik x 10 özellik) 50 özellik içeren veri kümesi	PCA (29 özellik)
		LDA (16 özellik)
		PCA+LDA (15 özellik)
	(5 teknik x 5 özellik) 25 özellik içeren veri kümesi	PCA (17 özellik)
		LDA (12 özellik)
		PCA+LDA (11 özellik)

Makine öğrenmesinde yer alan aşırı öğrenme (ELM) tabanlı sınıflandırıcılar kullanılarak ilk aşamada 75 özellik seçimi içeren veri kümesi ve 3 adet boyut indirgeme yöntemi kullanılarak oluşturulan veri kümelerinin başarımlarına bakılmıştır. Bu aşama 4 uygulamadan oluşmaktadır. Aşırı öğrenme tabanlı sınıflandırıcıların bu veri kümeleri için başarımları ve süreleri Tablo 8.6'da sunulmuştur. 75 özellikli ilk veri kümesi ve PCA, LDA ve PCA+LDA boyut indirgeme yöntemleri kullanılarak sırayla 33, 16 ve 15 özelliğe indirgenerek oluşturulan yeni veri kümeleri aynı sınıflandırma algoritmalarına tabi tutulmuştur. Tüm veri kümelerine bakıldığında 50 dinamik sözcük için en yüksek sınıflandırma başarımları ML-KELM sınıflandırıcı ile elde edilmiştir. Diğer yandan, uygulamalarda sırayla özellik sayısı düşmesine rağmen ML-KELM sınıflandırıcı başarımlarını korumuştur. Eğitim ve test süreleri açısından en az süreye sahip sınıflayıcı ise

ELM'dir. Uygulamalarda algoritma sürelerine bakıldığında sırayla özellik sayısı düşmesine rağmen sürelerde çok az bir azalma gözlemlenmiştir.

**Tablo 8.6.** 75 özellik içeren veri kümelerinin sınıflandırma başarımları ve süreleri

1.AŞAMA																
Uygulama	1.Uygulama (75 özellik)				2. Uygulama PCA (33 özellik)				3. Uygulama LDA (16 özellik)				4. Uygulama PCA+LDA (15 özellik)			
	Eğitim		Test		Eğitim		Test		Eğitim		Test		Eğitim		Test	
	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)
ELM	0,105	97	0,005	96	0,089	96	0,004	96	0,079	96	0,004	95	0,081	95	0,004	95
KELM	4,733	98	0,220	98	4,789	96	0,164	95	4,634	90	0,153	89	4,622	89	0,151	89
MLP-ELM	0,112	94	0,006	93	0,109	95	0,006	95	0,102	94	0,005	94	0,101	94	0,005	94
ML-ELM	0,167	98	0,006	98	0,155	97	0,006	97	0,142	95	0,005	95	0,140	94	0,005	94
ML-KELM	56,88	99	2,375	98	55,20	99	2,442	98	55,28	99	2,462	98	55,57	99	2,355	98

İkinci aşamada 50 özellik seçimi içeren veri kümesi ve 3 adet boyut indirgeme yöntemi kullanılarak oluşturulan veri kümelerinin başarımlarına bakılmıştır. Bu aşamada da 4 uygulamadan oluşmaktadır. Aşırı öğrenme tabanlı sınıflandırıcıların bu veri kümeleri için başarımları ve süreleri Tablo 8.7'de sunulmuştur. 50 özellikli ilk veri kümesi ve PCA, LDA ve PCA+LDA boyut indirgeme yöntemleri kullanılarak sırayla 29, 16 ve 15 özelliğe indirgenerek oluşturulan yeni veri kümeleri aynı sınıflandırma algoritmalarına tabi tutulmuştur. Tüm veri kümelerine bakıldığında 50 dinamik sözcük için en yüksek sınıflandırma başarımları ML-KELM sınıflandırıcı ile elde edilmiştir. Uygulamalarda sırayla özellik sayısı düşmesine rağmen ML-KELM sınıflandırıcı başarımlarını korumuştur. Eğitim ve test süreleri açısından en az süreye sahip sınıflayıcı ise ELM'dir. Uygulamalarda algoritma sürelerine bakıldığında sırayla özellik sayısı düşmesine rağmen sürelerde çok az bir azalma gözlemlenmiştir.

**Tablo 8.7.** 50 özellik içeren veri kümelerinin sınıflandırma başarımları ve süreleri

2. AŞAMA																
Uygulama	1. Uygulama (50 özellik)				2. Uygulama PCA (29 özellik)				3. Uygulama LDA (16 özellik)				4. Uygulama PCA+LDA (15 özellik)			
	Eğitim		Test		Eğitim		Test		Eğitim		Test		Eğitim		Test	
	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)
ELM	0,092	97	0,005	96	0,008	97	0,003	96	0,080	97	0,004	96	0,081	96	0,004	96
KELM	4,842	97	0,216	97	4,552	95	0,154	95	4,456	93	0,153	92	4,616	91	0,151	91
MLP-ELM	0,110	96	0,006	95	0,106	96	0,006	96	0,102	96	0,005	96	0,102	96	0,005	95
ML-ELM	0,167	98	0,006	98	0,147	97	0,005	97	0,141	96	0,005	95	0,140	95	0,005	95
ML-KELM	56,84	99	2,410	98	55,47	99	2,387	98	54,97	99	2,399	98	55,02	99	2,369	98

Üçüncü aşamada ise 25 özellik seçimi içeren veri kümesi ve 3 adet boyut indirgeme yöntemi kullanılarak oluşturulan veri kümelerinin başarımlarına bakılmıştır. Bu aşama 4 uygulamadan oluşmaktadır. Aşırı öğrenme tabanlı sınıflandırıcıların bu veri kümeleri için başarımları ve süreleri Tablo 8.8’de sunulmuştur. 25 özellikli veri kümesi ve PCA, LDA ve PCA+LDA boyut indirgeme yöntemleri kullanılarak sırayla 17, 12 ve 11 özelliğe indirgenerek oluşturulan yeni veri kümeleri aynı sınıflandırma algoritmalarına tabi tutulmuştur. Tüm veri kümelerine bakıldığında, 50 dinamik sözcük için en yüksek sınıflandırma başarımları ML-KELM sınıflandırıcısı ile elde edilmiştir. Aynı zamanda uygulamalarda sırayla özellik sayısı düşmesine rağmen ML-KELM sınıflandırıcısı başarımlarını korumuştur. Eğitim ve test süreleri açısından en az süreye sahip sınıflayıcı ise ELM’dir. Uygulamalarda algoritma sürelerine bakıldığında sırayla özellik sayısı düşmesine rağmen sürelerde çok az bir azalma gözlemlenmiştir.

**Tablo 8.8.** 25 özellik içeren veri kümelerinin sınıflandırma başarımları ve süreleri

3. AŞAMA																
Uygulama	1. Uygulama (25 özellik)				2. Uygulama PCA (17 özellik)				3. Uygulama LDA (12 özellik)				4. Uygulama PCA+LDA (11 özellik)			
	Eğitim		Test		Eğitim		Test		Eğitim		Test		Eğitim		Test	
	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)	Süre (s)	Yüzdellik(%)
ELM	0,086	96	0,004	96	0,081	95	0,004	95	0,080	94	0,004	94	0,081	94	0,004	94
KELM	4,579	93	0,183	93	4,471	89	0,152	88	4,647	81	0,151	81	4,579	78	0,151	78
MLP-ELM	0,108	95	0,006	95	0,105	94	0,005	94	0,110	94	0,005	94	0,104	94	0,005	94
ML-ELM	0,153	96	0,005	96	0,136	92	0,005	91	0,144	88	0,005	88	0,145	85	0,005	85
ML-KELM	56,42	99	2,550	99	55,01	98	2,369	98	55,38	98	2,437	98	54,44	98	2,348	98

Yukarıda sonuçları tablolarla verilen tüm uygulamalara bakıldığında genel olarak tüm sınıflandırıcılar arasında, ML-KELM sınıflandırıcısının başarımlarını koruduğunu ve en iyi başarımlarını verdiği gözlemlenmiştir. Aynı zamanda daha az kullanıcı müdahalesi sunan ML-KELM sınıflandırıcısının kararlı bir yapıya sahip olduğu görülmektedir. Ancak hem eğitim hem de test için geçen en uzun süreye sahiptir. Hesaplama ve çalışma zamanındaki bu farklılığın nedeni, mimarının yanı sıra bu mimaride kullanılan hesaplama yapısındaki farklılıktır. Yani: ML-KELM eğitimi sırasında her katman için eğitimde kullanılan örnek sayısı kadar düğüm oluşturularak örnek sayısı büyüklüğünde kare çekirdek matrisi oluşturur. Test aşamasında bu işlem bir kez yapılır. Bu nedenle, nispeten büyük

boyutlu matris işlemleri uzun zaman alır. Bu etkinin KELM için de geçerli olduğu yukarıda verilen sonuçlardan da görülmektedir. Ancak, kısaca bahsedilen hesaplama şeklinin diğer yapılara göre daha sağlam ve istikrarlı bir genelleme kabiliyeti getirdiği açıktır. Bunun yanında ELM sınıflayıcı yapılarının farklı olmasından dolayı çalışma sürelerinde farklılıklar oluşmuştur. PCA, LDA ve PCA+LDA uygulanmış veri kümelerini başarımları, süre ve özellik sayısı olarak karşılaştırdığımızda; PCA+LDA melez boyut indirgeme yönteminin sınıflandırma başarımları üzerinde etkili olduğu görülmektedir. Ayrıca, KELM sınıflandırıcının boyut indirgenmiş veri kümeleri üzerinde başarımlarının düştüğü söylenebilir. İki aşamalı (özellik seçimi ve boyut indirgeme) gerçekleştirilerek, özellik sayısı azaltarak ve boyut küçülterek, oluşturulan veri kümeleri hesaplama, süre, maliyet, depolama ve başarımları yönünden üstünlük sunmaktadır.

### **8.3. Kinect Tabanlı Dinamik Sözcük Tanıma Uygulaması**

Bu çalışma kapsamında 1 denek tarafından 20 dinamik sözcük için oluşturulan 500 örnekten oluşan orijinal veri kümesi için özellik seçimi ve boyut indirgeme işlemi yapılmıştır. Özellik seçimi algoritması kullanılarak 120 özellik içerisinde her bir teknik için 15 özellik seçilerek 5 teknik için toplamda 75 özellik seçimi yapılmıştır. Belirlenen özellik alt kümelerine ait PCA, LDA ve PCA+LDA yöntemleri ile boyut indirgenerek yeni veri kümeleri oluşturulmuştur. Özellik seçimi ve boyut indirgeme yöntemleri ile elde edilen veri kümelerinin başarımlarını karşılaştırılması ve en yüksek başarımları elde eden özellik alt kümesinin belirlenmesi hedeflenmiştir. Başarımlarının değerlendirilmesinde makine öğrenmesinde kullanılan sinir ağı tabanlı ve aşırı öğrenme tabanlı (ELM ) algoritmalar kullanılmıştır.

Birinci aşamada makine öğrenmesinde yer alan aşırı öğrenme (ELM) tabanlı sınıflandırıcılar kullanılarak 75 özellik seçimi içeren veri kümesi ve 3 adet boyut indirgeme yöntemi kullanılarak oluşturulan veri kümelerinin başarımlarına bakılmıştır. Bu aşama 4 uygulamadan oluşmaktadır. Aşırı öğrenme tabanlı sınıflandırıcıların bu veri kümeleri için başarımları ve süreleri Tablo 8.9'da sunulmuştur. 75 özellikli ilk veri kümesi ve PCA, LDA ve PCA+LDA boyut indirgeme yöntemleri kullanılarak sırayla 32, 13 ve 13 özelliğe indirgenerek oluşturulan yeni veri kümeleri aynı sınıflandırma algoritmalarına tabi tutulmuştur. Tüm veri kümelerine bakıldığında, 20 dinamik sözcük için en yüksek sınıflandırma başarımları ML-ELM sınıflandırıcı ile elde edilmiştir. Aynı zamanda uygulamalarda sırayla özellik sayısı düşmesine rağmen ML-KELM sınıflandırıcı başarımları

oranını genel olarak korumuştur. Eğitim ve test süreleri açısından en az süreye sahip sınıflayıcı ise ELM'dir. Uygulamalarda algoritma sürelerine bakıldığında sırayla özellik sayısı düşmesine rağmen sürelerde çok az bir azalma gözlemlenmiştir.

**Tablo 8.9.** 75 özellik içeren veri kümelerinin sınıflandırma başarımları ve süreleri

1.DÜZEY	1. Uygulama				2. Uygulama				3. Uygulama				4. Uygulama							
	(75 özellik)								PCA (32 özellik)				LDA (13 özellik)				PCA+LDA (13 özellik)			
	Eğitim		Test		Eğitim		Test		Eğitim		Test		Eğitim		Test					
Sınıflayıcı	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)				
ELM	0.0112	98	0.0018	94	0.0081	98	0.0014	95	0.0083	97	0.0014	96	0.0085	97	0.0014	96				
KELM	0.0424	96	0.0061	95	0.0367	96	0.0019	95	0.0374	90	0.0016	88	0.0386	90	0.0017	88				
MLP-ELM	0.0137	95	0.0011	90	0.0116	96	0.0008	90	0.0119	97	0.0008	95	0.0120	97	0.0008	94				
ML-ELM	0.0362	99	0.0032	99	0.0270	99	0.0028	99	0.0252	93	0.0029	91	0.0246	93	0.0029	92				
ML-KELM	0.2622	99	0.0202	92	0.2380	99	0.0196	92	0.2391	99	0.0202	94	0.2366	99	0.0201	95				

İkinci aşamada makine öğrenmesinde yer alan sinir ağı tabanlı sınıflandırıcılar kullanılarak 75 özellik seçimi içeren veri kümesi ve 3 adet boyut indirgeme yöntemi kullanılarak oluşturulan veri kümelerinin başarımlarına bakılmıştır. Bu aşama da 4 uygulamadan oluşmaktadır. Sinir ağı tabanlı sınıflandırıcıların bu veri kümeleri için başarımları ve süreleri Tablo 8.10'da sunulmuştur. 75 özellikli ilk veri kümesi ve PCA, LDA ve PCA+LDA boyut indirgeme yöntemleri kullanılarak sırayla 32, 13 ve 13 özelliğe indirgenerek oluşturulan yeni veri kümeleri YSA, MNN, FFNN, CFNN, RNN, GRNN ve PRNN sinir ağı tabanlı sınıflandırma algoritmalarına tabi tutulmuştur. Tüm veri kümelerine bakıldığında 20 dinamik sözcük için en yüksek sınıflandırma başarımları YSA sınıflandırıcı ile elde edilmiştir. Uygulamalarda sırayla özellik sayısı düşmesine rağmen YSA sınıflandırıcısının başarımlarında çok az bir değişiklik gözlemlenmiş olup sınıflandırıcılar arasında en yüksek başarımla sahip sınıflayıcı olarak YSA belirlenmiştir. Eğitim ve test süreleri açısından en az süreye sahip sınıflayıcı ise YSA'dır. Tablo 35'den de görüleceği üzere, diğer sınıflayıcıların başarımlarında dikkate değer sapmalar gözlemlenmiştir. Uygulamalarda algoritma sürelerine bakıldığında sırayla özellik sayısı düşmesine rağmen sürelerde çok az bir azalma gözlemlenmiştir.

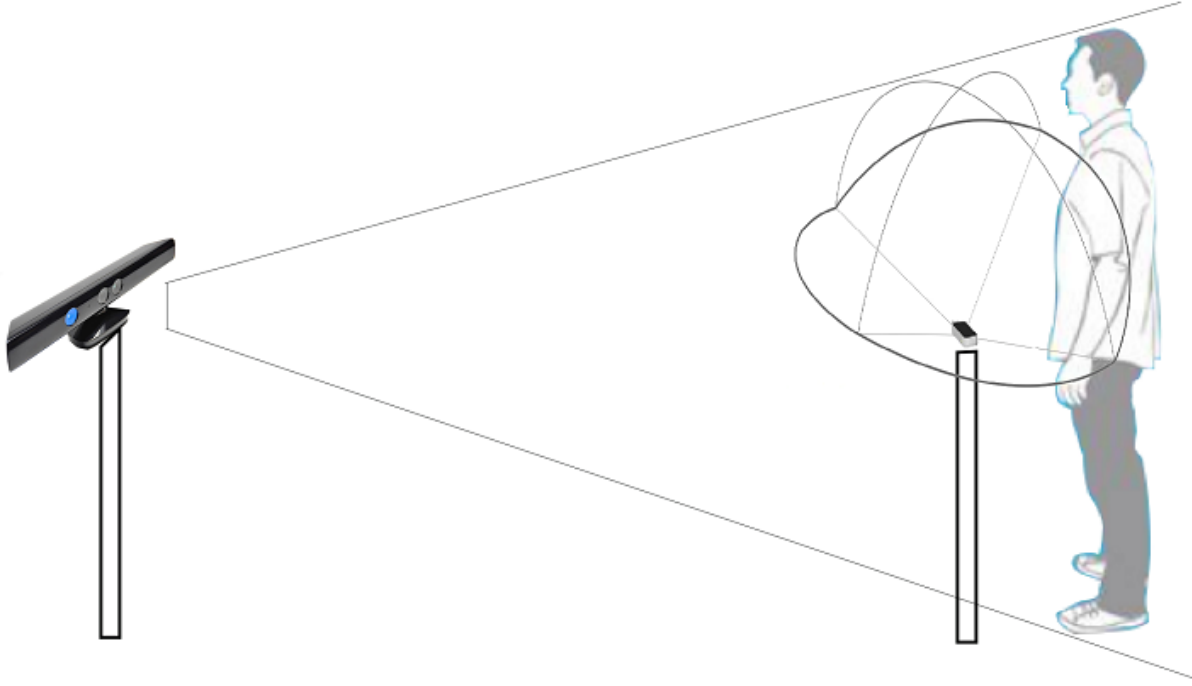
**Tablo 8.10.75** özellik içeren veri kümelerinin sınıflandırma başarımları ve süreleri

1.DÜZEY	1. Uygulama			2. Uygulama			3. Uygulama			4. Uygulama		
	(75 özellik)			PCA (32 özellik)			LDA (13 özellik)			PCA+LDA (13 özellik)		
	Eğitim	Test		Eğitim	Test		Eğitim	Test		Eğitim	Test	
Sınıflayıcı	süre (s)	süre (s)	Doğruluk (%)	süre (s)	süre (s)	Doğruluk (%)	süre (s)	süre (s)	Doğruluk (%)	süre (s)	süre (s)	Doğruluk (%)
YSA	0.307	0.022	97	0.158	0.012	98	0.139	0.011	94	0.143	0.012	96
ÇKYSA	0.378	0.015	91	0.246	0.015	96	0.194	0.015	87	0.218	0.016	90
FFNN	71.52	0.017	99	9.862	0.014	78	1.938	0.013	61	1.551	0.013	64
CFNN	554.7	0.018	92	53.093	0.014	99	4.604	0.014	91	5.063	0.014	94
RNN	69.96	0.016	99	12.58	0.014	79	3.796	0.014	70	3.464	0.014	70
GRNN	0.071	0.067	95	0.041	0.028	94	0.042	0.018	89	0.040	0.018	88
PNN	0.056	0.063	97	0.046	0.027	97	0.049	0.018	93	0.044	0.018	93

Makine öğrenmesinde kullanılan sinir ağı tabanlı ve aşırı öğrenme tabanlı (ELM) sınıflandırıcılar karşılaştırıldığında tüm veri kümeleri için en yüksek başarımları ELM, ML-KELM, YSA ve PRNN ağları ile elde edilmiştir.

#### 8.4. LMC + Kinect Tabanlı Dinamik Sözcük Tanıma Uygulaması

LMC cihazının etkileşim hacmi ve görüş alanı Kinect cihazına göre biraz daha sınırlıdır. TİD’de bazı sözcüklerin yapım aşamasında dirsek ve kol hareketlerinin bir kısmı LMC cihazının görüş alanı dışında kalabilmektedir. Bununla birlikte dinamik sözcüklerin yapımında el ve parmak kadar dirsek ve kol hareketleri de önem arz etmektedir. Kinect cihazı LMC cihazına göre etkileşim hacmi ve görüş alanı daha geniş ve farklıdır. Kinect el ve parmaklardan çok kol ve iskelet yapısına odaklanmaktadır. LMC ile birlikte Kinect cihazı da kullanılarak etkileşim hacmi ve görüş alanı artırılmıştır. Kinect cihazının çalışmaya eklenen özellik çıkarımlarının da uygulamaya katılarak daha fazla çeşit sözcük tanınması ve başarımın artırılması hedeflenmiştir. Sözcük seçiminde LMC cihazının görüş alanı dışına taşan sözcükler de eklenerek karma bir sözcük kümesi oluşturulmuştur. LMC ve Kinect cihazları görüş alanı gösterimi Şekil 8.1’de sunulmuştur.



**Şekil 8.1.** LMC ve Kinect cihazları görüş alanı gösterimi

Bu çalışma kapsamında 2 denek tarafından oluşturulan 21 dinamik sözcük için 20 tekrarlı olmak üzere toplam 840 örnekten oluşan orijinal veri kümesi için histogram ve zaman tabanlı özellik çıkarımı ve boyut indirgeme işlemi yapılmıştır. Histogram tabanlı özellik çıkarım kapsamında 230 özellik için belirlenen 12 farklı kategori ve bunlara ait oluşturulan aralıklar kullanılarak Tablo 24’de de gösterimi sunulan toplamda 1572 adet özellik çıkarımı elde edilmiştir. Histogram özellik çıkarımı sayesinde veri kümesi boyutunun eşitlenmesi ve küçültülmesi ve aynı zamanda normalleştirilmesi yapılmaktadır. Zamansal özellik çıkarımı kapsamında 230 özellik için belirlenen 5 farklı istatistiksel ölçüt kullanılarak toplamda 1050 adet özellik çıkarımı elde edilmiştir. Bu özellik çıkarımı da aynı şekilde veri kümesi boyutunun eşitlenmesine ve küçültülmesi işlemi yapmaktadır. Bu oluşturulan veri kümeleri PCA, LDA ve PCA+LDA yöntemleri ile boyut indirgenerek yeni veri kümeleri oluşturulmuştur. Bu yeni özellik çıkarımları ve boyut indirgeme yöntemleri ile elde edilen veri kümelerinin başarımlarını karşılaştırılması ve en yüksek başarımlarına sahip veri kümesinin belirlenmesi hedeflenmiştir. Başarımlarının değerlendirilmesinde makine öğrenmesinde kullanılan aşırı öğrenme tabanlı (ELM) algoritmalar kullanılmıştır.

Birinci aşamada makine öğrenmesinde yer alan aşırı öğrenme (ELM) tabanlı sınıflandırıcılar kullanılarak 1572 adet histogram özellik çıkarım içeren veri kümesi ve 3 adet boyut indirgeme yöntemi kullanılarak oluşturulan veri kümelerinin başarımlarına

bakılmıştır. Bu aşama 4 uygulamadan oluşmaktadır. Aşırı öğrenme tabanlı sınıflandırıcıların bu veri kümeleri için başarımları ve süreleri Tablo 8.11’de sunulmuştur. 1572 özellikli ilk veri kümesi ve PCA, LDA ve PCA+LDA boyut indirgeme yöntemleri kullanılarak sırayla 132, 16 ve 16 özelliğe indirgenerek oluşturulan yeni veri kümeleri aynı sınıflandırma algoritmalarına tabi tutulmuştur. Tüm veri kümelerine bakıldığında 21 dinamik sözcük için en yüksek sınıflandırma başarımları ML-ELM ve ML-KELM sınıflandırıcı ile elde edilmiştir. Aynı zamanda uygulamalarda sırayla özellik sayısı düşmesine rağmen ML-KELM sınıflandırıcı başarımlarını düşmediği ve koruduğu görülmüştür. Eğitim ve test süreleri açısından en az süreye sahip sınıflandırıcı ise ELM’dir. Uygulamalarda algoritma sürelerine bakıldığında sırayla özellik sayısı düşmesine rağmen sürelerde çok az bir azalma gözlemlenmiştir.

**Tablo 8.11.** Histogram özellik çıkarımına ait veri kümelerinin sınıflandırma başarımları ve süreleri

Histogram Tabanlı	1. Uygulama				2. Uygulama				3. Uygulama				4. Uygulama			
	(1572 özellik)				PCA (132 özellik)				LDA (16 özellik)				PCA+LDA (16 özellik)			
	Eğitim		Test		Eğitim		Test		Eğitim		Test		Eğitim		Test	
Sınıflayıcı	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)
ELM	0.0192	96	0.0025	84	0.0087	97	0.0014	90	0.0071	97	0.0013	97	0.0079	97	0.0013	96
KELM	0.0558	97	0.0221	94	0.0281	97	0.0020	94	0.0252	93	0.0012	92	0.0263	93	0.0013	91
MLP-ELM	0.0216	90	0.0018	72	0.0122	91	0.0007	75	0.0103	98	0.0006	94	0.0107	97	0.0007	94
ML-ELM	0.0600	99	0.0042	97	0.0296	99	0.0031	96	0.0231	97	0.0026	94	0.0244	96	0.0029	95
ML-KELM	0.3205	99	0.0182	93	0.1718	99	0.0133	93	0.1459	99	0.0128	95	0.1574	99	0.0126	95

Karışıklık matrisi, makine öğrenmesindeki sınıflandırma yöntemlerinin başarımlarını ölçümünü tanımlamak ve değerlendirmek için hedef niteliğe ait tahminlerin ve gerçek değerlerin karşılaştırıldığı sıklıkla kullanılan bir tablodur. Uygulamalara ait veri kümeleri tarafından doğru olarak tanınmayan işaret hareketlerinin ayrıntılarını karışıklık matrisi ile sunulabilir. Denekler tarafından bazı işaretlerin yapılış esnasında el ve parmak hareketlerinin yetersiz ya da eksik bir şekilde gerçekleştirilmesidir. Bu nedenle, bu tür hareketler diğer işaretlerle yanlış bir şekilde eşleştirilir ve tanıma performansını düşürür. Histogram özellik çıkarımına ait PCA+LDA eğitim ve test veri kümesinin ML-KELM sınıflandırıcı ait sonuçlarının tanıma performansı için karışıklık matrisi sırasıyla Tablo 8.12 ve Tablo 8.13 de sunulmuştur.

**Tablo 8.12.** PCA+LDA eğitim veri kümesinin ML-KELM ait tanıma performansı için karışıklık matrisi

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	<b>1,00</b>																				
2		<b>1,00</b>																			
3			<b>1,00</b>																		
4				<b>1,00</b>																	
5					<b>0,96</b>	0,04															
6					0,04	<b>0,96</b>															
7				0,07			<b>0,93</b>														
8								<b>1,00</b>													
9									<b>0,96</b>												0,04
10										<b>1,00</b>											
11											<b>1,00</b>										
12												<b>1,00</b>									
13								0,04					<b>0,96</b>								
14			0,04											<b>0,96</b>							
15															<b>1,00</b>						
16																<b>1,00</b>					
17																	<b>1,00</b>				
18																		<b>1,00</b>			
19																			<b>1,00</b>		
20																				<b>1,00</b>	
21															0,04						<b>0,96</b>

**Tablo 8.13.** PCA+LDA test veri kümesinin ML-KELM ait tanıma performansı için karışıklık matrisi

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	<b>1,00</b>																				
2		<b>0,83</b>																			0,17
3			<b>0,92</b>											0,08							
4				<b>1,00</b>																	
5					<b>0,83</b>	0,17															
6						<b>0,83</b>				0,08											0,08
7							<b>1,00</b>														
8								<b>0,92</b>									0,08				
9									<b>0,92</b>										0,08		
10										<b>1,00</b>											
11											<b>1,00</b>										
12												<b>1,00</b>									
13													<b>1,00</b>								
14														<b>1,00</b>							
15															<b>0,92</b>			0,08			
16																<b>1,00</b>					
17																	<b>0,83</b>			0,08	0,08
18																		<b>1,00</b>			
19																			<b>1,00</b>		
20																				<b>1,00</b>	
21																					<b>1,00</b>

İkinci aşamada makine öğrenmesinde yer alan aşırı öğrenme (ELM) tabanlı sınıflandırıcılar kullanılarak 1050 adet zamansal özellik çıkarım içeren veri kümesi ve 3 adet boyut indirgeme yöntemi kullanılarak oluşturulan veri kümelerinin başarımlarına bakılmıştır. Bu aşama da yine 4 uygulamadan oluşmaktadır. Aşırı öğrenme tabanlı sınıflandırıcıların bu veri kümeleri için başarımları ve süreleri Tablo 8.14’de sunulmuştur. 1050 özellikli ilk veri kümesi ve PCA, LDA ve PCA+LDA boyut indirgeme yöntemleri kullanılarak sırayla 137, 16 ve 16 özelliğe indirgenerek oluşturulan yeni veri kümeleri aynı sınıflandırma algoritmalarına tabi tutulmuştur. Tüm veri kümelerine bakıldığında 21 dinamik sözcük için en yüksek sınıflandırma başarımları ML-ELM ve ML-ELM sınıflandırıcı ile elde edilmiştir. Aynı zamanda uygulamalarda sırayla özellik sayısı düşmesine rağmen ML-ELM ve ML-KELM sınıflandırıcıların başarımlarının düşmediği ve genel olarak korunduğu görülmüştür. Eğitim ve test süreleri açısından en az süreye sahip sınıflandırıcı ise ELM’dir. Uygulamalarda algoritma sürelerine bakıldığında sırayla özellik sayısı düşmesine rağmen sürelerde çok az bir azalma gözlemlenmiştir.

**Tablo 8.14.** Zamansal özellik çıkarımına ait veri kümelerinin sınıflandırma başarımları ve süreleri

Zamansal Tabanlı	1. Uygulama				2. Uygulama				3. Uygulama				4. Uygulama			
	(1050 özellik)				PCA (137 özellik)				LDA (16 özellik)				PCA+LDA (16 özellik)			
	Eğitim		Test		Eğitim		Test		Eğitim		Test		Eğitim		Test	
Sınıflayıcı	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)	süre (s)	Doğruluk (%)
ELM	0.0149	96	0.0020	89	0.0085	98	0.0014	93	0.0072	99	0.0013	98	0.0077	99	0.0013	98
KELM	0.0449	99	0.0156	98	0.0284	99	0.0021	98	0.0248	96	0.0011	95	0.0265	96	0.0013	95
MLP-ELM	0.0172	93	0.0014	79	0.0118	94	0.0007	86	0.0100	99	0.0006	97	0.0110	99	0.0007	96
ML-ELM	0.0472	99	0.0037	98	0.0298	99	0.0031	98	0.0234	99	0.0027	98	0.0246	99	0.0029	98
ML-KELM	0.2634	99	0.0152	94	0.1692	99	0.0138	95	0.1478	99	0.0121	97	0.1560	99	0.0129	97

LMC + Kinect tabanlı dinamik sözcük tanıma uygulamasında kullanılan histogram ve zamansal tabanlı özellik çıkarımları kapsamında oluşturulan veri kümeleri için en yüksek başarımları ML-ELM ve ML-KELM sınıflandırıcılar ile elde edilmiştir. Aynı zamanda kullanılan histogram ve zamansal özellik çıkarımlarının başarımları karşılaştırıldığında yakın sonuçlar çıktığı gözlemlenmiştir.

## 9. SONUÇ VE ÖNERİLER

Sunulan bu çalışmada özellikle işaret dilini kullanarak hayatını devam ettiren işitme engelli insanların ve bu insanlar ile iletişim kuran kişilerin daha kolay anlaşmaları ve işaret dilini bilmeyen bir kişinin dahi bu dili anlayabilmesi ve iletişim kurabilmesi hedeflenmiştir. Bu çalışmada LMC ve Kinect cihazı kullanılarak, TİD'e ait statik parmak alfabesi harflerinin ve dinamik sözcüklerin tanınması üzerine çalışılmıştır. Çalışmanın ilk iki uygulaması, çift elle gerçekleştirilen 17 statik harf ve 50 dinamik sözcüğün LMC cihazı kullanılarak hazırlanan veri kümeleri üzerinde gerçekleştirilmiştir. Üçüncü uygulama 20 dinamik sözcüğün Kinect kullanılarak hazırlanan veri kümeleri üzerinde gerçekleştirilmiştir. Son uygulama ise 21 dinamik sözcüğün LMC+Kinect kullanılarak hazırlanan veri kümeleri üzerinde gerçekleştirilmiştir.

Öncelikle, statik parmak alfabesi harflerinin tanınmasında geliştirilen strateji ile ellerin farklı yönde kullanım çeşitliliği katılarak veri kümesi çeşitliliği artırılmıştır. İkinci çalışmada özellikle dinamik sözcük tanıma modeli ile yeni bir yaklaşım tarzı sunulmuştur. LMC ile yakalanan sıralı verileri kullanarak dinamik bir el hareketi tanıma yaklaşımı ortaya konulmuştur. LMC, birçok yönden üstünlükleri nedeniyle benzer cihazlara göre tercih edilmiştir. Bu sayede LMC cihazı, işitme engellilerin dinamik işaret dili tanıma kullanarak çevre ile kolay ve bağımsız iletişim kurmasını kolaylaştıracaktır. Önerilen model, farklı ve benzer şekilde yapılandırılmış sözcükleri kullanarak işaret girdilerini verimli bir şekilde yakaladığı için sağlamdır. Literatürdeki benzer çalışmalar daha çok statik harf ve sözcük işaretlerinin tanınması üzerinedir. Bu çalışma ile ELM tabanlı LMC ile TİD'de çift elle gerçekleştirilen dinamik sözcük hareketlerinin tanınmasında önemli bir adım atılmıştır. Dinamik hareketler için farklı sınıflandırma yöntemleri kullanılarak ve başarımları karşılaştırılmıştır. TİD'in çift elleri 50 dinamik işaret sözcüğünden oluşan bir işaret dili veri kümesi oluşturulmuştur. Bu veri kümesi kullanarak, özellik sayısı ve boyutu azaltılarak oluşturulan veri kümelerinin sözcük tanıma başarımlarının birçok yönü ele alınmıştır. Tanıma, ML-KELM yaklaşımı kullanılarak %98'lik en iyi doğruluğun kaydedildiği farklı sınıflandırma türleri ile gerçekleştirilmiştir. Bu çalışmada geliştirilen strateji ile özellik seçimi ve boyut indirgeme yöntemleri ile yüksek sınıflandırma başarımları elde edilmiştir. Belirtilen özellik çıkarımları ve seçimlerine göre işaretlerin oluşturduğu veri kümesi, diğer işaret dillerine de kolaylıkla uyarlanabilir.

**Tablo 9.1.** Uygulamalarda önerilen tanıma modeli aşamalarının özet gösterimi

Uygulama	Cihaz	Özellik sayısı	Kullanılan özellik seçimi, özellik çıkarımı, boyut indirgeme, sınıflandırıcı	Veri kümesi sayısı	Veri kümesi içeriği	En iyi sınıflandırıcı	% Başarım
1	LMC	119	<b>Boyut indirgeme yöntemleri</b> <b>(PCA, LDA, PCA+LDA)</b> Geleneksel sınıflandırıcılar	4 veri kümesi	17 statik alfabe 2 denek 50 tekrar	SVM RF	95 98
2	LMC	119	1.Düzye Özellik seçim algoritması (75 özellik) <b>Boyut indirgeme yöntemleri</b> <b>(PCA, LDA, PCA+LDA)</b> Aşırı öğrenme makinesi sınıflandırıcılar	4 veri kümesi	50 dinamik sözcük 4 denek 40 tekrar	ML-KELM	98
			2.Düzye Özellik seçim algoritması (50 özellik) <b>Boyut indirgeme yöntemleri</b> <b>(PCA, LDA, PCA+LDA)</b> Aşırı öğrenme makinesi sınıflandırıcılar	4 veri kümesi		ML-KELM	98
			3.Düzye Özellik seçim algoritması (25 özellik) <b>Boyut indirgeme yöntemleri</b> <b>(PCA, LDA, PCA+LDA)</b> Aşırı öğrenme makinesi sınıflandırıcılar	4 veri kümesi		ML-KELM	98
3	Kinect	120	1.Düzye Özellik seçim algoritması (75 özellik) <b>Boyut indirgeme yöntemleri</b> <b>(PCA, LDA, PCA+LDA)</b> <i>Sinir ağı tabanlı sınıflandırıcılar</i>	4 veri kümesi	20 dinamik sözcük 1 denek 25 tekrar	YSA	96
			1.Düzye Özellik seçim algoritması (75 özellik) <b>Boyut indirgeme yöntemleri</b> <b>(PCA, LDA, PCA+LDA)</b> <i>Aşırı öğrenme makinesi sınıflandırıcılar</i>	4 veri kümesi		ELM ML-KELM	96 96
4	LMC Kinect	110 120 (230)	Histogram özellik çıkarımı <b>Boyut indirgeme yöntemleri</b> <b>(PCA, LDA, PCA+LDA)</b> Aşırı öğrenme makinesi sınıflandırıcılar	4 veri kümesi	21 dinamik sözcük 2 denek 20 tekrar	ML-ELM ML-KELM	97 97
			Zamansal özellik çıkarımı <b>Boyut indirgeme yöntemleri</b> <b>(PCA, LDA, PCA+LDA)</b> Aşırı öğrenme makinesi sınıflandırıcılar	4 veri kümesi		ML-ELM ML-KELM	98 97

LMC+Kinect cihazları ile gerçekleştirilen uygulama kapsamında farklı uzunluk ve sürelerle sahip 21 dinamik sözcük için elde edilen veri kümesi kullanılarak 2 tür özellik çıkarımı kullanılmış ve başarımları karşılaştırılmıştır. Bu çalışmada kullanılan histogram ve zamansal özellik çıkarımı ile elde edilen veri kümelerinin boyutları eşitlenerek sınıflandırıcıya girişi kolaylaşır ve aynı zamanda verilerin boyutunun küçültülmesini sağlar. Bu özellik çıkarımları kapsamında oluşturulan veri kümeleri için ML-ELM ve ML-KELM sınıflandırıcılar ile en yüksek başarımlar elde edilmiştir. Kullanılan histogram ve zamansal özellik çıkarımlarının başarımları karşılaştırıldığında ise yakın sonuçlar çıktığı görülmüştür. Uygulamalarda önerilen tanıma modeli aşamalarının özet gösterimi Tablo 9.1’de sunulmuştur. Kısaca, bu çalışmanın katkıları şunlardır:

1. İşaret dili tanıma için bir sınıflandırma yöntemi olarak ELM öğreniminin kullanılmasına yönelik ilk uygulama olması
2. İşaret dili tanıma için yeni bir özellik seçim algoritması yaklaşımının kullanılması.
3. TİD’de histogram ve zamansal özellik çıkarımı yöntemi yaklaşımının kullanılması
4. TİD’de dinamik sözcük tanıma konusunda bu bağlamda ilk çalışma olması.
5. TİD’de LMC+Kinect cihazlarının kullanılması bu bağlamda ilk çalışma olması
6. TİD’de dinamik sözcük tanıma için kapsamlı bir veri kümesi oluşturma
7. Üç farklı boyut indirgeme yönteminin özellik çıkarma ve seçme yaklaşımlarının bütünleştirilmesiyle oluşturulan 9 farklı veri kümesi ile tanıma başarımlarının analizinin sunulması.

LMC cihazı ve API yazılımı yeni ve gelişime açık bir teknoloji olduğu için özellikle çift elle gerçekleştirilen harflerin algılanmasında yaşanan kısıtlamalar ve problemlerin gelecekte cihazın geliştirilmesiyle çözülebileceği ve daha iyi sonuçlar elde edilebileceği düşünülmektedir. Gelecekteki çalışmalarımızda, işaret cümlelerini tanımaya doğru genişletilecektir. Buna ek olarak, daha fazla kapsam alanı için yüz ifadeleri ve dudak hareketleri yakalanarak çerçeve genişletilebilir.

## KAYNAKÇA

- Abreu, J. G., Teixeira, J. M., Figueiredo, L. S., & Teichrieb, V.** (2016). Evaluating Sign Language Recognition Using the Myo Armband. *18th Symposium on Virtual and Augmented Reality, Gramado,RS,Brazil* , s. 64-70.
- Albright, S. C., Winston, W., & Zappe, C.** (2006). *Data Analysis & Decision Making*. Australia: 4th Edition, Thomson South-Western.
- Aliyu, S. O., Mohandes, M., & Deriche, M.** (2017). Dual LMCs Fusion for Recognition of Isolated Arabic Sign Language Words. *14th International Multi-Conference on Systems, Signals & Devices (SSD)*, s. 611-614.
- Al-Jarrah, O., & Halawani, A.** (2001). Recognition of gestures in Arabic sign language using neuro-fuzzy systems. *Artificial Intelligence* , 133, s. 117–138.
- Altun, O.** (2010). *İlgi Bölgeleri ve Yerel Tanımlayıcılar ile Genelleştirilmiş Hough Dönüşümü ve En Az Eylemsizlik Eksenli Tabanlı Hızlı Hizalama Yaklaşımı ile Türk İşaret Dili Tanıma Sistemi*. (Yayınlanmamış Doktora Tezi). Yıldız Teknik Üniversitesi FBE, İstanbul.
- Altun, O., & Albayrak, S. V.** (2011). Turkish Fingerspelling Recognition System Using Generalized Hough Transform. *Interest Regions, and Local Descriptors.Pattern Recognition Letters* , 32 (13), s. 1626-1632.
- Amatanon, V., Chanhang, S., Naiyanetr, P., & Thongpang, S.** (2014). Sign Language-Thai Alphabet Conversion Based on Electromyogram (EMG). *The 7th Biomedical Engineering International Conference (BMEICON), Fukuoka, Japan*, s. 1-4.
- Amin, M. A., & Yan, H.** (2007). Sign Language Finger Alphabet Recognition From Gabor-PCA Representation of Hand Gestures. *International Conference on Machine Learning and Cybernetics, Hong Kong, China* , s. 2218-2223.
- Anwar, A., Basuki, A., Sigit, R., Rahagiyanto, A., & Zikky, M.** (2017). Feature Extraction For Indonesian Sign Language (SIBI) Using Leap Motion Controller. *21st International Computer Science and Engineering Conference (ICSEC), Bangkok, Thailand*, s. 196-200.
- Aran, O.** (2008). *Vision Based Sign Language Recognition: Modelling and Recognizing Isolated Signs with Manual and Non-manual Components*. (Yayınlanmamış Doktora Tezi). Boğaziçi Üniversitesi FBE, İstanbul.

**Avola, D., Bernardi, M., & Cinque, L.** (2019). Exploiting Recurrent Neural Networks and LMC for the Recognition of Sign Language and Semaphoric Hand Gestures. *IEEE Transactions on multimedia* , 21 (1), s. 1-12.

**Bayrak, S. (2009).** *İşaret Dilinin Bilgisayarlı Yorumlanması.* (Yayınlanmamış Yüksek Lisans Tezi). KTÜ FBE, Trabzon.

**Bayrak, S., & Nabiye, V. V.** (2006). Recognition of Vowels Letters of Turkish Sign Language by Artificial Neural Networks. *IEEE 14th Signal Processing and Communications Applications, Antalya, Turkey*, s. 1-4.

**Bilgin, M., & Mutludogan, K.** (2019). American Sign Language Character Recognition with Capsule Networks. *IEEE 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey* , s. 1-6.

**Bose, B. K.** (2002). *Neural Network Principles and Applications.* Prentice Hall PTR, USA.

**Büyüksaraç, B.** (2015). *Sign Language Recognition By image Analysis.* (Yayınlanmamış Yüksek Lisans Tezi). ODTÜ Elektrik Elektronik Müh., Ankara.

**Cambria, E., Huang, G.-B., Kasun, L. L., Zhou, H., Vong, C. M., Lin, J., et al.** (2013). Extreme learning machines. *IEEE Intelligent Systems* , 28 (6), s. 30–59.

**Cao, J., Lin, Z., Huang, G. B., & Liub, N.** (2012). Voting based extreme learning machine. *Information Sciences* , 185 (1), s. 66–77.

**Caruso, L., R.Russo, & S.Savino.** (2017). Microsoft Kinect V2 vision system in a manufacturing application. *Robotics and Computer-Integrated Manufacturing*, 48, s. 174-181.

**Chophuk, P., Pattanaworapan, K., & Chamnongthai, K.** (2018). Fist American Sign Language Recognition Using Leap Motion Sensor. *International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, Thailand*, s. 541-544.

**Chuan, C.-H., Regina, E., & Guardino, C.** (2014). American Sign Language Recognition Using Leap Motion Sensor. *13th International Conference on Machine Learning ve Applications, Detroit, MI, USA*, s. 1-5

**Corti, A., Giancola, S., Mainetti, G., & Sala, R.** (2016). A metrological characterization of the Kinect V2 time-of-flight camera. *Robotics and Autonomous Systems* , 75, s. 584–594.

**Coşkun, M., & İstanbullu, A.** (2012). EEG İşaretlerinin FFT ve dalgacık dönüşümü ile analizi. *Akademik Bilişim '12 - XIV. Akademik Bilişim Konferansı* , s. 323-328.

- Çalışkan, A.** (2019). EMG sinyalleri için hibrid öznitelik çıkarma yöntemi geliştirilmesi. *ÖHÜ Müh. Bilim. Dergisi* , s. 652-664.
- Dahmani, D., & Larabi, S.** (2014). User-independent system for sign language finger spelling recognition. *J. Vis. Commun. Image R.* , 25, s. 1240–1250.
- Dasarathy, B. V.** (1991). Nearest-Neighbor Classification Techniques. *IEEE Computer Society Press, Los Alamitos, California*, s. 83-154.
- Demircioğlu, B., Bülbül, G., & Köse, H.** (2016). Recognition of Sign Language Hand Shape Primitives With Leap Motion. *7th workshop on the Representation and Processing of Sign Languages: Corpus Mining, 10th edition of the Language Resources and Evaluation Conference (LREC), Portoroz, Slovenya* , 1 (1), s. 47-52.
- Demircioğlu, B., Bülbül, G., & Köse, H.** (2016). Turkish Sign Language Recognition With Leap Motion. *24th Signal Processing and Communication Application Conference (SIU), Zonguldak*, s. 589-592.
- Demirel, U.** (2017). *Creating a generic hand and finger gesture recognizer by using forearm muscle activity signals.* (Yayınlanmamış Yüksek Lisans Tezi). ODTÜ Oyun Teknolojisi Bölümü, Ankara.
- Demirler, İ.** (2015). *Real-time recognition system of letter-based Turkish sign.* (Yayınlanmamış Yüksek Lisans Tezi). Gazi Üniversitesi FBE, Ankara.
- Deriche, M., Aliyu, S. O., & Mohandes, M.** (2019). An Intelligent Arabic Sign Language Recognition System Using a Pair of LMCs With GMM Based Classification. *IEEE Sensors Journal* , 19 (18), s. 1-12.
- Dikyuva, H., Makaroğlu, B., & Arık, E.** (2015). *Türk İşaret Dili Dilbilgisi Kitabı.* Ankara: T.C. Aile ve Sosyal Politikalar Bakanlığı.
- El-Bendary, N., Zawbaa, H. M., Daoud, M. S., Hassanien, A. E., & Nakamatsu, K.** (2010). Arabic Sign Language Alphabets Translator. *International Conference on Computer Information Systems and Industrial Management Applications (CISIM), Krakow, Poland* , s. 590-595.
- Enikeev, D., & Mustafina, S.** (2020). Recognition of Sign Language Using Leap Motion Controller Data. *2nd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russia* , s. 393-397.

- Enikeev, D., & Mustafina, S.** (2021). Russian Fingerspelling Recognition Using Leap Motion Controller. *3rd International Conference on Control Systems, Mathematical Modeling, Automation and Energy Efficiency (SUMMA), Lipetsk, Russian Federation* , s. 604-606.
- Eqab, A., & Shanableh, T.** (2017). Android Mobile App for Real-Time Bilateral Arabic Sign Language Translation Using Leap Motion Controller. *International Conference on Electrical and Computing Technologies and Applications (ICECTA), Ras Al Khaimah, United Arab Emirates*, s. 1-5.
- Fatmi, R., Rashad, S., Integlia, R., & Hutchison, G.** (2017). American Sign Language Recognition using Hidden Markov Models and Wearable Motion Sensors. *Transactions on Machine Learning and Data Mining* , 10 (2), s. 41-55.
- Friedman, N., Geiger, D., & Goldszmidt, M.** (1997). *Bayesian Network Classifiers*. Machine Learning, Kluwer Acad. Publisher.
- Goyal, S., & Goyal, G. K.** (2011). Cascade and Feedforward Backpropagation ANN Models For Prediction of Sensory Quality of Instant Coffee Flavoured Sterilized Drink. *Canadian Journal on Artificial Intelligence, Machine Learning and Pattern Recognition* , 2 (6), s.78-82.
- Göknar, G.** (2015). *Hand gesture recognition using Artificial Neural Networks*. (Yayınlanmamış Yüksek Lisans Tezi). Yeditepe Üniversitesi E.ve Elektronik Mühendisliği ,Ankara.
- Göknar, G., & Yıldırım, T.** (2005). Hand Gesture Recognition Using Artificial Neural Networks. *Proceedings of the IEEE 13th Signal Processing and Communications Applications Conference, Kayseri, Turkey*, s. 1-4.
- Guang-BinHuang, Zhu, Q.-Y., & Siew, C.-K.** (2006). Extreme learning machine: Theory and applications. *Neurocomputing* , 70, s. 489–501.
- Güngör, C., Bağrıaçık, T., Demirdöğen, İ., Günaydın, A., & Karahan, V.** (2015). *Türk İşaret Dili Sözlüğü*. Ankara: T.C. MEB Özel Eğitim ve Rehberlik Hizmetleri Genel Müdürlüğü.
- Haberdar, H.** (2005). *Saklı Markov Model Kullanılarak Görüntüden Gerçek Zamanlı İşaret Dili Tanıma Sistemi*. (Yayınlanmamış Yüksek Lisans Tezi). YTÜ FBE, İstanbul.
- Han, J., Kamber, M., & Pei, J.** (2006). *Data Mining: Concepts and Techniques*. San Francisco: Morgan Kaufmann Publishers.

- Hasan, M. M., Srizon, A. Y., Sayeed, A., & Hasan, M. A.** (2020). Classification of Sign Language Characters by Applying a Deep Convolutional Neural Network. *IEEE 2nd International Conference on Advanced Information and Communication Technology (ICAICT), Dhaka, Bangladesh* , s. 434-438.
- Haykin, S.** (2005). *Neural Networks. A Comprehensive Foundation, India:Pearson Education* .
- Hein, Z., Htoo, T. P., Aye, B., Htet, S. M., & Ye, K. Z.** (2021). Leap Motion based Myanmar Sign Language Recognition using Machine Learning. *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), St. Petersburg, Moscow, Russia* , s. 2304-2310.
- Hisham, B., & Hamouda, A.** (2019). Arabic Dynamic Gesture Recognition Using Classifier Fusion. *Journal Of Advances In Information Fusion* , 14 (1), s. 66-84.
- Hisham, B., & Hamouda, A.** (2017). Arabic Static and Dynamic Gestures Recognition Using Leap Motion. *Journal of Computer Science* , 13 (8), s. 337-354.
- Huang, C.-L., Chen, M.-C., & Wang, C.-J.** (2007). Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications*, 33, s. 847–856.
- Huang, G. B.** (2014). An Insight into Extreme Learning Machines: Random Neurons, Random Features and Kernels. *Cognitive Computation* , 6 (3), s. 376-390.
- Insani, C. N., Nurtanio, I., & Ilham, A. A.** (2019). The effect of light on Leap Motion Controller in the classification of Sign Language Translator System. *International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, Indonesia*, s. 296-300.
- Işıkdoğan, F., & Albayrak, S.** (2011). Automatic Recognition of Turkish Fingerspelling. *International Symposium on Innovations in Intelligent Systems and Applications, Istanbul, Turkey* , s. 264-267.
- Işıklıgil, E.** (2014). *A method for isolated sign recognition with kinect.* (Yayınlanmamış Yüksek Lisans Tezi). ODTÜ Bilgisayar Mühendisliği, Ankara.
- Kakayev, K.** (2016). *İşaretçi iskelet yapısını kullanarak saklı markov model ile Türk işaret dili tanıma sistemi.* (Yayınlanmamış Yüksek Lisans Tezi). Yıldız Teknik Üniversitesi FBEİstanbul.

- Karacı, A., Akyol, K., & Gültepe, Y.** (2018). Turkish Sign Language Alphabet Recognition with Leap Motion. *International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES'18), Safranbolu, Turkey*, s. 189-192.
- Kasun, L. L., Zhou, H., Huang, G.-B., & Vong, C.-M.** (2013). Representational Learning with ELMs for Big Data. *IEEE Intelligent Systems*, s. 30-59.
- Katılmış, Z., & Karakuzu, C.** (2021). ELM based two-handed dynamic Turkish Sign Language (TSL) word recognition. *Expert Systems With Applications*, s. 1-12.
- Katılmış, Z., & Karakuzu, C.** (2020). Recognition of Two-Handed Posture Finger Turkish Sign Language Alphabet. *5th International Conference on Computer Science and Engineering (UBMK), Diyarbakır, Turkey*, s. 181-186.
- Kaya, E.** (2018). *Machine learning techniques for surface electromyography based hand gesture recognition.* (Yayınlanmamış Yüksek Lisans Tezi). İTÜ Kontrol ve Otomasyon Müh., İstanbul.
- Kelly, D., McDonald, J., & Markham, C.** (2010). A person independent system for recognition of hand postures used in sign language. *Pattern Recognition Letters* , 31, s. 1359–1368.
- Ketenci, S., Kayıkçıoğlu, T., & Gangal, A.** (2016). Recognition of Sign Language Numbers via Electromyography Signals. *IEEE International Conference on Systems, Man, and Cybernetics (SMC),Budapest, Hungary*, s. 2593-2596.
- Khamid, Wibawa, A. D., & Sumpeno, S.** (2017). Gesture Recognition for Indonesian Sign Language Systems (ISLS) Using Multimodal Sensor Leap Motion and Myo Armband Controllers Based-on Naive Bayes Classifier. *International Conference on Soft Computing, Intelligent System and Information Technology (ICSIT),Denpasar, Indonesia* , s. 1-6.
- Khotimah, W. N., Saputra, R. A., Suciati, N., & Hariadi, R. R.** (2017). Alphabet Sign Language Recognition Using Leap Motion Technology and Rule Based Backpropagation-Genetic Algorithm Neural Network (RBBPGANN). *JUTI* , 15 (1), s. 95-103.
- Khotimah, W. N., Susanto, Y. A., & Suciati, N.** (2017). Combining decision tree and back propagation genetic algorithm neural network for Recognizing word gestures in indonesian sign Language using kinect. *Journal of Theoretical and Applied Information Technology* , 95 (2), s. 292-298.

- Kim, J., Wagner, J., Rehm, M., & Andre, E.** (2008). Bi-channel Sensor Fusion for Automatic Sign Language Recognition. *8th IEEE International Conference on Automatic Face & Gesture Recognition, Amsterdam, Netherlands*, s. 1-6.
- Kındıroglu, A. A.** (2011). *Bilgisayarla görme tabanlı çok dilli parmak alfabesi tanıma.* (Yayınlanmamış Yüksek Lisans Tezi). Boğaziçi Üniversitesi Bilgisayar Mühendisliği, İstanbul.
- Kosmidou, V. E., Hadjileontiadis, L. J., & Panas, S. M.** (2006). Evaluation of surface EMG features for the recognition of American Sign Language gestures. *Proceedings of the 28th IEEE EMBS Annual International Conference, New York City, USA* , s. 6197-6200.
- Kramer, J., Burrus, N., Echtler, F., C., D. H., & Parker, M.** (2012). *Hacking the Kinect.* A press.
- Kumar, P., Gauba, H., Roy, P. P., & Dogra, D. P.** (2017). A multimodal framework for sensor based sign language recognition. *Neurocomputing* , s. 1-18.
- Kumar, P., Gauba, H., Roy, P. P., & Dogra, D. P.** (2017). Coupled HMM-based multi-sensor data fusion for sign language recognition. *Pattern Recognition Letters* , 86, s. 1-8.
- LMC.** (2020). *Leap Motion Controller.* 8 1, 2021 tarihinde [https://developer-archive.leapmotion.com/documentation/csharp/devguide/Leap\\_Overview.html](https://developer-archive.leapmotion.com/documentation/csharp/devguide/Leap_Overview.html). adresinden alındı
- Loupe, G.** (2015). *Understanding Random Forest.* (Yayınlanmamış Doktora Tezi). University of Liege .
- Madushanka, A., Senevirathne, R., Wijesekara, L., Arunatilake, S., & Sandaruwan, K. (2016). Framework for Sinhala Sign Language recognition and translation using a wearable armband. *Sixteenth International Conference on Advances in ICT for Emerging Regions (ICTer), Negombo, Sri Lanka* , s. 49-57.
- Mapari, R. B., & Kharat, G.** (2017). Analysis of Multiple Sign Language Recognition Using Leap Motion Sensor. *International Journal of Research in Advent Technology (IJRAT)*, s. 541-544.
- Mapari, R. B., & Kharat, G.** (2015). Real Time Human Pose Recognition Using Leap Motion Sensor. *IEEE International Conference on Research in Computational Intelligence ve Communication Networks (ICRCICN), Kolkata, India*, s. 323-328.

- Martinez, A. M., & Kak, A.** (2001). PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence* , 23, s. 228-233.
- Memiş, A.** (2013). *Kinect RGB Görüntülerde ve Derinlik Haritalarında Uzam-zamansal Özellikleri Kullanarak İşaret Dili Tanıma*. (Yayınlanmamış Yüksek Lisans Tezi). Yıldız Teknik Üniversitesi FBE, İstanbul.
- Memiş, A., & Albayrak, S.** (2013). Turkish Sign Language Recognition Using Spatio-temporal Features on Kinect RGB Video Sequences and Depth Maps. *21st Signal Processing and Communications Applications Conference (SIU), Haspolat, Turkey*, s. 1-4
- Mittal, A., Kumar, P., Roy, P. P., Balasubramanian, R., & Chaudhuri, B. B.** (2019). A Modified LSTM Model for Continuous Sign Language Recognition Using Leap Motion. *IEEE Sensors Journal* , 19 (16), s. 1-8.
- MLP-ELM.** (2021). *MLP-ELM, Extreme Learning Machine for Multilayer Perceptron*. <https://www.mathworks.com/matlabcentral/fileexchange/69813-multilayers-perceptron-based-extreme-learning-machine>. adresinden alınmıştır
- Mohandes, M., Aliyu, S. O., & Deriche, M.** (2014). Arabic Sign Language Recognition using the Leap Motion Controller. *IEEE 23rd International Symposium on Industrial Electronics (ISIE), Istanbul, Turkey*, s. 960-965
- Mohandes, M., Aliyu, S. O., & Deriche, M.** (2015). Prototype Arabic Sign Language Recognition using Multi-Sensor Data Fusion of Two Leap Motion Controllers. *IEEE 12th International Multi-Conference on Systems, Signals & Devices*, s. 1-6
- Mukai, N., Harada, N., & Chang, Y.** (2017). Japanese Fingerspelling Recognition Based on Classification Tree and Machine Learning. *Nicograph International (NicoInt), Kyoto, Japan* , s. 19-24.
- Munib, Q., Habeeb, M., Takruri, B., & Al-Malik, H. A.** (2007). American sign language (ASL) recognition based on Hough transform and neural networks. *Expert Systems with Applications* , 32, s. 24-37.
- Murakami, K., & Taguchi, H.** (1991). Gesture Recognition Using Recurrent Neural Networks. *Proceeding of the SIGCHI Conference on Human Factors in Computing Systems, New Orleans, LA, USA* , s. 237-242.

- Naglot, D., & Kulkarni, M.** (2016). Real Time Sign Language Recognition using the Leap Motion Controller. *International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India*, s. 1-5
- Naidu, C., & Ghotkar, A.** (2016). Hand Gesture Recognition Using Leap Motion Controller. *International Journal of Science and Research (IJSR) ISSN (Online)* , 5 (10), s. 2319-7064.
- Otiniano-Rodriguez, K., & Camara-Chavez, G.** (2014). Finger Spelling Recognition from RGB-D Information Using Kernel Descriptor. *XXVI Conference on Graphics, Patterns and Images, Arequipa, Peru*, s. 92-97.
- Öz, C., & Leu, M. C.** (2011). American Sign Language word recognition with a sensory glove using artificial neural networks. *Engineering applications of artificial intelligence* , 24, s. 1204-1213.
- Öz, C., & Leu, M. D.** (2007). Linguistic properties based on American Sign Language isolated word recognition with artificial neural networks using a sensory glove and motion tracker. *Neurocomputing* , 70, s. 2891-2901.
- Özbay, S., & Safar, M.** (2017). Real-Time Sign Languages Recognition based on Hausdorff distance, Hu invariants and Neural Network. *International Conference on Engineering and Technology (ICET), Antalya, Turkey*, s. 1-8
- Pauchot, J., Tommaso, L. D., Lounis, A., & Benassarou, M.** (2015). Leap Motion Gesture Control With Carestream Software in the Operating Room to Control Imaging. *Installation Guide and Discussion, Surgical Innovation*, s. 1-6
- Porfirio, A. J., Wiggers, K. L., Oliveira, L. E., & Weingaertner, D.** (2013). LIBRAS Sign Language Hand Configuration Recognition Based on 3D Meshes. *IEEE International Conference on Systems, Man, and Cybernetics, Manchester, UK* , s. 1588-1593.
- Pramunanto, E., Sumpeno, S., & Legowo, R. S.** (2017). Classification of Hand Gesture in Indonesian Sign Language System using Naive Bayes. *International Seminar on Sensor, Instrumentation, Measurement and Metrology (ISSIMM), Surabaya, Indonesia*, s. 187-191
- Rahagiyanto, A., Basuki, A., Sigit, R., Anwar, A., & Zikky, M.** (2017). Hand Gesture Classification for Sign Language Using Artificial Neural Network. *21st International Computer Science and Engineering Conference (ICSEC), Bangkok, Thailand* , s. 205-209.

- Rakun, E., Andriani, M., Wiprayoga, I. W., Danniswara, K., & Tjandra, A.** (2013). Combining Depth Image and Skeleton Data from Kinect for Recognizing Words in the Sign System for SIBI. *ICACISIS*, s. 387-392
- Rioux-Maldague, L., & Giguere, P.** (2014). Sign Language Fingerspelling Classification from Depth and Color Images Using a Deep Belief Network. *Canadian Conference on Computer and Robot Vision, Montreal, QC, Canada* , s. 92-97.
- Rojas, R.** (1996). *Neural Networks A Systematic Introduction*. New York: Springer Verlag.
- Saengsri, S., Niennattrakul, V., & Ratanamahatana, C. A.** (2012). TFRS:Thai Fingerspelling Sign Language Recognition System. *Digital Information and Communication Technology and it's Applications (DICTAP), Bangkok, Thailand* , s. 457-462.
- Sarawate, N., Leu, M. C., & Öz, C.** (2015). A real-time American Sign Language word recognition system based on neural networks and a probabilistic model. *Turkish journal of electrical engineering and computer sciences* , 23, s. 2107-2123.
- Silanon, K., & Suvonvorn, N.** (2014). Finger-spelling recognition system using fuzzy finger shape and hand appearance features. *Digital Information and Communication Technology and it's Applications (DICTAP), Bangkok, Thailand* , s. 419-424.
- Sincan, O. M., Tur, A. O., & Keles, H. Y.** (2019). Isolated Sign Language Recognition with Multi-scale Features using LSTM. *IEEE 27th Signal Processing and Communications Applications Conference (SIU), Sivas, Turkey* , s. 1-4.
- Sombandith, V., Walairach, A., & Walairacht, S.** (2017). Recognition of Lao Sentence Sign Language using Kinect Sensor. *14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, s. 656-659.
- Specht, D. F.** (1990). Probabilistic neural networks. *Neural Networks* , 3 (1), s. 109-118.
- Starnes, T., & Pentland, A.** (1995). Real-time American Sign Language Recognition from Video using Hidden Markov Models. *Proceedings of International Symposium on Computer Vision-ISCV, Coral Gables, FL, USA* , s. 265-270.
- Starnes, T., Weaver, J., & Pentland, A.** (1997). A Wearable Computer Based American Sign Language Recognizer. *First International Symposium on Wearable Computers, Cambridge, MA, USA* , s. 130-137.

- Sun, C., Zhang, T., Bao, B.-K., & Xu, C.** (2013). Latent support vector machine for sign language recognition with Kinect. *IEEE International Conference on Image Processing*, s. 4190-4194, Melbourne, VIC, Australia .
- Sun, C., Zhang, T., Bing-Kun Bao, C. X., & Mei, T.** (2013). Discriminative Exemplar Coding for Sign Language Recognition with Kinect. *IEEE transactions on cybernetics*, 43 (5), s. 1418-1428.
- Sutarman, Zain, J. B., Majid, M. B., & Hermawan, A.** (2015). Recognition of Malaysian Sign Language Using Skeleton Data with Neural Network. *International Conference on Science in Information Technology (ICSITech)*, s. 231-236
- Tang, J., Deng, C., Huang, G.-B., & Zhao, B.** (2015). Compressed-Domain Ship Detection on Spaceborne Optical Image Using Deep Neural Network and Extreme Learning Machine. *IEEE Transactions on Geoscience and Remote Sensing* , 53 (3), s. 1174-1185.
- Tangsuksant, W., Adhan, S., & Pintavirooj, C.** (2014). American Sign Language Recognition by Using 3D Geometric Invariant Feature and ANN Classification, Fukuoka, Japan. *The 7th 2014 Biomedical Engineering International Conference (BMEICON)* , s. 1-5.
- Tasmere, D., & Ahmed, B.** (2020). Hand Gesture Recognition for Bangla Sign Language Using Deep Convolution Neural Network. *IEEE 2nd International Conference on Sustainable Technologies for Industry 4.0 (STI), Dhaka, Bangladesh* , s. 1-5.
- TİD.** (2020). *Türk İşaret Dili (TİD) Sözlüğü*. 08,01,2021 tarihinde <https://www.isaretce.com> adresinden alındı
- Ullah, F.** (2011). American Sign Language Recognition System for Hearing Impaired People Using Cartesian Genetic Programming. *The 5th International Conference on Automation, Robotics and Applications, Wellington, New Zealand* , s. 96-99.
- Unutmaz, B., Karaca, A. C., & Güllü, M. K.** (2019). Turkish Sign Language Recognition Using Kinect Skeleton and Convolutional Neural Network. *27th Signal Processing and Communications Applications Conference (SIU)*, s. 1-4.
- Übeyli, E. D., & Güler, İ.** (2004). Dalgacık Dönüşümünün Kullanımı ile Teşhis Sistemleri için Öznitelik Çıkarma. *Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu*, s. 1-5.
- Vachirapipop, M., Tiraronnaku, W., Soymat, S., & Hnoohom, N.** (2017). Sign Translation with Myo armbands. *21st International Computer Science and Engineering Conference (ICSEC), Bangkok, Thailand*, s. 148-152.

- Varsha, M., & Nair, C. S.** (2021). Indian Sign Language Gesture Recognition Using Deep Convolutional Neural Network. *IEEE 8th International Conference on Smart Computing and Communications (ICSCC), Kochi, Kerala, India* , s. 193-197.
- Wang, H., Leu, M. C., & Öz, C.** (2006). American Sign Language Recognition Using Multi-dimensional Hidden Markov Models. *Journal Of Information Science And Engineering*, 22, s. 1109-1123.
- Wang, Z., & Zhang, J.** (2021 ). Continuous Sign Language Recognition based on Multi-Part Skeleton Data. *IEEE International Joint Conference on Neural Networks (IJCNN), Shenzhen, China* , s. 1-8.
- Weerasekera, C. S., Jaward, M. H., & Kamrani, N.** (2013). Robust ASL Fingerspelling Recognition Using Local Binary Patterns and Geometric Features. *International Conference on Digital Image Computing: Techniques and Applications (DICTA), Hobart, TAS, Australia*, s. 1-4
- Wibowo, M. D., Nurtanio, I., & Ilham, A. A.** (2017). Indonesian sign language recognition using leap motion controller. *11th International Conference on Information & Communication Technology and System (ICTS), Surabaya, Indonesia*, s. 67-72.
- Wong, C. M., Vong, C. M., Wong, P. K., & Cao, J.** (2018). Kernel-based Multi-layer Extreme Learning Machine for Representation Learning. *IEEE Transactions on Neural Networks and Learning Systems* , 29 (3), s 757-762.
- Yakut, H.** (2013). *İşaret dili harflerinin görüntü işleme yöntemleriyle Tanınması için bir uygulama.* (Yayınlanmamış Yüksek Lisans Tezi). Fırat Üniversitesi FBE, Elazığ.
- Zhu, X., & Wong, K.-Y. K.** (2012). Single-frame hand gesture recognition using color and depth kernel descriptors. *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), Tsukuba, Japan* , s. 2989-2992.