

**DERİNLİK BİLGİSİ KULLANILARAK İNSAN HAREKETLERİNİN
TANINMASI**

**RECOGNITION OF HUMAN ACTIONS USING DEPTH
INFORMATION**

ALİ SEYDİ KEÇELİ

YRD. DOÇ. DR. AHMET BURAK CAN
Tez Danışmanı

Hacettepe Üniversitesi
Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliğinin
Bilgisayar Mühendisliği Anabilim Dalı İçin Öngördüğü
DOKTORA TEZİ
olarak hazırlanmıştır.

2014

ALİ SEYDİ KEÇELİ'nin hazırladığı “Derinlik Bilgisi Kullanılarak İnsan Hareketlerinin Tanınması” adlı bu çalışma jürimiz tarafından **BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Başkan :.....
(Prof. Dr. Hayri Sever)

Üye (Danışman) :.....
(Yrd. Doç. Dr. Ahmet Burak Can)

Üye :.....
(Doç. Dr. Selim Aksoy)

Üye :.....
(Yrd. Doç. Dr. Nazlı İkizler Cinbiş)

Üye :.....
(Yrd. Doç. Dr. Murat Aydos)

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **DOKTORA TEZİ** olarak onaylanmıştır.

Prof. Dr. Fatma Sevin Düz

Fen Bilimleri Enstitüsü Müdürü

ETİK

Hacettepe Üniversitesi Fenbilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında

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

beyan ederim.

17/12/2014

ALİ SEYDİ KEÇELİ

ÖZET

Derinlik Bilgisi Kullanılarak İnsan Hareketlerinin Tanınması

Ali Seydi Keçeli

Doktora Bilgisayar Mühendisliği

Tez Danışmanı: Yrd. Doç. Dr. Ahmet Burak Can

Aralık 2014

Derinlik algılayıcılar kullanılarak insan hareketlerinin tanınması günümüzde özellikle oyun endüstrisinde giderek yaygınlaşan bir teknolojidir. Derinlik algılayıcıları ortamlara ilişkin 3 boyuttan elde edilen güvenilir öznitelikler sağlayarak kısa mesafelerde doğru hareket tanıma oranını artırmaktadır.

Bu tez kapsamında Microsoft Kinect RGBD algılayıcıdan elde edilen derinlik bilgisini kullanarak insan hareketlerini tanımayı amaçlayan çeşitli yöntemler sunulmaktadır. Hareket tanıma amaçlı olarak ilk önce, bir eklem iskelet modelinden elde edilen açı ve yer değiştirme bilgileri kullanılmıştır. Daha sonra hareketleri zamansal örüntü olarak ele alan Saklı Markov Modelleri ve zaman serilerini temel alan yöntemler üzerine çalışılmıştır. Elde edilen özniteliklerden oluşturulmuş bir sözlük yardımıyla gözlem dizileri haline getirilen hareketler, Saklı Markov Modelleri yardımıyla tanınmaya çalışılmıştır. Hareketler ayrıca zaman serileri olarak ele alınmış ve serilerden elde edilen öznitelikler üzerinde boyut indirilmesi yaparak sınıflama yapan bir yöntem çalışılmıştır. Daha sonra, iskelet modelinden elde edilen nitelikler yanında ham derinlik bilgisi de kullanılarak doğru tanıma oranı yükseltilmeye çalışılmıştır. Son olarak da çalışılan bu yöntemlerden yola çıkılarak düşük gecikmeli hareket tanımaya yönelik bir yöntem önerilmiştir. Geliştirilen yöntemler kendi oluşturduğumuz HUN-3D ve literatürde yer alan ve sıkça kullanılan MSRC-12 ve MSR-Action 3D veri kümeleri üzerinde de test edilmiştir. Geliştirilen yöntemler veri kümelerinden bağımsız olarak kolay hesaplanabilir öznitelikler ile başarılı sonuçlar üretmiştir.

Anahtar Kelimeler: Hareket Tanıma, Örüntü Tanıma, Support Vector Machine, Random Forest, Microsoft Kinect, Derinlik Haritası

Danışman: Yrd. Doç. Dr. Ahmet Burak Can, Hacettepe Üniversitesi, Bilgisayar Mühendisliği Bölümü

ABSTRACT

Recognition of Human Actions Using Depth Information

Ali Seydi Keçeli

Doctor of Philosophy Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Ahmet Burak Can

December 2014

Human action recognition using depth sensors is an emerging technology especially in game console industry. Depth information provides 3D robust features about environments and increases accuracy of action recognition in short ranges.

This thesis presents various approaches to recognize human actions using depth information obtained from the Microsoft Kinect RGBD sensor. In the first studied approach, information about angle and displacement of joints is obtained from a joint skeleton model to recognize actions. Then actions are considered as temporal patterns and studied on Hidden Markov Models and time series. In the Hidden Markov Model based model, actions are converted into observation series by utilizing a vocabulary constructed from the features. Besides actions are considered as time series and actions are classified after applying dimension reduction on features extracted from the series. Then, in addition to features from the skeletal model, features are obtained from raw depth data to increase classification ratio. Finally, combining the experience from all studied methods, a low latency action recognition method is proposed. The constructed models are tested on our own HUN-3D dataset and MSRC-12, MSR-Action 3D datasets, which are widely used in the literature. The proposed approaches produce robust results independent from the dataset with simple and computationally cheap features.

Keywords: Action Recognition, Pattern Recognition, Support Vector Machine, Random Forest, Microsoft Kinect, Depth Maps

Advisor: Asst. Prof. Dr. Ahmet Burak Can, Hacettepe University, Department of Computer Engineering

TEŐEKKÜR

Yazar, bu alıőmanın gerekleőmesinde katkılarından dolayı, aőađıda adı geen kiői ve kuruluőlara itenlikle teőekkür eder.

Sayın Yrd. Dr. Ahmet Burak Can, tez konusunun belirlenmesini sađlamıő, tez alıőmasının ynlendirilmesi ve tez metninin hazırlanması sırasında yol gsterici olmuőtur.

Sayın Do. Dr. Selim Aksoy ve sayın Yrd. Do. Dr. Nazlı İkizler tez izleme komitesinde yer alarak tez alıőmasının ynlendirilmesi ve tez metnini inceleyerek son őeklini almasına katkı sađlamıőtır.

Sayın Prof. Dr. Hayri Sever ve sayın Yrd. Do. Dr. Murat Aydos tez metnini inceleyerek son őeklini almasında katkı sađlamıőtır.

Tez alıőması sırasında, Aydın Kaya ve Hseyin Temuin maddi ve manevi desteklerini esirgememiőtir.

Her daim yanımda olan, sevgi ve desteklerini hi esirgemeyen aileme ve niőanlıma canı gnlden teőekkr ediyorum.

İÇİNDEKİLER DİZİNİ

| | <u>Sayfa</u> |
|---|--------------|
| ÖZET | I |
| ABSTRACT | II |
| TEŞEKKÜR | III |
| İÇİNDEKİLER | IV |
| ŞEKİLLER | VI |
| ÇİZELGELER | VIII |
| SİMGELER VE KISALTMALAR..... | X |
| 1. Giriş..... | 1 |
| 2. İlgili Çalışmalar | 5 |
| 2.1. Tek Kameradan Elde Edilen Görüntülerin Kullanıldığı Yöntemler | 5 |
| 2.2. Birden Fazla Kameradan Elde Edilen Görüntülerin Kullanıldığı Yöntemler | 15 |
| 2.3. RGBD Algılayıcıdan Elde Edilen Verilerin Kullanıldığı Yöntemler..... | 18 |
| 2.4. Hareket Bölütleme | 30 |
| 2.4.1. Sınır Tespiti | 31 |
| 2.4.2. Kayan Pencere | 31 |
| 2.4.3. Dilbilgisi Birleştirme..... | 31 |
| 3. Derinlik Bilgisi Kullanılarak İnsan Hareketlerinin Tanınması | 32 |
| 3.1. İskelet Çıkarma..... | 32 |
| 3.2. Kullanılan Veri Kümeleri | 34 |
| 3.3. Öznitelik çıkarma | 37 |
| 3.4. Hareketlerin Sınıflanması | 42 |
| 3.4.1. Support Vector Machine | 42 |
| 3.4.2. Random Forest..... | 44 |
| 3.5. Sınıflandırma Sonuçları | 45 |
| 3.5.1. HUN-3D Üzerinde Sınıflama Sonuçları..... | 46 |
| 3.5.2. MSRC-12 ve MSR-Action 3D Veri kümeleri için deney sonuçları | 53 |
| 3.6. Sonuçlar | 58 |
| 4. Saklı Markov Modelleri İle Hareket Tanıma | 60 |
| 4.1. Markov Zincirleri | 60 |
| 4.2. Saklı Markov Modeli | 63 |
| 4.2.1. İleri/Geri Algoritması..... | 64 |
| 4.2.2. Viterbi Algoritması..... | 65 |
| 4.2.3. Baum-Welch Algoritması..... | 67 |
| 4.3. SMM ile Hareket Tanıma | 68 |

| | |
|--|-----|
| 4.3.1. K-means Kümeleme | 69 |
| 4.3.2. Sınıflama | 69 |
| 4.4. Sonuçlar | 76 |
| 5. Zaman Serilerinden Öznitelik Çıkarımı İle Hareket Tanıma..... | 77 |
| 5.1. Zaman Serisinden Öznitelik Çıkarımı | 77 |
| 5.1.1. Wavelet Packet Transform | 78 |
| 5.1.2. Bulanık Entropi Tabanlı Boyut İndirgeme | 79 |
| 5.2. Öznitelik Çıkarma ve Sınıflama | 81 |
| 5.3. Sonuçlar | 90 |
| 6. Ham Derinlik Bilgisinden Faydalanılarak Çok Modlu Hareket Tanıma..... | 91 |
| 6.1. Derinlik Haritası Üzerinde Polar Koordinat Sistemi Tanımlama..... | 92 |
| 6.2. Surf İlgili Noktaları Kullanımı | 94 |
| 6.3. Yıldız İskelet Modeli..... | 95 |
| 6.4. Derinlik Geçmiş Haritası | 95 |
| 6.5. Derinlik Gradyan Görüntüsü | 98 |
| 6.6. Sınıflama Deneyleri | 100 |
| 6.7. Sonuçlar | 102 |
| 7. Düşük Gecikme Süreli Hareket Tanıma..... | 103 |
| 7.1. Düşük Gecikme Süreli Hareket Tanıma için Geliştirilen Model..... | 106 |
| 7.1.1. Çerçeve Tabanlı Tanıma | 106 |
| 7.1.2. Hareket Dizisi Tabanlı Tanıma | 108 |
| 7.2. Öznitelikler..... | 109 |
| 7.3. Öğrenme..... | 110 |
| 7.4. Sonuçlar | 113 |
| 8. Sonuç | 115 |
| Kaynaklar..... | 119 |
| ÖZGEÇMİŞ | 127 |

ŞEKİLLER

| | <u>Sayfa</u> |
|--|--------------|
| Şekil 2.1. Örnek MHI verileri ([11]'den alınmıştır) | 7 |
| Şekil 2.2. Örnek MEI verileri ([11]'den alınmıştır) | 7 |
| Şekil 2.3. Yıldız iskelet modeli ([27]'den alınmıştır) | 11 |
| Şekil 2.4. Yıldız iskelet modeliyle tanımlanmış kod kümesi ([27]'den alınmıştır) | 12 |
| Şekil 2.5. Noguchi ve diğerleri'nin [35] yönteminde ilk 3 aşamaya ait akış ([35]'den alınmıştır)..... | 13 |
| Şekil 2.6. Mosift öznelikleri çıkarımı akış çizelgesi..... | 14 |
| Şekil 2.7. DMM hesaplanması ([64]'den alınmıştır) | 24 |
| Şekil 2.8. EigenJointlerin hesaplanması ([68]'den alınmıştır) | 27 |
| Şekil 2.9. FTP Çıkarım Süreci ([70]'den alınmıştır)..... | 30 |
| Şekil 3.1. Shotton ve diğerleri tarafından geliştirilen yönteme ait tanımlı vücut parçaları ([7]'den alınmıştır)..... | 33 |
| Şekil 3.2. Elde edilen insan iskeleti | 34 |
| Şekil 3.3. HUN-3D Veri kümesinde yer alan hareketlere ilişkin örnekler | 36 |
| Şekil 3.4. Hesaplanan açı değerleri | 37 |
| Şekil 3.5. Birim vektörler | 38 |
| Şekil 3.6. Kolları çapraz tutma için histogramlar | 40 |
| Şekil 3.7. Tekme için histogramlar | 40 |
| Şekil 3.8. İskelete ait kalça noktası | 41 |
| Şekil 3.9. El sallama sırasında el bileğinin yer değiştirmesi | 41 |
| Şekil 3.10. Düzlemler ve destek vektörler | 43 |
| Şekil 3.11. Öznelik uzayında boyut genişletme | 44 |
| Şekil 3.12. SVM ve RF yöntemleri için ROC eğrileri | 52 |
| Şekil 3.13. MSRC-12 de Test-1 için sınıflama oranları | 54 |
| Şekil 3.14. MSRC-12 de Test-2 için sınıflama oranları | 55 |
| Şekil 3.15. MSR-Action-3D de Test-1 için sınıflama oranları | 56 |
| Şekil 3.16. MSR-Action-3D de Test-2 için sınıflama oranları | 57 |
| Şekil 4.1. MSRC-12 için kullanılan bir SMM, X'ler durumlar, a'lar geçiş olasılıkları ve b'ler gözlem olasılıklarıdır | 62 |
| Şekil 4.2. SMM ile hareket tanıma | 69 |
| Şekil 5.1. Yöntemin Akış Çizelgesi | 78 |
| Şekil 6.1. Shotton ve diğerleri [7] tarafından geliştirilen iskelet modeli ve derinlik haritası. | 92 |
| Şekil 6.2. Oluşturulan koordinat sistemi..... | 93 |

| | |
|---|-----|
| Şekil 6.3. El sallama hareketine ilişkin DHG hesaplaması..... | 98 |
| Şekil 6.4. El sallama hareketine ilişkin GG hesaplaması | 100 |
| Şekil 7.1. Düşük Gecikme Süreli Hareket Tanıma Yöntemine ait akış çizeneği | 109 |
| Şekil 7.2. Literatürde yer alan diğer yöntemlerle geliştirilen yöntemin karşılaştırması , Li ve diğ. [3], Wang ve diğ.[70] Ellis ve diğ.[88] Zanfir ve diğ [90] | 114 |

ÇİZELGELER

| | <u>Sayfa</u> |
|---|--------------|
| Çizelge 3.1. HUN-3D üzerinde Eklem açıları ile sınıflama sonuçları | 47 |
| Çizelge 3.2. HUN-3D üzerinde Yer değiştirme bilgisi ile elde edilen sonuçlar | 47 |
| Çizelge 3.3. Eklem açıları ve eklem yer değiştirme bilgisinin birlikte kullanıldığı sınıflama sonuçları | 48 |
| Çizelge 3.4. HUN-3D'de Test-2 de SVM modeli için hata matrisi | 49 |
| Çizelge 3.5. RF Modeli için Test-2 de hata matrisi | 50 |
| Çizelge 3.6. Test-2 de farklı çubuk sayıları için sınıflama sonuçları | 50 |
| Çizelge 3.7 Öz-nitelikler üzerinde boyut indirgeme yapıldıktan sonra sınıflama sonuçları | 51 |
| Çizelge 3.8. MSRC-12 veri kümesi için RF ve SM modellerinde sınıflama sonuçları | 53 |
| Çizelge 3.9 MSR-Action 3D veri kümesi için RF ve SM modellerinde sınıflama sonuçları | 56 |
| Çizelge 3.10. Geliştirilen Yöntemin Literatürde yer alan diğer yöntemler ile karşılaştırılması | 57 |
| Çizelge 4.1. MSR-Action 3D nin Test-1 de SMM ile sınıflandırılması sonucu elde edilen hata matrisi | 72 |
| Çizelge 4.2. MSR-Action 3D nin Test-2 de SMM ile sınıflandırılması sonucu elde edilen hata matrisi | 73 |
| Çizelge 4.3. MSRC-12 nin Test-1 de SMM ile sınıflandırılması sonucu elde edilen hata matrisi | 74 |
| Çizelge 4.4. MSRC-12 nin Test-2 de SMM ile sınıflandırılması sonucu elde edilen hata matrisi | 75 |
| Çizelge 4.5. SMM İle Elde Edilen Test Sonuçları | 75 |
| Çizelge 5.1. Zaman serileri kullanılan model için MSR-Action 3D Sınıflama Başarımı | 83 |
| Çizelge 5.2. Test-1 için MSR-Action 3D Random Forest Hata Matrisi | 83 |
| Çizelge 5.3. Test-1 için MSR-Action 3D SVM Hata Matrisi | 84 |
| Çizelge 5.4. Test-2 için MSR-Action 3D Random Forest Hata Matrisi | 85 |
| Çizelge 5.5. Test-2 için MSR-Action 3D SVM Hata Matrisi | 86 |
| Çizelge 5.6. MSRC-12 Sonuçları | 87 |
| Çizelge 5.7. Test-1 için MSRC12 SVM Hata Matrisi | 87 |
| Çizelge 5.8. Test-1 için MSRC12 Random Forest Hata Matrisi | 88 |
| Çizelge 5.9. Test-2 için MSRC12 SVM Hata Matrisi | 88 |
| Çizelge 5.10. Test-2 için MSRC-12 Random Forest Hata Matrisi | 89 |
| Çizelge 5.11. Geliştirilen yöntemin MSR-Action 3D veri kümesi için diğer yöntemlerle karşılaştırılması | 90 |
| Çizelge 6.1. Polar koordinat sistemi ile MSR-Action 3D üzerinde elde edilen sonuçlar | 94 |

| | |
|---|-----|
| Çizelge 6.2. Geliştirilen Yöntemlerin Literatürde yöntemlerle MSR-Action 3D üzerinde RF sınıflayıcısı ile kıyaslaması | 101 |
| Çizelge 6.3. Ham Derinlik Bilgisi İle Elde Edilen Test Sonuçları, SVM için | 102 |
| Çizelge 7.1. Cross Subject Test için harekete ilişkin çerçeve görünme oranlarına göre başarımlar oranları | 114 |
| Çizelge 8.1 Geliştirilen yöntemlerin karşılaştırmalı başarımları | 117 |

SİMGELER VE KISALTMALAR

| | |
|------|-----------------------------------|
| AUC | Area Under Curve |
| CA | Classification Accuracy |
| CCD | Comparative Code Descriptor |
| DTW | Dynamic Time Warping |
| DMM | Depth Motion Map |
| FCM | Fuzzy C-means |
| FFT | Fast Fourier Transform |
| HOG | Histogram of Oriented Gradients |
| HMM | Hidden Markov Model |
| KNN | K-Nearest Neighbour |
| MHI | Motion History Volume |
| MHT | Motion History Template |
| MVT | Motion Volume Template |
| PCA | Principle Component Analysis |
| RF | Random Forest |
| RGBD | Red Green Blue Depth |
| ROC | Receiver Operating Characteristic |
| SIFT | Scale Invariant Feature Transform |
| SDK | Software Development Kit |
| SMM | Saklı Markov Modeli |
| SVM | Support Vecot Machine |
| SURF | Speeded Up Robust Features |
| WPT | Wavelet Packet Transform |

1. Giriş

Hareket tanıma, bilgisayarlı görü alanında 80'li yıllardan itibaren çalışılmaya başlanmış ve günümüzde de yoğun olarak çalışılmaya devam edilen bir konudur. İnternet ağının yaygınlaşması ve görsel algıdaki artış, video ve görüntülerin analizi çalışmalarını arttırmıştır. Özellikle video ve görüntülerdeki hareketlerin belirlenmesi, ilgili çoklu ortam kaynağının etiketlenmesi ve belirli bir hareket ile ilgili video ve görüntülerin aranması konularında hareket tanıma çalışmaları önemlidir. Günümüzde gittikçe artan güvenlik uygulamalarında şüpheli hareketlerin tespiti ve kalabalıkların analizinde, alışveriş mekanlarında müşteri davranışlarının analiz edilmesinde, yalnız yaşayan yaşlı insanların evlerinde oluşabilecek olumsuz durumların tespitinde, radyo terapi sistemlerinde hasta hareketlerinin takibi ve tedavinin doğru uygulanabilmesinde, akıllı ev ve bina sistemlerinde, insan-bilgisayar etkileşimi bağlamında yeni arayüzler tasarlanmasında, etkileşimli oyun uygulamalarında hareket tanıma yöntemleri sıklıkla uygulanmaktadır.

İnsan hareketlerinin tanınması 2 boyutlu videolar üzerinde yaygın olarak çalışılmıştır [10,11,12,15]. Genelde gri seviye değerleri (*intensity*), renk ve hareket tabanlı öznitelikler kullanılarak video görüntüleri üzerinden hareketler tanınmaya çalışılmıştır. Daha sonraları, 2 boyutlu videolardan elde edilen öznitelikler yanında, 3. boyut bilgisinden çıkarılabilecek öznitelikleri kullanmak amacıyla, birden fazla kamera kullanılan çalışmalar yapılmıştır [40,41,42]. Bu yaklaşımda, farklı açılardan çekilmiş görüntülerdeki ortak noktalar bulunup bu noktaların yardımıyla 2 boyuttan 3 boyuta geçiş sağlanmaktadır. Birden fazla kamera kullanılarak elde edilen derinlik bilgisi ile insan hareketleri daha başarılı şekilde tanınabilmektedir. Fakat, bu yaklaşım hesaplama açısından pahalı ve zaman alıcı olduğundan gerçek zamanlı uygulamalar geliştirmeye uygun değildir. Birden fazla kamera kullanma prensibine dayanan, hareket yakalama (*motion capture*) sistemleri de aynı işlevleri yerine getirmelerine rağmen hem çalışma ortamının günlük kullanıma uygun olmaması, hem de ekonomik maliyetinin yüksek olması açısından kısıtlı uygulamalar için uygulanabilmektedir [20].

Derinlik (*RGBD*) algılayıcılarının gelişmesiyle birlikte, derinlik haritalarından elde edilen öznitelikler hareket tanımda yaygınca kullanılmaya başlanmıştır. Bireylerin hareketinin çevrimiçi (*online*) analiz edilmesi gereken uygulamalarda gözlemlenen ortamdaki bireylerin gerçek zamanlı olarak tespiti ve takibi gerekmektedir. Günümüz teknolojileri incelendiğinde, insan bedenindeki tüm uzuvların gerçek zamanlı takibi, derinlik algılayıcılar ile etkin ve hata oranı düşük bir şekilde gerçekleştirilebilmektedir. Derinlik algılayıcılar sıradan bir kamera gibi kullanılabilen ve özel bir düzeneğe ihtiyaç duyulmadan 3 boyutlu bir tarama gerçekleştirebilmektedir. Derinlik algılayıcılar, yaydıkları kızıl ötesi ışınların yansımalarının bir CMOS algılayıcısı tarafından algılanması ile gerçek zamanlı olarak derinlik haritası çıkarılabilmektedir. Bu algılayıcılar görüntüleme menzillerinin düşük olması sebebiyle dış ortamlarda kullanılamazlar da iç ortamlarda hareketlerin tespitinde kullanılabilirler. Derinlik algılayıcıların, birden fazla kamera veya hareket yakalama sistemi kullanan yaklaşımlara göre en büyük avantajı, düşük maliyetle, gerçek zamanlı derinlik haritası hesaplayabilmesidir. Bu nedenlerle, son zamanlarda oyun konsollarında bu teknoloji oldukça yaygınlaşmıştır. Derinlik algılayıcıları ile yapılan hareket tanıma çalışmalarının büyük bir kesimi sadece derinlik bilgisi kullanırken [1,2,3,4], bazı çalışmalarda hem derinlik bilgisi hem de RGB kameralardan elde edilen öznitelikleri beraber kullanılmaktadır [5,6]. Hareketlerin doğru ve hızlı bir şekilde tanınabilmesi, öncelikle kişinin gerçek zamanda tespit ve takibine dayanmaktadır. Derinlik algılayıcılar, insan tespit ve takibin gerçek zamanlı yapılabilmesinde son derece etkin ve hızlıdır [7,8]. Derinlik haritası kullanılarak, ön plan ve arka plan nesnelere kolayca ayırt edilebilmekte ve nesnelere konum bilgileri hesaplanabilmektedir.

Tez çalışması kapsamında, derinlik algılayıcıları kullanılarak insan hareketlerinin otomatik olarak algılanıp sınıflandırılması üzerine çalışılmıştır. Derinlik algılayıcıdan elde edilen veriler kullanılarak, tanınmaya çalışılan insan hareketleri 3 boyutlu olarak modellenmiştir. İlk olarak derinlik haritalarına dayalı bir iskelet modelinden elde edilen öznitelikler kullanılmıştır. Daha sonra hareketlere ilişkin zamansal örüntülerin de tanıma başarımını artırma amacıyla kullanılabileceği düşünülerek, hareketlerden elde edilen zamansal bilgiler kullanılarak Saklı Markov Modelleri ve zaman serilerine dayalı modeller oluşturulmuştur. Zamansal bilgilerden sonra, ham derinlik verileri üzerine başka yöntemler üzerine de çalışılmıştır. Tez çalışmasında en son olarak, yukarıdaki yöntemlerden elde edilen birikimle, düşük gecikmeli hareket tanıma

yöntemleri üzerine çalışılmıştır. Tez çalışması kapsamında, derinlik algılayıcı olarak Microsoft Kinect RGBD algılayıcıdan elde edilen veriler kullanılmıştır. Yöntemlerin test edilmesinde tez çalışması kapsamında oluşturduğumuz HUN-3D veri kümesi yanında, literatürdeki MSR-Action 3D ve MSRC-12 veri kümeleri kullanılmıştır. Tez metninin planı ise şu şekildedir :

Tezin 2. bölümde hareket tanıma konusunda farklı yaklaşımlara sahip literatürdeki çalışmalara değinilmiştir. İlk olarak tek kamera ile hareket tanıma yapılmaya çalışılan yöntemler anlatıldıktan sonra, birden fazla kamera kullanılan yöntemlere değinilmiştir. Daha sonra, derinlik algılayıcılarla yapılan çalışmalar anlatılmıştır. Son olarak hareket bölütleme yöntemlerine değinilmiştir.

3. bölümde iskelet modelinden elde edilen öznitelikler kullanılarak hareketler tanınmaya çalışılmıştır. Microsoft Kinect RGBD algılayıcıdan elde edilen derinlik bilgisi Shotton ve diğerleri [7] tarafından ortaya atılan yöntemle işlenerek hareketi gerçekleştiren kişinin tespit ve takibi yapılmıştır. Bu yöntem, RGBD algılayıcı yardımıyla, vücut eklemlerinin 3 boyutlu konumlarını gerçek zamanlı olarak bulmaktadır. Derinlik haritasındaki her piksele ilişkin derinlik ve pozisyon bilgileri kullanılarak öznitelikler elde edilmiştir. Bu kesimde kullanılan öznitelikler diğer kesimlerde geliştirilen yöntemlerde de kullanılmıştır. Elde edilen öznitelikler kullanılarak Support Vector Machine (SVM) ve Random Forest (RF) modelleri eğitilmiştir. Bu modeller hem kendi oluşturduğumuz ve 4 farklı aktöre ait hareketleri içeren HUN-3D, hem de literatürde yer alan pek çok çalışma tarafından kullanılan MSR-Action 3D ve MSRC-12 veri kümeleri ile test edilmiştir.

4. ve 5. bölümde, hareketlere ilişkin zamansal örüntüler de dikkate alınarak tanıma başarımı artırılmaya çalışılmıştır. 4. bölümde hareketler, birden fazla çerçeveden oluşan bölütlere (*segment*) ayrılmış ve bu bölütler kümelenecek hareket serileri etiket serilerine dönüştürülmüş. Daha sonra bu etiket serileri üzerinde Saklı Markov Modelleri kullanılarak hareketler tanınmaya çalışılmıştır. 5. bölümde hareketler zaman serisi olarak ele alınmış ve bu seriler üzerinden hareketler tanınmaya çalışılmıştır. Zaman serisi üzerinden öznitelik çıkarmaya yarayan bir yöntem yardımıyla hareket serilerinin tümü aynı uzunluktaki öznitelik vektörlerine indirgenmişlerdir. Bu bölümde de iskelet modelinden elde edilen öznitelikler

kullanılmıştır. Zaman serileri ile hareket tanımada yine SVM ve RF sınıflayıcıları kullanılmıştır.

6. bölümde ise ham derinlik haritası üzerinden elde edilen öznitelikler de kullanılarak hareketler tanınmaya çalışılmıştır. İskelet modelinden elde edilen öznitelikler derinlik haritasından elde edilen öznitelikler ile birleştirilerek sınıflama başarımı arttırılmıştır. Bu yöntemde de SVM ve RF sınıflayıcıları kullanılarak hareketler sınıflanmıştır.

Özellikle etkileşimli çevrimiçi uygulamalarda düşük gecikmeli hareketin tespiti ve sınıflaması önem taşımaktadır. 7. bölümde hareket serilerinin tümünü gözlemeden, düşük gecikmeli hareket tanımaya yönelik bir yöntem geliştirilmeye çalışılmıştır. Bu bölümde yapılan çalışmada, hareketin ilk %30'luk kesimini gördükten sonra, %60 düzeyinde bir başarı elde edilmiştir. Bu yöntemde en yüksek başarı Adaboost sınıflayıcısı ile elde edilmiştir.

Son bölümde ise genel olarak tez çalışması kapsamında yapılan deneylerde elde edilen bilgiler irdelenmiş ve gelecekte bu çalışmalara nelerin eklenebileceği üzerinde durulmuştur.

2. İlgili Çalışmalar

İnsan hareketlerinin algılanması günümüzde bilgisayarlı görünün önemli araştırma alanlarından birisidir. Bu alanda hedeflenen, bilinmeyen bir video görüntüsünden devam etmekte olan insan hareketlerinin algılanmasıdır. Bir başka deyişle, devam etmekte olan insan hareketlerinin tespit edilmesi ve gerçekleştirilen hareketin doğru olarak sınıflandırılmasıdır. Özellikle, gözetleme sistemleri, insan bilgisayar etkileşimi, çocuk ve yaşlıların takibi gibi alanlarda insan hareketlerinin takibi ve tanınmasının büyük önemi vardır.

Hareket tanıma konusunda literatürde yer alan çalışmalar 2 ana başlık altında incelenebilir. Bunlar video kamera görüntüleri üzerinden hareket tanıma çalışmaları ve derinlik bilgisin kullanıldığı çalışmalardır. Video görüntüleri üzerinde yapılan çalışmalar eskiye dayanmakla birlikte sayıca da oldukça fazladır. Video kameralara dayalı sistemlerde yine tek kameradan ve birden fazla kameradan elde edilen verileri kullanan yöntemler olarak iki gruba ayrılabilir. RGBD algılayıcıların kullanıldığı yöntemler ise, çalışmalara başladığımız dönemde henüz yeni çalışılmaya başlanmış bir alan olup bu konudaki çalışmalar halen günceldir. Bu iki ana kesim için literatürde yer alan bazı çalışmalar ilerleyen alt kesimlerde açıklanmıştır. İlerleyen alt-kesimde ilk olarak tek kamera kullanılarak hareket tanımaya çalışan yöntemler, sonra birden fazla kamera kullanarak tanımaya çalışanlar ve son olarak da derinlik bilgisi kullanarak tanımaya çalışan yöntemler incelenmiştir.

2.1. Tek Kameradan Elde Edilen Görüntülerin Kullanıldığı Yöntemler

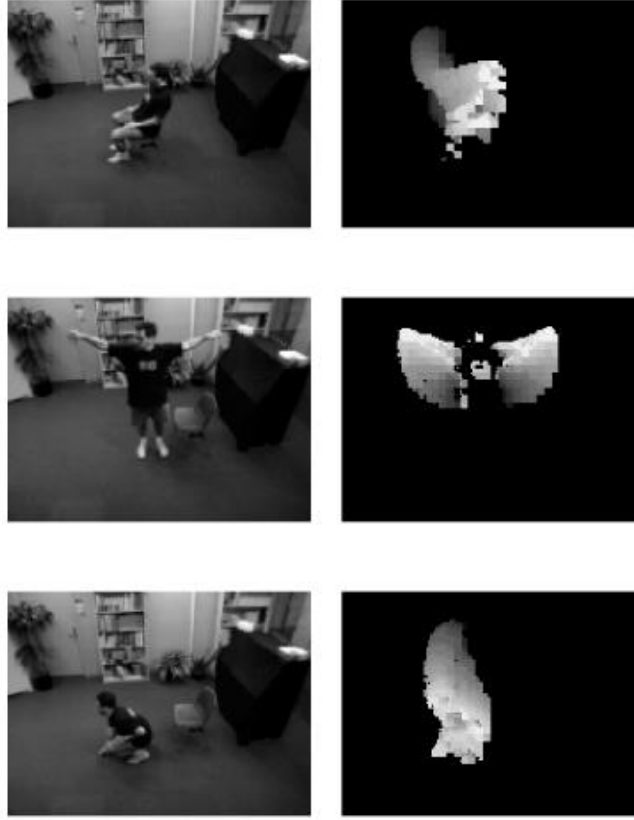
Bu kesimde hareket tanımada kullanılan temel yöntemler üzerinde durulmuştur. Bu yöntemler tek açıdan elde edilmiş video görüntülerini girdi olarak kullanan yöntemlerdir.

İnsan hareketleri temelde 4 gruba ayrılabilir [10]: aktiviteler, etkileşimler, grup aktiviteleri ve el kol hareketleri. Aktiviteler tek kişi tarafından gerçekleştirilen eylemlerdir. Etkileşimler ise bireyin bir başka birey veya nesne ile yaptığı eylemlerdir. Top oynama, boya yapma, çanta taşıma etkileşimlere örnek olarak verilebilir. Grup aktiviteleri ise iki veya daha fazla insan tarafından gerçekleştirilen etkinliklerdir. Grup aktivitelerine örnek olarak toplantı yapma verilebilir. El kol hareketleri ise el sallama

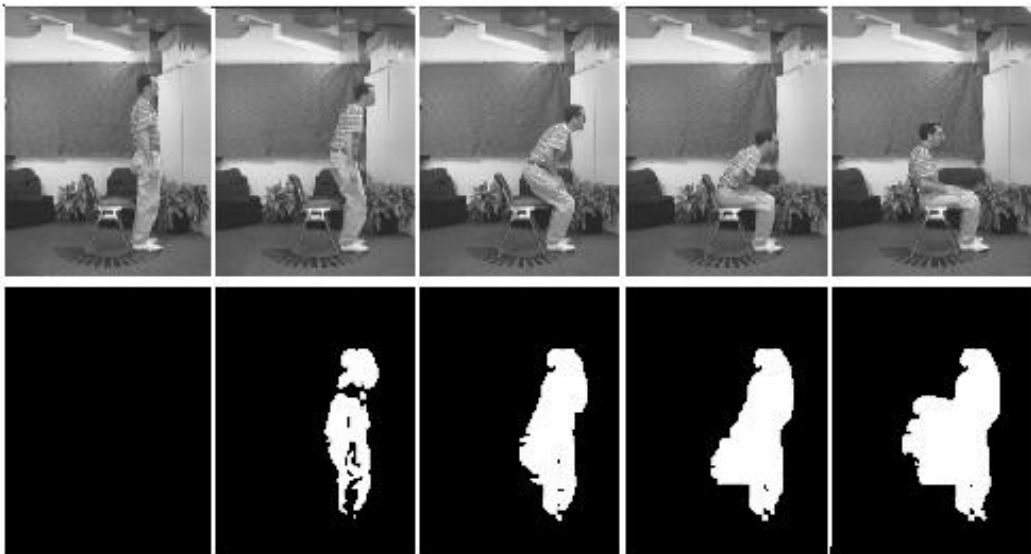
ve çırpma hareketleridir. Literatürde bahsi geçen hareket çeşitleri için çeşitli çalışmalar mevcuttur. Fakat video görüntüsü kullanan bu çalışmalar büyük video aranması, getirilmesi ve etiketlenmesi konusuna odaklanmaktadır.

Aggarwal ve Ryoo'ya göre, insan hareketlerinin algılanması yöntemleri temelde iki gruba ayrılır [10]. Bunlar tek katmanlı ve hiyerarşik hareket tanıma yöntemleridir. Tek katmanlı yöntemler doğrudan hareket tanımaya dönük çalışmalardır. Çok katmanlı yöntemler ise önce basit yada atomik hareketleri tanıdıktan sonra tanınan atomik hareketler yardımıyla karmaşık hareketleri tanıyan yöntemlerdir. Tek katmanlı yöntemlerde yine kendi içinde iki gruba ayrılır: uzay-zaman yaklaşımları ve sıradüzensel yaklaşımlar. Uzay-zaman yaklaşımları da yine kendi aralarında 3 gruba ayrılırlar: uzay-zaman (*space-time approaches*), takip-izi (*trajectories*) ve uzay-zaman öznitelikleri (*space-time features*). Uzay-zaman yaklaşımlarında girdi olarak verilen iki boyutlu video görüntüsü 3 boyutlu olarak ele alınır. Burada 3. boyut zamandır. Bu yöntemler, genel olarak insan pozunun tanınması ve basit hareketlerin algılanmasında kullanılmaktadır. Kesimin ilerleyen kısımlarında, literatürde yer alan yöntem türlerine ait örneklerden bahsedilmektedir

Bobick ve Davis [11] tarafından geliştirilen yöntemde videoya ait iki boyutlu görüntü dizisi kullanılarak MHI (*Moiton History Image*) ve MEI (*Motion Energy Image*) görüntüsü oluşturulmuştur. MHI'da silüetin önce bulunduğu konumlar, daha düşük gri seviye değerleri ile temsil edilirken silüetin sonlara doğru bulunduğu konumlar daha yüksek gri seviye değerleri ile temsil edilmektedir. MEI'da ise silüetin geçtiği tüm noktalar aynı gri seviye değerleri ile gösterilmektedir. Oluşturulan bu görüntülere daha sonra şablon eşleştirme (*template matching*) uygulanarak ilgili hareket tanınmaya çalışılmıştır. Bu yöntemin etkileşimli uygulamaları da bulunmaktadır (Kreşlerde çocukların gösterilen hareketi tekrarlayıp tekrarlamadıklarını kontrol eden bir çeşit oyun). Şekil 2.1'de örnek videodan elde edilmiş MHI verileri görülmektedir. Solda video görüntüsü sağda ise elde edilen MHI'lar görülmektedir. Şekil 2.2'de ise örnek başka bir videodan elde edilmiş MEI verileri görülmektedir. Üst satırda video görüntüleri, alt satırda ise elde edilen MEI'ler görülmektedir.



Şekil 2.1. Örnek MHI verileri ([11]'den alınmıştır)



Şekil 2.2. Örnek MEI verileri ([11]'den alınmıştır)

Shechtman ve Irani [12], 3 boyutlu uzay-zaman hacim (*space-time volume*) verilerinden hareket akışlarını (*motion flow*) hesaplayıp hareket analizinde kullanmıştır. Hareketin tanınması için yine MHI yaklaşımında [2] olduğu gibi şablon eşleştirme yapılmıştır. Oluşturulan hacimdeki her nokta (x,y,t) için bir parça çıkarılmış. Çıkarılan her parçadan akış (*flow*) değerleri hesaplanmıştır. Elde edilen bu akış değerleri ile şablona ait değerler karşılaştırılarak hareket tanınmaya çalışılmıştır.

Ke ve diğerleri [13] tarafından geliştirilen yöntemde videoya ait görüntüler önce mean-shift algoritması kullanılarak bölütlenmiştir. Bölütleme işlemi her çerçeve için değil, videodan elde edilen 3 boyutlu hacimsel veri üzerinde yapılmıştır. Elde edilen bu 3 boyutlu şekiller akış içinde bilgi içermektedir. Hareket tanıma işlemi, elde edilen şeklin, tanıma için oluşturulan modellere yakınlığı bulunarak yapılmıştır. Şekle ait öznitelikler ile akışa ilişkin öznitelikler SVM algoritması ile sınıflandırılmasıyla hareketin tanınması sağlanmıştır. Kullanılan öznitelikler, eğitim kümesindeki tüm şablonlarla korelasyon (*correlation*) yapılarak elde edilmiştir. Eğer n tane şablon varsa, eğitim kümesinin bunların hepsiyle korelasyonu sonucunda n sütunlu bir eğitim verisi elde edilmiştir. Elde edilen bu veriyle SVM algoritması eğitilmiştir.

İnsan hareketlerinin tespitinde kullanılan diğer bir yaklaşım türü ise uzay-zaman izleridir. Bu yöntemlerde, hareketi yapan insana ilişkin belirli noktalar takip edilip, bu noktaların hareketlerinin oluşturduğu izler kullanılarak hareket tanınmaya çalışılır. Sheikh ve diğerleri [14] tarafından geliştirilen yöntemde, insan hareketleri, eklem noktalarının 4 boyutlu (x,y,z,t) hareketi olarak ifade edilmiştir. Görüntü üzerinde dönüşüm yapılarak hareket eden kişinin eklemlerinin 3 boyutlu hareketi modellenmiştir. Daha sonra elde edilen eklem noktalarının 3 boyutlu (x,y,t) hareketleri ile önceden oluşturulmuş modeller arasındaki açı farkları kullanılarak hareket tanınmıştır. Gözlemlenen 3 boyutlu hareket şekli ile modellenen hareketlerin şekli arasındaki açının Cos değeri hesaplanmış ve maksimum değerinde elde edildiği modele ilişkin hareketin yapıldığı kabul edilmiştir.

Uzay zaman öznitelikleri bağlamında Laptev ve Lindeberg [15] tarafından geliştirilen yöntemde harekete bağlı olarak değişiklik göstermeyen ilgi noktaları (*interest points*) kullanılarak hareket tanınmaya çalışılmıştır. İlk olarak genelde nesne tanıma için kullanılan ilgi noktaları çıkartılmış ve daha sonra bu ilgi noktaları görüntüye ait uzay-zaman hacminde aranmıştır. İlgi noktalarının çıkarılmasında Harris kenar bulma

yöntemi kullanılmıştır [16]. Bu noktaların uzay zaman hacminde aranmasıyla farklı hareket örüntüleri tespit edilebilmiştir. Daha sonra benzer hareketlerin tespiti için, ilgi noktaları kümelenmiştir. Kümeleme için K-means algoritması kullanılmıştır. Hareket türü ise, hangi kümeden ilgi noktaları içerdikleri bilgisiyle elde edilmektedir.

Lublinerman ve diğerleri [17] insan hareketlerini Linear Time Invariant (LTI) olarak modelleyen bir sistem öne sürmüştür. Bu sistem, videoya ait görüntü dizisini silüet dizisine çevirmiş ve iki tip öznitelik çıkarmıştır. Bu öznitelikler, silüetin genişliği ve Fourier tanımlayıcılarıdır. Bir hareket, bu özniteliklerdeki değişiklikler olarak tanımlanmıştır. Hareket süresince her kare için öznitelikler çıkarılıp bir matriste toplanmıştır. Daha sonra matris üzerinde SVD (Single Dalue Decomposition) uygulanmıştır. Hareketlerin sınıflandırılmasında SVM algoritması kullanılmıştır. Bu çalışmada yavaş yürüme, hızlı yürüme, topla yürüme ve eğilerek yürüme hareketleri tanınmıştır.

Shao ve Gao [18], zamana bağlı uzaysal (*spatio temporal*) ilgi noktalarını bularak hareketleri tanımaya çalışmıştır. Bu yöntemde, ilk olarak görüntü dizisi üzerine bir fonksiyon (response function) uygulanmış ve fonksiyonunun yüksek değerler ürettiği noktalar ilgi noktaları olarak seçilmiş ve 3 boyutlu video dizisinde bu noktayı ve çevresindeki noktaları içeren bir kübik parça çıkarılmıştır. Çıkarılan kübik parçaların ortogonal düzlemlerine 2 boyutlu wavelet dönüşümü uygulanmıştır. Wavelet dönüşümü sonucu elde edilen dönüşüm katsayıları öznitelik olarak kullanılmıştır. Çıkarılan öznitelikler daha sonra SVM algoritması kullanılarak sınıflandırılmıştır. KTH veri seti [40] üzerinde %94'e yakın başarı elde edilmiştir.

İkizler ve Duygulu [19] video görüntülerinde insan pozlarını farklı açı ve büyüklüklere sahip dikdörtgenler histogramı olarak tanımlamıştır. Bu çalışmada ilk olarak görüntüde yer alan silüet bilgisi bir izleyici (*tracker*) [20] kullanılarak bulunmuş ve daha sonra bulunan silüet üzerindeki dikdörtgensel bölgeler çıkarılmaya çalışılmıştır. 12 farklı açıda filtre görüntü üzerinde gezdirilmiş ve düşük değer döndüren kesimler elenmiştir. Alınan dikdörtgensel kesimlerin açılarına göre histogram çıkarılmıştır. Histogram çıkarma işleminde silüet 3x3'lük parçalara ayrılmış her parça için ayrı ayrı histogram çıkarılıp daha sonra bu histogramlar birleştirilmiştir. DWT, SVM ve en yakın komşu yöntemleri kullanılarak insan hareketleri sınıflandırılması ve tanınması sağlanmıştır.

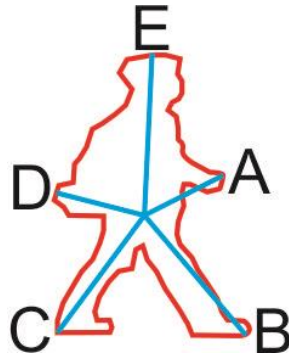
Rajpoot [21] 3 boyutlu wavelet dönüşümünü kullanarak, bir video görüntüsünde yürüyüş yapan insanları tanımıştır. Bu çalışmada ilk olarak, görüntünün belirli bir zaman aralığındaki ortanca değeri alınarak arka plan bilgisi çıkarılmıştır. Daha sonra çıkarılan bu arka plan ile görüntü üzerinde hareket eden insan silüetleri bulunmuştur. Silüet çıkarma işleminden sonra morfolojik aşındırma işlemiyle gürültü giderilmiştir. Daha sonra görüntüdeki her kare için, çıkarılan silüet bilgisinden elde edilen 3 boyutlu (x,y,t) dizisi üzerinde wavelet dönüşümü uygulanmıştır. En küçük alt bant kodlaması öznitelik olarak kullanılmıştır. Daha sonra elde edilen nitelikler üzerinde Principle Component Analysis (PCA) [22] ve Linear Discriminant Analysis (LDA) [23] boyut indirgeme yöntemleri uygulanmıştır. Son olarak çıkarılan öznitelikler eğitim kümesindeki örnekler ile Euclid uzaklığı kullanılarak karşılaştırılmış ve tanıma yapılmıştır.

Min ve Kasturi [24], el ve ayakların hareket izlerini kullanarak temel bale hareketlerini tanımıştır. Görüntülerde yer alan piksellerin optik akış (*optic flow*) değerleri bulunmuştur. Baskın değerlere sahip noktalarındaki renk dağılımları göz önünde tutularak hareket izleri çıkarılmıştır. Görüntü, tespit edilen iz sayısına göre parçalara bölünmüştür. Aynı sayıda iz içeren bölütler aynı kümede yer almıştır. Her iz için x ve y yönlerindeki elemanları içeren bir birim vektör hesaplanmış ve öznitelik olarak kullanılmasıyla bir SMM oluşturulmuştur. Oluşturulan bu model kullanılarak sağ el hareketi, sol el hareketi, ayak hareketleri tanınmıştır.

Gupta ve Davis [25], insan hareketlerinin tanınmasında, kullanılan nesnelerin tanınmasından faydalanmıştır. Bu çalışmada, insan hareketi tanıma ve nesne tanıma yaklaşımları birbirlerini tamamlar biçimde kullanılmıştır. Örneğin şekil olarak birbirleriyle benzer özellik gösteren su ve sprej şişeleri, bu nesnelere kullanan kişilerin, nesnelere kullanarak yaptıkları hareket bilgileriyle (sprej kullanan kişinin şişeyi sallaması, su şişesini kullanan kişinin şişeden içmesi) seçilmiştir. Benzer şekilde görüntü olarak birbirleriyle aynı örüntüye sahip sallama hareketlerinin amacı, uygulandığı nesnenin bilgisi kullanılarak (sprej şişesi veya bayrak) ayrıştırılmıştır. Nesnelerin tanınmasında Histogram of Oriented Gradients (HOG) yaklaşımından faydalanılmıştır. Bu yöntemde, görüntüdeki gradyan değerlerine ilişkin bir histogram çıkarılmakta ve çıkarılan bu histogramdaki değerler öznitelik olarak kullanılmaktadır. Çıkarılan öznitelikler ile Adaboost algoritması [26] kullanılarak nesnelere tanınmıştır. Videolardaki insan hareketleri üç kısma ayrılmıştır: erişim, uygulama ve nesne

tepkisi. Eriřim hareketi, seilen grnt bařlangıcından, znenin nesne ile etkileřime getięi ana kadarki kısmı kapsar. Uygulama hareketi, eriřim hareketinden sonraki ereveden bařlar ve znenin nesneyi kullanarak gerekleřtirdięi hareketi (*manipulation*) kapsar. Eriřim ve uygulama hareketleri daha nce gerekleřtirilen el yrngesi algoritmasını kullanmaktadır. Bu algoritma SMM kullanarak vcuttaki uzuvları bltlemekte ve uzuv hareketlerine gre hareketi tanımlamaktadır. Nesne tepkisi hareketi, uygulama hareketinden sonra bařlar ve uygulanan eylemden sonra nesne zerinde oluřan deęiřimleri ierir.

Yuan ve dięerleri [27] nin alıřmasında video grntleri zerinden silet ıkarıp yıldız iskelet modeli zerinden znelikler ıkarılarak hareketler tanınmaya alıřılmıřtır. Yıldız iskelet modeli arka plan ıkarma ile tespit edilen silet zerinden gerekleřtirilmiřtir. Bu iřlemden ilk olarak elde edilen silete ait merkez nokta bulunmuřtur. Daha sonra silet eperinde yer alan noktaların bu merkez noktasına olan uzaklıkları hesaplanmıřtır. Bu noktalar ierisinde yerel maksimumlar bulunarak yıldız iskelet oluřturulmuřtur. Őekil 2.3.'te rnek bir yıldız iskelet modeli yer almaktadır.

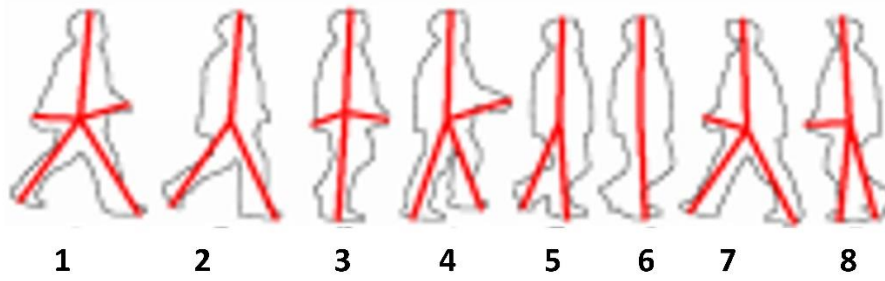


Őekil 2.3. Yıldız iskelet modeli ([27]'den alınmıřtır)

Yıldız iskelet modelinden znelik olarak yıldız modelinde yer alan noktaların x ve y koordinatları kullanılmıřtır. SVM kullanılarak tm pozlar sınıflanmıřtır. Belli anahtar pozlar tanımlanmıř ve poz kmesi SVM sınıflayıcısı ile sınıflanarak hareket bir poz dizisi haline getirilmiřtir. Poz dizisinden hareket tanıma iinse String (katar) karřılařtırma yntemlerinden [28] faydalanılmıřtır.

Yıldız iskelet modeli kullanan dięer bir yntem ise Chen ve dięerleri [29] tarafından geliřtirilmiřtir. Geliřtirilen bu yntemde ise yıldız iskelet modeli poz tanımlayıcı olarak

kullanılmıştır. Elde edilen her poz önceden tanımlanmış bir kod kümesindeki öğelerle karşılaştırılmış ve hareket bir kod dizisi haline getirilmiştir. Tanımlanan kod kümesi Şekil 2.4.'te gösterilmektedir. Şekilde yer alan her poz bir kod ile eşlenmiştir. Hareketin kod dizisi haline getirilmesinden sonra tanıma işleminde Saklı Markov Modeli (SMM) kullanılmıştır. Bu çalışma kapsamında yine peşi sıra gelen hareketlerin tanınabilmesi amacıyla bir yöntem önerilmiştir. Önerilen yöntem kayan pencere (*sliding window*) yöntemine dayanmaktadır. Hareketler son 30 çerçeve baz alınarak tanınmaktadır. Kısacası T anındaki bir hareket tanınırken T-W kümesindeki çerçeveler kullanılmaktadır. W pencere boyutunu belirtmektedir.



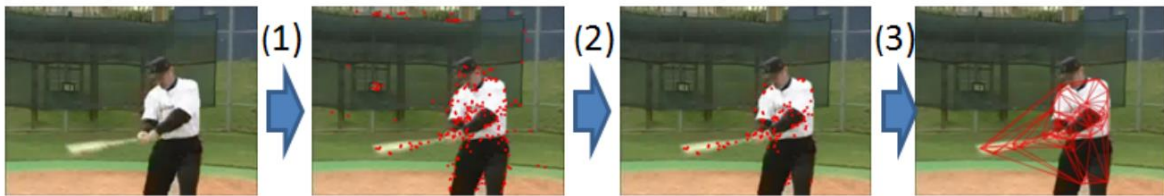
Şekil 2.4. Yıldız iskelet modeliyle tanımlanmış kod kümesi ([27]'den alınmıştır)

Şimdiye kadar anlatılan yöntemler, basit aktivitelerin tanınmasında kullanılan yöntemlerdir. Karmaşık hareketlerin tanınmasında hiyerarşik yöntemler kullanılmaktadır. Bu yöntemlerde, tanınan basit hareketleri temel olarak karmaşık hareketler tanınmaya çalışılır. Nguyen ve diğerleri [30], hareket tanıma için çok katmanlı bir SMM (*Hidden Markov Model*) önermiştir. Önerilen bu modelde, normal SMM'den farklı olarak her durum kendi içinde başka bir SMM'dir. Özetle olarak kapalı bir ortamda hareket eden kişilerin takip ettiği yollar kullanılmıştır. Oda belli kesimlere ayrılmış ve odadaki eşyaların hangi kesimlerde buldukları işaretlenmiştir. Kişinin belirli bir sırada uğradığı kesimler bilgisi kullanılmıştır. Alt SMM'ler ile basit hareketler tanınmış (tv izleme, buzdolabını açma), daha sonra bu bilgiden yola çıkarak daha üst düzey hareketler (kısa yemek, atıştırıp çıkmak) tanınmaya çalışılmıştır. Benzer bir çalışma yine aynı şekilde Zhang ve diğerleri [31] tarafından grup etkinliklerinin tanınması için yapılmıştır. Yapılan çalışmada, konuşma, yazma, bekleme temel faaliyetleri alt düzey SMM'ler kullanılarak tanınmıştır. Sonrasında üst düzey SMM ile sunum, tartışma, monolog gibi daha üst düzey aktiviteler tanınmaya çalışılmıştır.

Shi ve diğeri [32] tarafından geliştirilen yöntemde, P-net (*Propagation Network*) adı verilen SMM benzeri bir yapı kullanılmıştır. P-net yönteminde bir hareket SMM'de olduğu gibi birden fazla durumla ve bunlar arasında geçiş ve gerçekleşme olasılıkları olarak ifade edilmiştir. P-netin SMM'den farkı P-nette aynı anda birden fazla duruma geçişe izin verilmesidir. Bu çalışmada da diğeri hiyerarşik çalışmalarda olduğu gibi hareket birkaç temel harekete bölünmüş ve bunlardan karmaşık hareketi tanımlayan bir ağ oluşturulmuştur.

Ivanov ve Bobick [33] SMM'nin yanı sıra, bağlamdan bağımsız dil bilgisi (*context free grammar*) de kullanmışlardır. Çalışmaları iki kesimden oluşmaktadır. İlk kesimde SMM kullanılarak temel hareketler tanınmıştır. İkinci katmanda ise tanınan bu hareketlerden oluşturulan bir dil bilgisine ait katar (*string*) işlenerek hareket tanınmaya çalışılır. Minnen ve diğeri'nin yaptığı çalışmada da [34], karmaşık insan hareketlerinin tanımlanmasında bağlamdan bağımsız dil bilgisi kullanılmıştır.

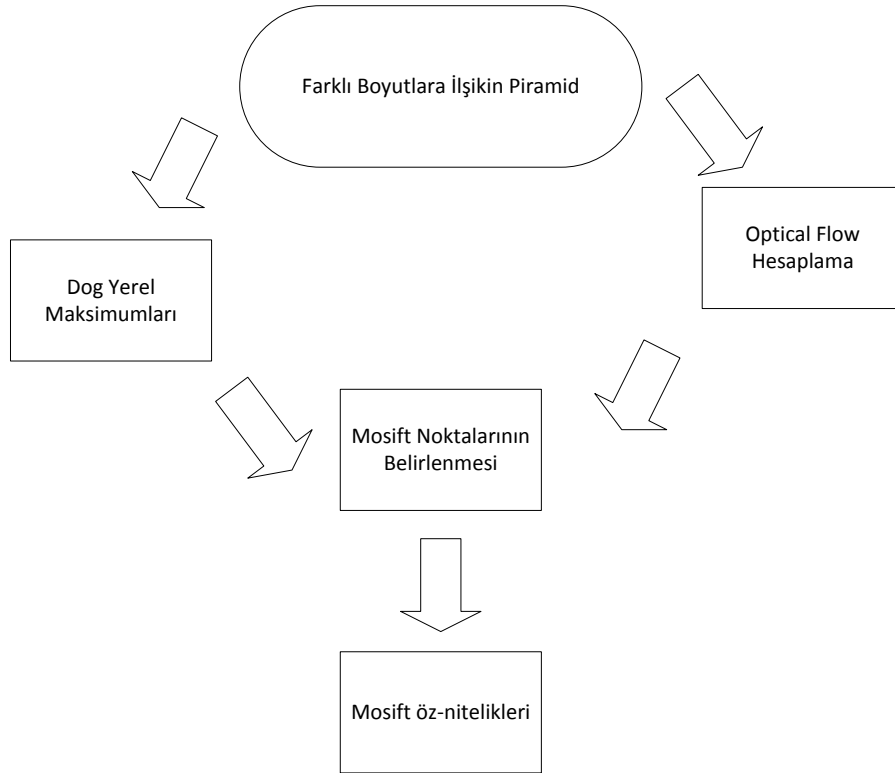
Noguchi ve diğeri [35] tarafından geliştirilen yöntem, video görüntüleri üzerinde ilgi noktalarının tespiti ve bu noktaların hareketinden faydalanarak insan hareketlerini tanımaya çalışmıştır. Bu yöntemde ilk olarak video görüntüsü üzerinde SURF [36] ilgi noktaları çıkarılmıştır. Daha sonra çıkarılan ilgi noktalarına ait hareket vektörleri Lucas-Kanade [37] optik flow yöntemi kullanılarak hesaplanmıştır. Optik Flow değerleri hareketli noktaların seçiminde kullanılmaktadır. Optik flow değerleri ilk çerçeve (*frame*) ve N/2 inci çerçeveler arası için hesaplanmıştır. N burada video da yer alan çerçeve sayısını belirtmektedir. Bir sonraki aşamada optik flow değerleri kullanılarak seçilen hareketli ilgi noktaları arasında Delunay üçgenlemesi yapılmıştır. Bahsedilen aşamalara ilişkin akış Şekil 2.5.'te gösterilmiştir.



Şekil 2.5. Noguchi ve diğeri'nin [35] yönteminde ilk 3 aşamaya ait akış ([35]'den alınmıştır)

Optik flow deęerleri kullanılarak yerel hareket znitelikleri bulunmuştur. Bu znitelikler Lucas-Kanade [37] yöntemine yre hesaplanan 5 boyutlu hareket matrisidir. Tm ereveler iin ıkarılan bu deęerler birleştirelmiştire. Son olarak yerel hareket znitelikleri ve SURF ilgi noktalarından elde edilen znitelikler birleştirelerek bir Multiple Kernel Learning (MKL) sınıflayıcısı eęitilmiştire. MKL-SVM yönteminde kullanılan doęrusal ekirdeęi (*linear kernel*) ait aęırlık deęerlerini tahmin etmekte ve bu aęırlık deęerlerini kullanan bir SVM modeli ile hareketler tahmin edilmeye alıřılmaktadır.

Chen ve dięerleri [38] tarafından el hareketleri ile tv kontrol iin nerilen yntem yine aynı řekilde ilgi noktalarının tespitine dayanmaktadır. nerilen yntemde ilgi noktalarının tespitinde Mosift [39] yntemi kullanılmaktadır. Mosift yntemi esasında SIFT yntemine dayanmaktadır. Fakat ilgi noktalarının seilmesinde Optical Flow deęerlerini de gz nnde bulundurmasından dolayı hareketli ve dndrme ve boyut deęiřiklięinden baęımsız noktaların seilmesini saęlamaktadır. Mosift ilgi noktalarının hesaplanmasına iliřkin akıř řekil 2.6.'da verilmiştire. DoG (*Difference of Gaussians*) hesaplaması SIFT te olduęu gibidir. Fakat SIFT duraęan resimler iin geliřtirilmiştire. Mosift videolar iin geliřtirildięinden Optical Flow deęerleri hesaplanmıřtır



řekil 2.6. Mosift znitelikleri ıkarımı akıř izelgesi

Optical Flow deęerleri videoda tespit edilen SIFT noktalarının etrafına karřılık gelen kboidler zerinde hesaplanmaktadır. Kısacası yeterince hareketli SIFT noktaları seęilmektedir. zniteliklerin ıkarılmasında ise her ilgi noktasının etrafında yer alan gradyan (*gradient*) byklę ve yn kullanılmıřtır. Nokta etrafındaki blge 16x16 lık blmlere ayrılmıř, bu blmlerde 4x4 lk gridler olarak gruplanmıřtır. Her gride ait 8 ubuklu bir histogram ıkarılmıřtır. Sonu olarak 4x4x8 toplamda 128 elemanlı bir znitelik vektr oluřturulmuřtur.

Mosift noktaları tespit edildikten sonra hareketin tespiti iin tespit edilen noktalar K-means algoritması kullanılarak kmelenmiřtir. Kmeleme iřlemi ile bir szlk (*code book*) elde edilmiřtir. Her video iin ierdięi szcklerin histogramı ıkarılmıřtır. ıkarılan bu szck histogramı ile eęitilen bir SVM modeli sınıflama iin kullanılmıřtır. Mosift gerek zamanlı kullanıma olanak saęlamadıęı iin paralel bir gerekleřtirmesi yapılmıřtır.

2.2. Birden Fazla Kameradan Elde Edilen Grntlerin Kullanıldıęı Yntemler

Bu kesimde farklı aılardan elde edilmiř grntler ve derinlik bilgisinin kullanıldıęı yntemler zerinde durulmuřtur. Birden fazla kamera kullanılmasıdaki ama hareket iliřkin 3 boyutlu bilgi edinilebilmesidir. nceden bahsedildięi zere birden fazla kamera yardımıyla derinlik haritası ıkartılabilmesi mmkndr. Ayrıca birden fazla kamera yardımıyla hareket 3 boyutlu bir Őekil olarak da modellenenebilir.

Weinland ve dięerleri [40], farklı aılardan ekilmiř grntleri kullanarak yine 3 boyutlu bir grsel gvde (*visual hull*) elde etmiř ve bu 3 boyutlu grsel gvdeye ait Motion History Volume (MHV) verisi oluřturmuřtur. MHV verisi, MHI verisinin 3 boyutlu hali gibi dřnlebilir. Fakat bu alıřmada hareketin tanınmasında doęrudan Őablon karřılařtırma kullanılmamıřtır. MHV verisinde yer alan noktalar zerinde geometrik dnřm uygulanmıř ve Őekil silindirik bir koordinat sistemine oturtulmuřtur. Daha sonra MHV zerinde Fast Fourier Transform (FFT) uygulanmıřtır. Silindirik koordinat sistemindeki konumlar ve FFT dnřmnde belli frekanslar iin elde edilen deęerler (*magnitude*) znitelik olarak kullanılmıřtır. Tanıma iřlemi, Őablona ait znitelikler ile girdi olarak gelen znitelikler karřılařtırılarak yapılmıřtır. Karřılařtırma iin Euclid uzaklıęı kullanılmıřtır.

Pehlivan ve Duygulu [41] tarafından geliştirilen yöntemde farklı açıdan çekilmiş görüntüler kullanılarak hareket eden kişinin 3 boyutlu hacimsel görüntüsü çıkarılmıştır. Daha sonra çıkarılan 3 boyutlu görüntüde yer alan silüet, y ekseninde belli aralıklarla dilimlere ayrılmıştır. Elde edilen dilimler ikili görüntüye dönüştürülmüştür. Her dilimde yer alan ikili maskeyi çevreleyen dışsal çember ve maske içine sığan en büyük çember bulunmuştur. Daha sonra bu çemberlerin sayısı ve alanları öznitelik olarak kullanılarak hareket tanınmaya çalışılmıştır. Her poza ait öznitelikler, hareket matrisi adı verilen yapılarda toplanmıştır. Hareket matrisleri SVM sınıflama algoritması ve en yakın komşu yöntemi kullanılarak sınıflandırılmıştır. Sınıflamada ilk olarak, SVM ile yapılan hareketin üst gövde veya alt gövdeye hareketi olup olmadığı belirlenmiştir. Daha sonra, örneklem kümesinde yer alan hareketler ile karşılaştırma yapılarak hareket tanınmıştır.

Roh ve diğerleri [42] tarafından geliştirilen yöntemde, stereo kameralardan alınan görüntüler kullanılarak 3 boyutlu MHI benzeri bir hacimsel veri elde edilmiştir. Elde edilen hacimsel veriden daha sonra MHI yaklaşımında olduğu gibi şablonlar elde edilip, şablon eşleştirme ile hareket tanınmaya çalışılmıştır. İlk olarak stereo görüntüden elde edilen derinlik haritası kullanılarak ön planda yer alan nesnelere arka plandan ayırt edilmiştir. Daha sonra yine derinlik haritasından faydalanılarak hacimsel veri hesaplanmıştır. Daha sonra MHI benzeri olan Volume Motion Template (VMT) verisi hesaplanmıştır. Farklı anlardaki VMT verileri arasındaki fark kullanılarak hareketin yönelimi bulunmuştur. Daha sonra iki VMT farkından elde edilen hareket vektörü üzerinde dönüşüm yapılarak 3 boyuttan iki boyuta indirgeme (xy düzlemine dönüşüm) yapılır. Elde edilen iki boyutlu görüntüye PMT denilmektedir. En son aşamada, PMT verileri kullanılarak hareket tanınmaktadır.

Huang ve diğerleri [43], çoklu kameradan elde edilmiş hacimsel veri üzerinde, 3 boyutlu silindirik şekil bağlamında poz tanımaya yönelik bir yöntem önermiştir. Bu yöntemde, kameralardan gelen görüntüler üzerinde ilk olarak bölütleme yapılmıştır. Elde edilen bölütlerden, 3 boyutlu şekiller oluşturulmuştur. Oluşturulan 3 boyutlu şekil çok katlı bir silindire yerleştirilmiş ve vokseller silindirik histogramda farklı çubuklara (bin) düşmüştür. Elde edilen histogram ve Discrete Hidden Markov Model (DHMM) kullanılarak insan pozları tanınmıştır.

Gu [44] , 15 eklemli bir insan modeli kullanarak hareketleri tanımlamıştır. Bu çalışmada farklı açılardan çekilmiş görüntülerden oluşan XMAS [45] ve INRIA [46] veri kümeleri kullanılmıştır. İlk olarak, farklı açılardan elde edilmiş görüntülerden 3 boyutlu noktalar kümesi (*point cloud*) elde edilmiştir. Daha sonra, durağan haldeki kişi görüntülerinden elde edilen noktalar kümesinden, modele ilişkin pozisyon, yükseklik ve yön bilgileri Karhunen-Loeve (KL) [47] algoritmasıyla tahmin edilir. Bir sonraki adımda gövde bölütlemesi yapılır. Gövde bölütlemesi yapılırken ilk olarak her nokta modelde kendine en yakın olan gövde bölümüne atanır. Bu işlem sırasında bazı noktalar yanlış gövde kısımlarına atanabilirler. Gövde bölütlerinin kenarlarında yer alan bu noktalar belirsiz noktalar olarak etiketlenir ve bu noktalar üzerinde kümeleme yapılır. Vücut 10 tane silindirik ve 1 tane küresel veri kümesi olarak ele alınır. Gövde bölütleri bulunduktan sonra eklem yerleri kolaylıkla elde edilmiştir. Çıkarılan bölütler daha sonra Adaptive Particle Filter [48] kullanılarak takip edilmiştir. Oluşturulan insan modelinden konum, yükseklik, yön, tüm vücuda ait eklemlerin konumları ve üst bacak eklemlerinin konumları bilgisi alınarak hareket tanımda kullanılmıştır. Çıkarılan bu öznitelik dizileri ve iki katmanlı bir SMM sınıflayıcısı kullanılarak hareket tanınmaya çalışılmıştır. İlk katmandaki sınıflama sonucuna göre üst katmanda ağırlık verilecek öznitelikler belirlenmektedir.

Pehlivan ve Duygulu [49] tarafından yapılan bir çalışmada, yine farklı açılardan alınmış görüntülerden elde edilen 3 boyutlu hacimler kullanılmıştır. Oluşturulan 3 boyutlu hacim üzerinde farklı yön açalarına ve büyüklüklere sahip silindirik yapılar bulunmaya çalışılmıştır. Bu amaçla farklı yerel eksenlere sahip silindirlere bir grup çekirdek (*kernel*) oluşturulmuştur. Oluşturulan çekirdeklerden yüksek tepki verenlerin sayısı (histogramlar) insan pozunu tanımlamak için kullanılmıştır. Başarımını artırmak için oluşturulan hacim, düşey ekseninde N tane eşit parçaya bölünmüş ve bu parçalar üzerindeki silindir sayıları bulunmaya çalışılmıştır. Poz tanıma için K en yakın komşu (*K-nearest neighbor*, KNN) ve SVM yöntemleri kullanılmıştır. Hareket tanıma içinse hareketler poz dizisi olarak kabul edilmiş ve tanıma için DTW ve SMM kullanılmıştır. DTW ile silindirlere ilişkin histogramların aynı konumlarındaki dalgalanmaları takip edip, en iyi eşlemeyi bulmaya çalışılır. SMM de ise hareketi tanımlayan pozlar K-means kullanılarak kümelenebilir ve her küme bir kodlama sözcüğü ile eşleştirilmiştir. Oluşturulan bu kodlama sözcükleri ile bir SMM oluşturulmuştur.

Ferrer ve diğeri [50], çok sayıda kamera kullanarak bir odada bulunan birden fazla kişiye ilişkin hareketleri analiz etmiştir. Bu çalışmada ilk olarak kameralardan gelen görüntülerde ön planda yer alan nesnelere çıkarılmıştır. Bir sonraki adımda elde edilen ön plan görüntülerinden 3 boyutlu şekiller oluşturulmuştur. Bu şekillerden Motion History Volume (MH) ve Motion Energy Volume (MEV) oluşturulmuştur. Aynı zamanda kişi silüetleri elipsoid şeklinde modellenmiş ve elipsoidin 3 eksenindeki eksen uzunlukları ve dönüş değerlerinden faydalanılarak vokseller sağ-sol, el-ayak olarak sınıflandırılmıştır. MHV ve MEV'ye ait değişimsiz istatistiksel momentleri [51] hesaplanmıştır. Modelden, uzuvların etrafındaki hareket voksellerinin miktarı ve moment değerleri öznitelik olarak kullanılmıştır. Çıkarılan öznitelikler Bayesian sınıflandırıcısı ile kullanılarak, insan hareketleri tanınmaya çalışılmıştır.

Kohen ve Li [52], farklı açılardan alınan görüntülerdeki 2 boyutlu silüetlerden 3 boyutlu bir görsel gövde (*visual hull*) oluşturup, bu gövdenin şeklini tanımlayan özniteliklerden kişinin pozunu tanımlamıştır. Gövde oluşturulduktan sonra gövdeyi çevreleyen bir küre tanımlanır. Kürenin ağırlık merkezi gövdeyi oluşturan voksellerin ağırlık merkezidir. Küre yüzeyinde bazı noktalar seçilir ve gövde üzerindeki her nokta ve seçilmiş küre noktalarına göre uzaklığı bulunmuştur. Uzaklığa ilişkin açı değerleri de hesaplanır. Hesaplanan bu değerlere ait bir histogram oluşturulur. Histogramdaki her kutu farklı açı ve uzaklık ikililerine karşılık gelmektedir. Elde edilen histogramın sütun değerleri öznitelik olarak kullanılarak SVM sınıflandırıcısının eğitilmesinde kullanılmıştır. Gözlemlenen yeni görüntülerden elde edilen histogramlar bu sınıflandırıcıdan geçirilerek insan pozları tanınmıştır.

2.3. RGBD Algılayıcıdan Elde Edilen Verilerin Kullanıldığı Yöntemler

Son senelerde yaygınlaşan 3 boyutlu derinlik algılayıcıların, insan hareketlerinin tanınması, ortam haritasının çıkarılması gibi alanlarda kullanımı günümüzde gittikçe yaygınlaşmaktadır. Bu kesimde RGBD algılayıcıların kullanıldığı yöntemler üzerinde durulmuştur. Özellikle günümüz oyun konsollarındaki etkileşimli oyunlarda, insan hareketlerinin tanınarak oyunun yönlendirilmesi konusunda bir tür derinlik algılayıcısı olan RGBD algılayıcılar kullanılmaktadır. RGBD algılayıcıların en büyük avantajı gerçek zamanlı olarak derinlik haritası (görüntüde yer alan nesnelere uzaklık bilgilerinin tutulduğu matris) elde edebilmeleridir.

RGBD algılayıcı kullanarak insanın hareketlerinin algılanması ve insanın modellenmesi bilgisayarlı görü alanında nispeten yeni bir alandır [1,2,7,8]. Genelde önceki yöntemlere göre uygulaması daha kolaydır ve başarılı sonuçlar alınabilmektedir. Özellikle birden fazla RGBD algılayıcı kullanarak, görüş açısından bağımsız insan hareketlerini tanıma konusunda çalışmalar henüz yenidir. Derinlik bilgisi üzerinden hareket tanımada aşılması gereken ilk sorun hareketi gerçekleştiren kişinin silüetinin çıkarılmasıdır. Bu sorun pek çok çalışmada Shotton ve diğerleri [7] tarafından ortaya atılan ve Kinect SDK ile gerçekleştirilmiş olan yöntemler ile aşılmıştır. Bu yöntem RGBD algılayıcı yardımıyla, vücut eklemlerinin 3 boyutlu konumlarını gerçek zamanlı olarak bulmaktadır. Derinlik haritasındaki her piksele ilişkin derinlik ve pozisyon bilgileri kullanılarak öznitelikler elde edilmiştir. Önceden tanımlanmış derinlik haritaları üzerinde etiketlenmiş vücut kesimlerine ait öznitelikler kullanılarak Random Forest (RF) yöntemi ile eğitilmiştir. Eklemlerin konumlarının tespitinde ilk olarak vücut kesimlerinin tanınması yapılmıştır. Daha sonra eklemlerin bulunması için 2 boyutlu koordinatlar ve derinlik haritası kullanılarak 3 boyutlu koordinatlara dönüşüm yapılmış ve mean-shift [9] algoritması kullanılarak konumlar bulunmuştur.

RGBD algılayıcılar giriş kesiminde bahsedildiği üzere, gerçek zamanda hareket tanıma da yaygın olarak kullanılmaya başlanmıştır. RGBD algılayıcılar ile çalışılırken gerçekleştirilmesi gereken adımlardan ilki, derinlik haritasındaki insan silüetlerinin çıkarılmasıdır. Xia ve Aggarwal [2] tarafından yapılan çalışmada RGBD algılayıcıdan (Kinect) elde edilen derinlik haritasındaki insan silüeti bulunmaya çalışılmıştır. İlk olarak elde edilen derinlik haritası üzerinde median filtrelemesi yapılarak gürültü giderilmiştir. Daha sonra derinlik haritasının kenar bilgileri Canny kenar bulma algoritması [53] kullanılarak çıkarılmıştır. Önceden oluşturulan bir kafa şablonu, ikili kenar maskesi üzerinde gezdirilerek 2 boyutlu Chamfer karşılaştırılması [54] yapılmıştır. Belli bir eşik değerin üzerinde sonuç elde edilen noktalar için, o derinlikte olabilecek tahmini bir insan kafasının boyu hesaplanmıştır. Eğer hesaplanan bu değer, belirlenen aralıkta ise o bölgeye ait kenar bilgisi çıkarılmıştır. Luber [8] ve diğerleri HOG [55] yaklaşımını derinlik bilgisi üzerinde uygulanmış ve kişileri tespit edildikten sonra takip işleminde is Multi-Hypothesis Tracker (MHT) yöntemi kullanılmıştır [56].

Doliotis ve diğeri [5] el hareketlerin takibi için derinlik bilgisinin kullanıldığı bir yöntem önermişlerdir. Önerilen yöntemde ilk olarak deri ve hareket tanıma ile el tespit edilmiştir. Deri tanıma için RGB değerlerden oluşan bir histogram kullanılmıştır. Hareket tanıma da ise, her çerçevedeki gri seviye piksel değerlerin bir sonraki çerçevedekilerle farkından yararlanılmıştır. Renk ve hareket bilgileri içinde bir skor değeri tanımlanmıştır. Deri için skor değeri piksele ilişkin rengin deri rengi olma olasılığı, hareket için skor değeri ise piksel değerlerindeki farktır. İki skurun toplamının en fazla değeri aldığı alt pencereler (*sub-window*) el konumu olarak tanınmıştır. Bir sonraki aşamada derinlik bilgisi kullanılmıştır. En büyük 5 bileşen (*connected component*) alınmış ve bu bileşenlerden derinlik bilgisi en az olan seçilmiştir. Önceki adımda bulunan alt pencerelerden seçilen bileşen üzerinde yer alan pencere seçilerek el konumu doğru olarak belirlenmiştir. Son olarak el konumuna ilişkin konum bilgileri (x,y) DTW (Dynamic Time Warping) [57,58] algoritmasında kullanılarak el hareketleri tanınmaya çalışılmıştır.

Li [3], Kinect RGBD algılayıcıdan alınan derinlik haritasının kartezyen düzlemler (x-y,x-z,y-z) üzerinde projeksiyonunu almıştır. Alınan bu projeksiyonların kenarları çıkartılıp, elde edilen kenar noktalarına en yakın, derinlik haritasındaki 3 boyutlu noktalar seçilmiştir. Toplamda 80 adet nokta seçilmiştir. Seçilen bu noktalar poz tanımlamak için kullanılmıştır. Hareket tanıma içinse, hareket çizelgesi tanımlanmıştır. Bu çizelgede, hareketler belirli bir poz dizisi olarak tanımlanmıştır. Çizelgede her harekete ait belirli bir poz dizisi, bu pozlar arasındaki geçiş olasılıklarını belirten bir geçiş matrisi bulunmaktadır. Tanınan pozlar öncelikle kümelenmiştir. Her küme öne çıkan bir poz biçimini belirtmektedir. Hareketler bu öne çıkan pozların bileşimidir. Daha sonra gözlenen değerlerin belirli bir poz olma olasılığı hesaplanır ve hangi poz için en büyük olasılık değerini veriyorsa o poza atama yapılır.

Sung [1] tarafından yapılan çalışmada RGBD algılayıcı kullanılarak insan hareketleri tanınmaya çalışılmıştır. Hareket tanıma da 3 tip öznitelik kullanılmıştır. Bunlar gövde duruşu öznitelikleri, el konumu öznitelikleri ve hareket bilgisidir. Bu öznitelikler, RGBD algılayıcı yardımıyla oluşturulan 15 eklemlilik iskelet kullanılarak çıkarılmıştır. İskelet çıkarma işlemi için PrimeSense [59] iskelet çıkarma uygulaması kullanılmıştır. Gövde duruşu öznitelikleri, el ve ayaktaki eklemlerin gövde ile olan açılarıdır. El konumu öznitelikleri ise sol ve sağ ele ait eklemin, son 60 çerçevede dikeyde en yüksek ve en

düşük konumları ile iki elin kulak, baş ve gövdeye göre konumlarıdır. Hareket öznitelikleri, 11 eklemin koordinatlarının zamana bağlı olarak değişimidir. Son 3 saniyedeki 9 çerçevede eklemlerin dönüşüm bilgileridir. Daha sonra hiyerarşik bir SMM kullanılarak hareketler tanınmıştır. İlk olarak elde edilen poz bilgilerinden birinci katmanda alt aktivite tanınmıştır. Üst katmanda ise alt aktiviteleri kullanarak daha karmaşık aktiviteler tanınmıştır.

Popa [6], Kinect RGBD algılayıcı kullanarak alışverişle ilgili kişi hareketlerini tanımlamıştır. Geliştirilen sistemde RGBD algılayıcı ve kamera birlikte kullanılmıştır. Sistem hem kameradan hem de algılayıcıdan gelen bilgiler ışığında hareketi tanıyıp, iki taraftan gelen bulguları birleştirmektedir. İki kesimde de ilk olarak hareketi tanınacak olan kişi tespit edilmektedir. Kamerada arka plan çıkarma ile, RGBD algılayıcıda ise derinlik haritası kullanılarak kişi silüeti elde edilmektedir. Kameradan elde edilen silüet üzerinde moment değerleri [60] hesaplanmış ve bu değerler öznitelik olarak kullanılmıştır. Kamera görüntüsünden elde edilen silüet çerçeveleri arasındaki farklar alınarak hareket tanımlayıcı öznitelikler elde edilmiştir. Art arda gelen silüet görüntülerinin farkları alınarak hareket matrisi oluşturulmuştur. Bu matriste sabit kalan yerler 0, hareketin başladığı yerler -1, kişinin hareket ettiği yerler ise 1 değeri ile belirtilmiştir. Hareket matrisi içinde moment değerleri hesaplanmıştır. Hareket matrisine benzer bir başka matris de RGBD algılayıcı kullanılarak çıkarılmıştır. Fakat burada ikili silüet yerine derinlik haritası kullanılmış ve kişinin algılayıcıdan uzaklaşma ve yaklaşma bilgisi kaydedilmiştir. Belirtilen öznitelikler çıkarıldıktan sonra hareketin tanınması işlemine geçilmiştir. Tanıma işlemi için SVM, KNN, LDC ve SMM algoritmaları kullanılmıştır. Alışverişle ilgili arama, deneme, inceleme, alma ve sepete atma işlemleri tanınmıştır. En iyi sonuçlar LDC sınıflandırıcısı ile elde edilmiştir.

Raptis ve diğerlerinin çalışmasında [61], Microsoft Kinect sistemini kullanarak dans figürleri tanınmaya çalışılmıştır. Bu çalışmada Shotton ve diğerleri [7] tarafından geliştirilmiş olan iskelet modelleme yöntemi kullanılmış ve iskelet modelinden elde edilen nitelikler kullanılarak dans figürleri tanınmaya çalışılmıştır. Gövdeye ilişkin eklemlerin hareketleri gözlemlenmiş ve gözlemlenen konum değerleri üzerinde PCA algoritması [22] uygulanmış ve (u,r,t) bileşenleri çıkarılmıştır. Bunun dışında 1. (bilek ve dizler) ve 2. dereceden (eller ayaklar) eklemler kullanılarak öznitelikler çıkarılmıştır. Bu öznitelikler sol bileğin merkezden uzaklığı, sol omuzun ve sol bileğin

u ve r bileşenleri ile yaptığı açılarıdır. Yine aynı şekilde sol bilek ve sol elin r bileşeni ile yaptığı açı değerleri hesaplanmıştır. Belirtilen öznitelikler üzerinde kademeli bir sınıflayıcı ve DTW algoritması [58] kullanılarak figürler tanınmıştır.

Cheng ve diğerleri [62] tarafında geliştirilen yöntemde renk ve derinlik bilgisi birlikte kullanılarak günlük hareketler tanınmaya çalışılmıştır. Çalışmada ilk olarak veri üzerinde bir ön işleme yapılmış ve görüntülerde yer alan hareketlerin başlama ve bitiş noktaları manuel olarak belirlenmiştir. Derinlik haritası üzerinde hareketleri tanımlanabilmesi için Comparative Coding Descriptor (CCD) yöntemi önerilmiştir. Bu yöntem, hacimsel derinlik verisi üzerinde göze çarpan (*salient*) noktaların tespiti ve bu noktalar etrafından alınan küboidlerin karşılaştırılmasına dayanmaktadır. Seçilen noktalara ilişkin öznitelik çıkarımı Eşitlik 2.1’de belirtilmiştir.

$$\varphi(p_i) = \begin{cases} 1, & D(p_i) - D(p_r) \geq \gamma \\ -1, & D(p_i) - D(p_r) \leq -\gamma \\ 0, & \text{else} \end{cases} \quad (2.1)$$

r referans noktasını, i küboid içindeki nokta numarasını, D(p) ise derinlik değerini belirtmektedir. γ değeri ise karşılaştırma eşik değeridir. Eşitlik 2.1 de küboid içindeki noktalar üzerinde dolaşarak CCD öznitelik vektörü çıkarıldıktan sonra bu vektörlerin karşılaştırılması için bir uzaklık ölçütü tanımlanmıştır. Küboidlerden elde edilen CCD nitelikleri arasındaki benzerlikler bu ölçüt kullanılarak belirlenmiştir. Bu ölçüt kullanılarak öznitelikler K-medoids kümeleme yöntemiyle kümelenebilirlerdir.

Harekete ilişkin video ve derinlik görüntüleri üzerinden elde edilen öznitelikler birleştirilerek tanıma yapılmıştır. Video görüntüleri üzerinde STIP yöntemi ile ilgi noktaları tespit edilmiş. Bu noktalar etrafında yer alan kesimler için HOF ve HOG öznitelikleri çıkarılmıştır. Derinlik haritası üzerinde ise stip noktalarına karşılık gelen noktalar etrafında CCD öznitelikleri çıkarılmıştır. Çıkarılan bu öznitelikler kümelenecek bir sözlük oluşturulmuş ve Bag of Words yöntemiyle hareketler tanınmaya çalışılmıştır.

Hsieh ve diğerleri [63] tarafından geliştirilen yöntemde video görüntülerinde yer alan insan silüetine ait noktalar bir polar koordinat sistemi üzerine oturtulmuş ve bu polar koordinat sistemi belli bölgelere ayrılmıştır. Silüete ait noktalardan hangi bölgelerde buldukları sayılmış ve bir histogram oluşturulmuştur. Vücudun orta noktasını

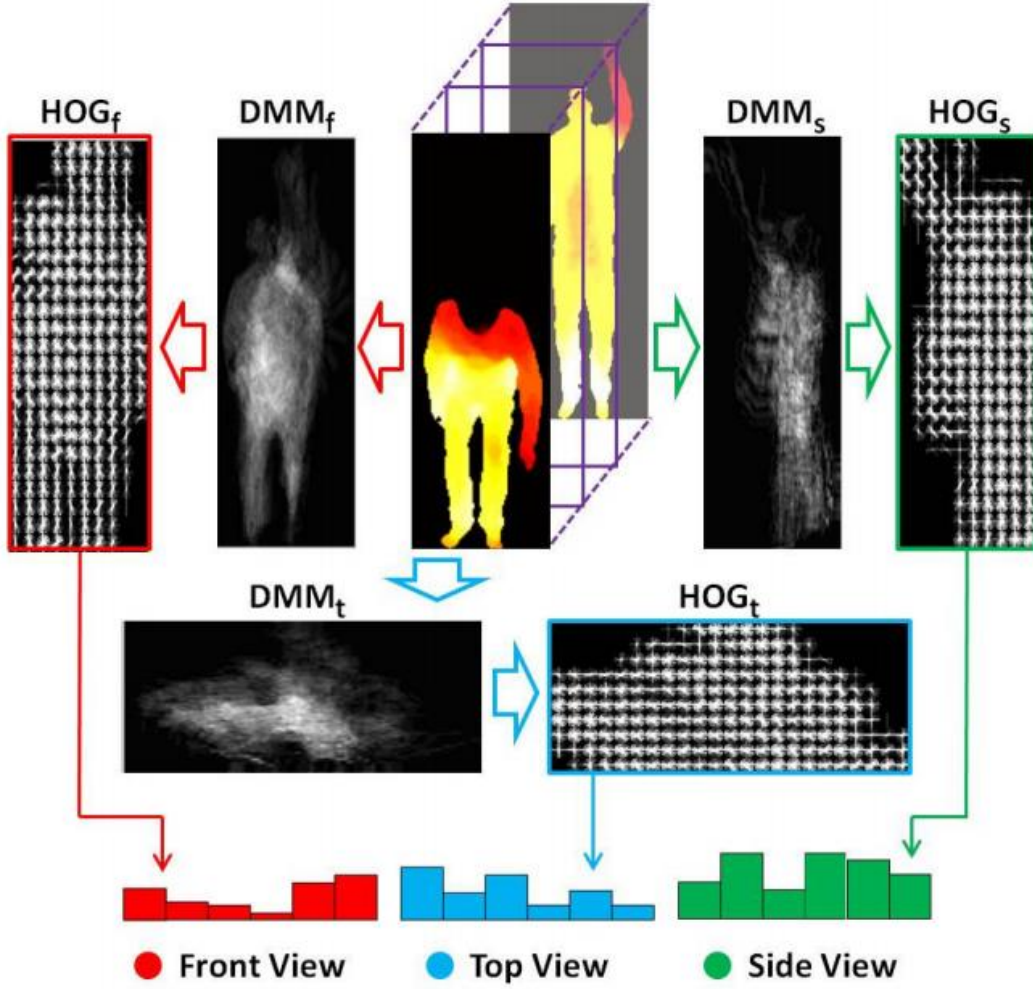
merkez alan bir polar koordinat sistemi yanında, silüete ait alt ve üst kesimlerin orta noktasını da merkez alan polar koordinat sistemleri tanımlanmış ve alt ve üste kesimler içinde histogramlar oluşturulmuştur. Oluşturulan histogramlar birleştirilerek hareketler tanınmaya çalışılmıştır. Bahsedilen öznitelikler video görüntülerinde yer alan tüm çerçeveler için çıkarıldığından boyut indirgeme işlemine gidilmiştir. Boyut indirgeme için PCA yönteminden faydalanılmıştır. Sınıflayıcı olarak da KNN yönteminden faydalanılmıştır.

Yang ve diğerleri [64] tarafından yapılan çalışmada derinlik bilgisi üzerinden çıkarılan öznitelikler kullanılarak hareketler tanınmaya çalışılmıştır. Bu yöntem derinlik görüntüleri üzerinden Depth Motion Map (DMM) oluşturulması ve oluşturulan DMM üzerinden HOG (Histogram of Oriented Gradients) özniteliklerini çıkarılmasına dayanmaktadır. Elde edilen HOG öznitelikleri ile eğitilen bir SVM modeli ile tanıma yapılmıştır.

DMM oluşturulması, peş peşe gelen derinlik haritaları arasındaki farkların toplanması ile yapılmaktadır. Hareketi ifade eden derinlik haritası serisinin ön, yan ve üst projeksiyonları alınıp, her projeksiyon için ayrı ayrı DMM Eşitlik 2.2. kullanılarak hesaplanmıştır.

$$DMM_v = \sum_{i=1}^{N-1} |map_v^{i+1} - map_v^i| > \varepsilon \quad (2.2)$$

$v \in \{f, s, t\}$ projeksiyonu, i indis numarasını belirtmektedir. DMM hesaplanırken belirli bir ε eşik değerinin üzerinde yer alan farklar dikkate alınmıştır. Projeksiyonlar için DMM hesaplanması Şekil 2.7.'de gösterilmektedir. Şekilde DMM ye ait tüm yüzeylerden elde edilen HOG'lar gösterilmiştir. Daha sonra hesaplanan DMM'ler için HOG öznitelikleri çıkartılmıştır.



Şekil 2.7. DMM hesaplanması ([64]'den alınmıştır)

HOG özniteliklerinin çıkarılması aşamasında 23x10 ayırık hücre ve 8 gradyan oryantasyonu kullanılmıştır. Elde edilen HOG öznitelikleri ile bir SVM modeli eğitilmiş ve MSR-Action3D ile test edilmiştir. Bu yöntemde Cross-Subject Test için %86 başarı oranına erişilmiştir.

Xia ve Aggarwal [65] tarafından geliştirilen yöntemde derinlik haritaları kullanılarak 3 boyutlu bir hacim oluşturulması, bu 3 boyutlu hacim üzerinde, STIP yönteminden türetilmiş DSTIP yöntemiyle ilgi noktalarının (*interest points*) bulunmasına dayanmaktadır. Bu yöntemde ilk olarak derinlik haritalarından 3 boyutlu bir hacim oluşturulmuştur. Hacim, derinlik haritalarının peş peşe eklenmesiyle elde edilmiştir. Daha sonra DSTIP yöntemine geçilmiştir. Bu yöntemde ilk olarak Eşitlik 2.3 gösterildiği üzere hacim üzerinden gaussian filtrelemesi yapılmıştır.

$$D_s(x, y, t) = D_s(x, y, t) * g(x, y|\sigma) \quad (2.3)$$

Eşitlik 2.3'te * konvülyasyon işlemini belirtmektedir. Daha sonra düzeltilmiş hacim üzerinde Eşitlik 2.5 te verilen gabor filtresi uygulanmıştır.

$$D_{st}(x, y, t) = D_s(x, y, t) * h(t|\tau, \omega) . s \quad (2.4)$$

$$h(t|\tau, \omega) = e^{-t^2/2\tau^2} . e^{2\pi i \omega t} \quad (2.5)$$

Eşitlik 2.5 te '.' matris çarpımını s ise düzeltme fonksiyonunu belirtmektedir. D_{st} elde edilmesinden sonra ilgi noktalarının çıkartılmasına geçilmiştir. Bu amaçla ilk olarak D_{st} üzerindeki tepki (*response*) R Eşitlik 2.6 ile hesaplanmıştır.

$$R(x, y, t) = \|D_{st}(x, y, t)\|_2^2 \quad (2.6)$$

Eşitlik 2.5 aşağıdaki eşitlikler ile açılarak ifade edilebilir:

$$R(x, y, t) = (D * g * h_{ev} . s)^2 + (D * g * h_{od} . s)^2 \quad (2.7)$$

$$h_{ev}(t|\tau, \omega) = \cos(2\pi \omega t) e^{-t^2/2\tau^2} \quad (2.8)$$

$$h_{od}(t|\tau, \omega) = \sin(2\pi \omega t) e^{-t^2/2\tau^2} \quad (2.9)$$

İlgi noktaları R üzerinde yerel maksimumların bulunmasıyla elde edilir. İlgi noktalarının bulunmasından sonra her nokta etrafından bir küboid çıkarılmıştır. Çıkarılan küboidler bloklara yarıştırmış. Ayrıştırılan her blok için içerdikleri derinlik piksel değerlerine ilişkin bir histogram oluşturulmuştur. İki blok (p,q) arasındaki benzerlik Eşitlik 2.10 ile ifade edilmektedir.

$$S(p, q) = \sum_{n=1}^M \sqrt{h_p^{(n)} h_q^{(n)}} \quad (2.10)$$

Öznitelik vektörü tüm blok kombinasyonlarının benzerliklerinin peş peşe eklenmesiyle oluşturulur. Oluşturulan öznitelik vektörü bir küboidi tanımlamaktır.

Daha sonra Bag of Words yöntemi kullanılmıştır. Öznitelikler K-means algoritması kullanılarak kümelenebilir ve bir sözlük (*codebook*) oluşturulmuştur. Veri kümesinde yer alan tüm hareketler sözcük dizisi olarak tanımlanmış ve eğitilen bir SVM modeli

ile tanınmaya çalışılmıştır. MSR-Action3D veri kümesi üzerinde Cross-Subject Test ile 89.3% doğru tanıma oranına erişilmiştir.

Oreifej ve diğerleri [66] hareket tanımda yalnızca derinlik bilgisini kullanan bir diğer yöntemi önermişlerdir. Önerilen yöntemde derinlik haritası dizileri histogram olarak tanımlanmıştır. İlk olarak derinlik haritalarından 4 boyutlu bir hacim ($V(x,y,z,t)$) oluşturulmuştur. Sonra Eşitlik 2.11 kullanılarak bu hacme ilişkin matematiksel normal değerleri hesaplanmıştır.

$$n = \nabla S = \left(\frac{\partial z}{\partial x}, \frac{\partial z}{\partial y}, \frac{\partial z}{\partial t} \right)^T \quad (2.11)$$

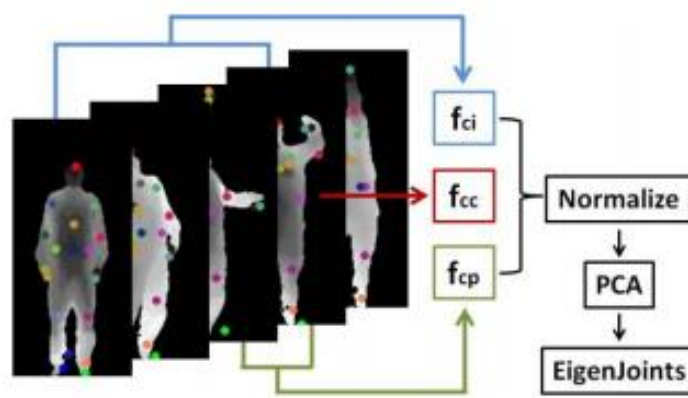
Bu çalışmada normal yöneylemin, gradyan yöneyleminden daha çok bilgi içerdiği önerilmektedir. Yüzey normallerinin bulunmasından sonra hesaplanan normallerin dağılımının nicellendirilmesi ilgili normallerin belirli histogram çubuklarına atanmasıyla yapılmaktadır. Bu işlem sonucunda HON4D adı verilen normal histogramı oluşturulmuştur. Ek olarak sınıflar arasındaki ayrımı arttırmak için ek projeksiyonlar (histogram çubukları) tanımlanmıştır. Son olarak da SVM modeli eğitilerek tanıma işleminde kullanılmıştır. MSR-Action3D veri kümesiyle Cross-Subject Test'inde 88.9% doğru tanıma oranı yakalanmıştır.

Zhao ve diğerleri tarafından [67] önerilen yöntemde RGB ve derinlik haritasından çıkarılan öznitelikler birleştirilerek hareketler tanınmaya çalışılmıştır. Geliştirilen yöntemde eş zamanlı olarak gelen RGB görüntüleri ve derinlik haritalarından çıkarılan öznitelikler birleştirilmiştir. RGB görüntülerinde ilk olarak STIP yöntemi kullanılarak ilgi noktaları tespit edilmiştir. RGB görüntülerinden çıkarılan HOG ve HOF (Histogram of Flow) özniteliklerinin yanı sıra derinlik haritasında tespit edilmiş ilgi noktalarına karşılık gelen bölgelerden Local Depth Pattern (LDP) öznitelikleri çıkartılmıştır. Her ilgi noktası etrafındaki bölge hücrelere bölünmüş ve her hücre için hücre içindeki derinlik değerlerinin ortalaması hesaplanmış ve her hücre ikilisi arasında ortalama değer farkları alınmıştır. Hücreler arasındaki ortalama derinlik değeri farkları öznitelik vektörü haline getirilmiştir. Son olarak RGB görüntüsünden elde edilen HOG, HOF ve derinlik bilgisinden elde edilen LDP değerleri kullanılarak tanıma işlemine geçilmiştir. Her nokta için HOG, HOF ve LDP öznitelikleri çıkarılıp birleştirilmiştir. Oluşturulan bu birleşik öznitelik vektörleri K-means ile kümelenecek bir

sözlük (*codebook*) oluşturulmuştur. Hareketler sözcük dizisi olarak ele alınmış ve SVM modeli ile tanınmaya çalışılmıştır.

Yang ve Tian [68] tarafından yapılan çalışmada eklem iskeletinden elde edilen öznitelikler kullanılarak hareketler tanınmaya çalışılmıştır. Hareketlerin tanınması için öncelikle EigenJoints'ler Şekil 2.8.'de gösterildiği biçimde hesaplanmıştır. Şekilde derinlik haritaları ve bu haritalardan elde edilen öznitelikler belirtilmiştir. Burada f_{cc} her çerçevede yer alan eklem koordinatlarının diğer eklem eklem koordinatlarına olan farklarını belirtmektedir ve Eşitlik 2.12 ile hesaplanmaktadır.

$$f_{cc} = \{x_i - x_j | i, j = 1, 2, \dots, N; i \neq j\} \quad (2.12)$$



Şekil 2.8. Eigen Jointlerin hesaplanması ([68]'den alınmıştır)

f_{cp} ise bir çerçevede yer alan eklemlerin bir önceki çerçevede yer alan eklemlerle olan farkını belirtmektedir.

$$f_{cp} = \{x_i^c - x_j^p | x_i^c \in x_c; x_j^p \in x_p\} \quad (2.13)$$

f_{ci} ise bir çerçevede yer alan eklemlerin başlangıç çerçevesinde yer alan eklemlerle olan farkını belirtmektedir.

$$f_{ci} = \{x_i^c - x_j^i | x_i^c \in x_c; x_j^i \in x_i\} \quad (2.14)$$

Elde edilen öznitelikler üzerinde normalizasyon uygulanarak her üçü kanalda yer alan değerler $[-1,+1]$ aralığına çekilmiştir. Görüldüğü üzere öznitelik kümesi çok boyutludur. Her karşılaştırma 3 eleman içermekte, 20 eklem olduğu düşünülürse her çerçeve için 2970 karşılaştırma yapılması gerekmektedir. 3 kanalın birleştirilmesi ve normalize edilmesinden oluşturulan öznitelikler üzerinde PCA ile boyut indirilmesi yapılmıştır. PCA ile elde edilen eigen value'lar her çerçeve için tanımlayıcı olarak kullanılmıştır. Bu çalışmada sınıflayıcı olarak Naive-Bayes-Nearest-Neighbour (NBNN) [69] kullanılmıştır. NBNN parametrik olmayan bir sınıflayıcı olup pek çok sınıflayıcıya göre üstünlükleri bulunmaktadır. Bu üstünlükler : 1) çok sayıda sınıf olduğu durumlara için uygun olması; 2) aşırı uyuma (*overfitting*) engel olması ; 3) eğitim aşaması içermemesi. NBNN ile hareket tanıma işlemi Eşitlik 2.15'te belirtildiği biçimde gerçekleştirilmiştir.

$$C^* \underset{c}{\operatorname{argmin}} \sum_{i=1}^M ||d_i - NN_c(d_i)||^2 \quad (2.15)$$

Eşitlik 2.15 te d_i çerçeve i için Eigen Joint tanımlayıcılarını belirtmektedir. M çerçeve sayısı, $NN_c(d_i)$ ise c sınıfında yer alan en yakın komşuları belirtmektedir. Ayrıca Yang ve Tian [68] tanımlayıcı çerçevelerin seçimi içinde bir yöntem önermektedirler. Accumulated Motion Energy (AME) denilen bu yöntemde derinlik haritasının 3 eksen içinde projeksiyonu alınıp (f_v), toplam enerjisi hesaplanmaktadır. Her çerçeve için AME değeri Eşitlik 2.16 ile hesaplanmaktadır. AME değeri belirli bir eşik değerinden yüksek olan çerçeveler alınmaktadır.

$$AME(i) = \sum_{v=1}^3 \sum_{j=1}^i |f_v^j - f_v^{j-1}| \quad (2.16)$$

Bu yöntem MSR-Action 3D veri kümesi üzerinde Test-1 de %95, Test-2 de %97 Cross- Subject Test te ise % 83.3 başarı sağlamıştır.

Wang ve diğerleri [70], tarafından geliştirilen yöntemde yine eklem iskeletine ait öznitelikler ve derinlik haritasından çıkarılan öznitelikler kullanılmıştır. Eklemlerden çıkarılan öznitelikler bir eklem için aynı çerçevede yer alan diğer eklemelere olan göreceli konumudur:

$$p_{ij} = p_i - p_j \quad (2.17)$$

Bir i eklemi için öznitelik Eşitlik 2.18 ile belirtilmektedir.

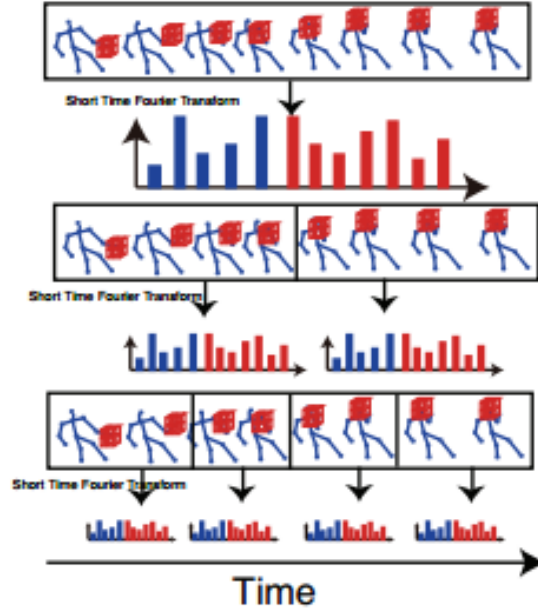
$$p_i = \{p_{ij} | i \neq j\} \quad (2.18)$$

Derinlik haritası üzerinden ise Local Occupancy Patterns (LOP) öznitelikleri çıkarılmıştır. LOP öznitelikleri derinlik haritası üzerinden bir eklem etrafındaki noktalardan elde edilmektedir. T anında elde edilen derinlik haritası üzerinde her eklem yer aldığı lokal bölge $N_x \times N_y \times N_z$ şeklinde gridlere ayrılmıştır. Her grid bölgesi (bin_{xyz}) içine düşen noktalar sayılmıştır ve bir sigmoid normalizasyon fonksiyonundan (δ) geçirilmiştir. Bu işlem Eşitlik 2.19 ile ifade edilmektedir.

$$o_{xyz} = \delta(\sum_{q \in bin_{xyz}} I_q) \quad (2.19)$$

I_q belirtilen konumda bir nokta var ise 1 yoksa 0 değerini almaktadır.

Eklem bilgileri ve derinlik haritasından belirtilen öz nitelikler çıkarıldıktan sonra bu öznitelikler üzerinde Fourier Temporal Pyramid (FTP) şeklinde gösterilmiştir. Bu işlemin amacı her çerçeve için çıkarılan özniteliklerden sabit boyutta bir öznitelik kümesi elde etmektir. İlk olarak p_i eklem öznitelikleri ve o_i derinlik öznitelikleri birleştirilecek g_i elde edilmiştir (i eklem numarasını belirtmektedir). g_j bir zaman fonksiyonudur ve $g_j[t]$ şeklinde tanımlanabilir. $g_j[t]$ üzerinde Fourier Dönüşümü uygulanarak Fourier katsayıları hesaplanmış ve düşük frekans katsayıları öznitelik olarak kullanılmıştır. Bu işlem tüm eklem için ayrı yapılmıştır. FTP çıkarımı Şekil 2.9.'da belirtilmektedir.



Şekil 2.9. FTP Çıkarım Süreci ([70]'den alınmıştır)

Hareketlerin tanınması amacıyla ise Actionlet Ensemble yöntemini ortaya atmışlardır. Bu yöntemde her actionlet eklemlerin bir alt kümesidir. Bu yöntemde ilk olarak her eklem için bir temel sınıflayıcı eğitilmiştir. Sonra her sınıf için tanımlayıcı bir eklem kümesi bulunmaya çalışılmıştır. Bu işlem her sınıf için sınıflamada belirli bir güven (confidence) değeri üstünde sonuç veren eklemlerin eklenmesiyle yapılmıştır. Bu çalışmada MSR-Action 3D veri kümesi ile Cross-Subject Test'te 0.882 oranında başarı elde edilmiştir.

2.4. Hareket Bölütleme

Hareket tanıma işlemlerinde ister erken tanıma amaçlı isterse tüm dizi görüldükten sonra tanımaya dayalı bir yöntem olsun tanıma işlemine geçilmeden önce hareketlerin bölütlenmesi gerekmektedir. Pek çok yöntemde [100,101] hareket sınıflandırılıp tanınmadan önce bölütlenmiştir (segmentation). Hareket tanıma amacıyla çalışmaların büyük kısmı önceden bölütlenmiş ve sadece bir hareketi içeren veri kümeleri kullanılarak geliştirilmiştir. Hesaplama açısından tanıma öncesi bölütleme daha faydalı görünmekle birlikte beraberinde getirdiği zorluklar bulunmaktadır. Hareket bölütleme ve etiketleme için genel bir yöntem ve bir kelime hazinesi bulmak oldukça zor bir sorundur. Genel hatlarıyla hareket bölütleme de en

az hareket tanıma kadar zor bir problemdir. Hareket bölütleme konusunda üç temel yaklaşım bulunmaktadır. Bunlar sınır tespiti, kayan pencere ve dilbilgisi birleştirmedir. Bu yöntemler aşağıda açıklanmıştır.

2.4.1 Sınır Tespiti

Sınır tespiti hareket tespitinde sıkça kullanılan ortak bir yaklaşımdır. Bu yaklaşım hareketler arasındaki sınırların bulunması prensibine dayanmaktadır. Sınır bulma işlemi genel olarak hız ve ivmelenmede görülen değişiklikleri takip edilmesiyle gerçekleştirilir.

Marr ve Vaina [91], insan hareketlerini bölütlemeye dinlenme durumlarını önermişlerdir. Eklem yer değiştirmelerindeki yerel minimumları kullanarak farklı hareketleri bölütlemeye çalışmışlardır. Benzer şekilde Rubin ve Richards [92] iki tip hareket sınırı belirlemişlerdir bunlar start-stop sınırları ve dinamik sınırlardır. Start-stop Marr'ın belirttiği dinlenme durumlarına benzerken, dinamik sınırlar hareket sırasında oluşan tepki ve duraksamalardır. Rui ve Anandan [93] optik akış görüntüleri üzerinde SVD uygulamış ve SVD bileşenlerine ait izlerde oluşan tutarsızlıkları tespit ederek videoyu bölütlemiştir. Ogale [94] optik akış üzerinde yerel maksimum ve minimumları tespit ederek bölütleme yapmaya çalışmıştır.

2.4.2 Kayan Pencere

Kayan pencere yaklaşımı, hareket dizisi üzerinde tanıma işlemlerinin tek veya farklı boyutlarda birden çok kayan pencere gezdirilerek yapıldığı yaklaşımlardır. Bu tür yaklaşımın kullanıldığı çok sayıda çalışma bulunmaktadır [95,96,97]. Bu yaklaşımın ne büyük sorunu hesaplama açısından maliyetli olmasıdır.

2.4.3 Dilbilgisi Birleştirme

Bu yaklaşım genel olarak dilbilgisindeki bir sözcükten sonra gelebilecek sözcüklerin olasılıklarının bilinmesi prensibine dayanır. SMM kullanılan yaklaşımlar genel olarak bu yöntemi uygulamaktadır [79,98,99]. Bu yaklaşımdaki en büyük sıkıntı sınıflayıcıları eğitecek hareketler arasındaki geçişi içeren büyük örnek veri kümelerinin bulunmamasıdır. Sesli ifade tanımda bu veri kümeleri dökümanlardan elde edilirken hareket tanımda böyle bir imkan bulunmamaktadır.

3. Derinlik Bilgisi Kullanılarak İnsan Hareketlerinin Tanınması

Bu kesimde RGBD algılayıcılardan elde edilen derinlik bilgisi ile oluşturulan iskelet modeli üzerinden öznitelikler çıkarılarak hareketlerin tanınmasına yönelik gerçekleştirilen çalışmalar açıklanmıştır. Çıkarılan özniteliklerden sınıflayıcı modeller oluşturulmuş, daha sonra ise bizim tarafımızdan oluşturulan sınırlı bir veri kümesi ve bu tür çalışmalarda yaygın olarak kullanılan veri kümeleri üzerinde tasarlanan modellerin başarımları test edilmiştir.

3.1. İskelet Çıkarma

3 boyutlu derinlik algılayıcıların, insan hareketlerinin tanınması, ortam haritasının çıkarılması gibi alanlarda kullanımı günümüzde gittikçe yaygınlaşmaktadır. RGBD algılayıcılar ile çalışılırken gerçekleştirilmesi gereken adımlardan ilki, derinlik haritasındaki insan silüetlerinin çıkarılmasıdır. Literatürde derinlik üzerinden silüet tespiti için çeşitlik yaklaşımlar bulunmaktadır. Xia ve diğerleri [2] tarafından yapılan güncel bir çalışmada Microsoft Kinect RGBD algılayıcıdan [71] elde edilen derinlik haritasındaki insan silüeti bulunmaya çalışılmıştır. Tez çalışması kapsamında ise silüet tespiti yerine Shotton ve diğerleri [7] tarafından geliştirilmiş olan ve RGBD algılayıcı yardımıyla, gerçek zamanlı olarak vücut eklemlerinin 3 boyutlu konumlarını bulan bir yöntemden faydalanılmıştır. Kullanılan bu yöntemde vücut eklemlerinin konumları yüksek bir doğrulukla tespit edilmektedir. Bu yöntem temelde vücut parçalarını tanımaya dayanmaktadır. Lokal olarak vücut parçaları tanıdıktan sonra tespit edilemeyen eklemler ise tahmin edilmektedir. İnsan vücudu 31 parça olarak tanımlanmıştır. Bu parçalar baş, boyun, gövde, bel, sağ-sol ayak, el, bilek, dirsek, omuz, diz ve baldırlardır. Tanımlanan parçalar Şekil 3.1.'de belirtilmiştir. Her grupta solda derinlik haritası sağda ise elde edilen sentetik vücut parçaları yer almaktadır. Derinlik haritası üzerindeki her nokta için karşılaştırmada kullanılmak üzere öznitelik tanımlanmıştır. Tanımlan öznitelik Eşitlik 3.1'de belirtilmiştir. Bu eşitlikte x ilgili pikseli d_l ilgili derinlik haritasını u, v ise ofset değerlerini belirtmektedir. Önceden tanımlanmış derinlik haritaları üzerinde etiketlenmiş vücut kesimlerine ait öznitelikler kullanılarak Random Forest algoritması [76] eğitilmiştir. Çalışmada çok sayıda örnek kullanılması sınıflayıcıyı güçlü kılmıştır. Algoritmanın eğitiminde kullanılan karar ağaçlarının herbiri için yaklaşık 300.000 görüntü kullanılmıştır.

$$f_{\theta}(I, x) = d_I\left(x + \frac{u}{d_I(x)}\right) - d_I\left(x + \frac{v}{d_I(x)}\right) \quad (3.1)$$

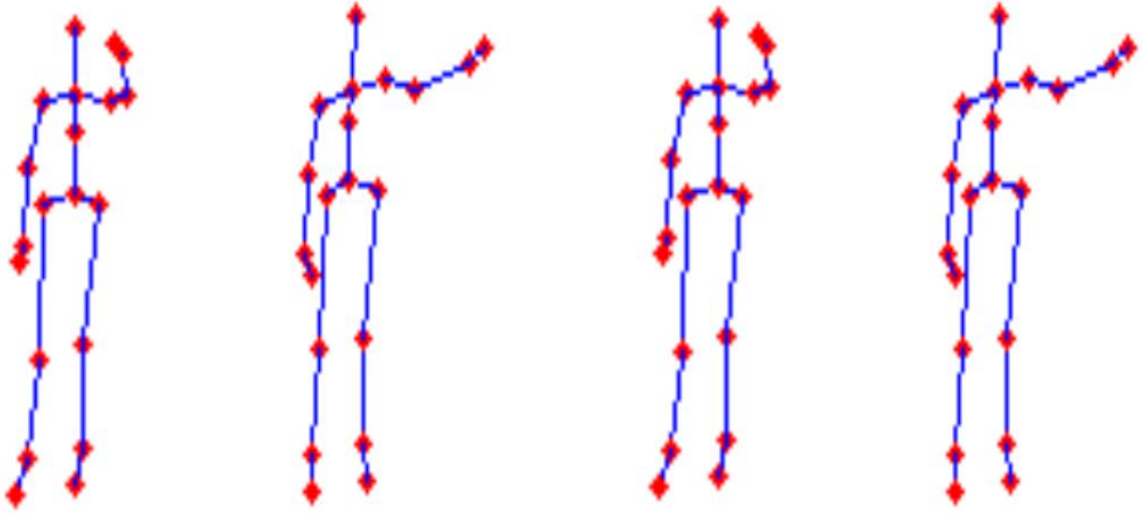
İskelet modeli oluşturulurken ilk olarak vücut kesimlerinin tanınması yapılmıştır. RGBD algılayıcıdan gelen derinlik haritası yardımıyla görüntüde yer alan her piksele ilişkin derinlik ve pozisyon (2 boyuttaki derinliğe bağlı konum) bilgileri kullanılarak öznitelikler elde edilmiş ve RF sınıflayıcısı ile ilgili sınıfa yani vücut parçasına ait olduğu tespit edilmiştir. Daha sonra eklemlerin bulunması için 2 boyutlu koordinatlar ve derinlik haritası kullanılarak 3 boyutlu koordinatlara dönüşüm yapılmış ve mean-shift [9] algoritması kullanılarak konumlar bulunmuştur.



Şekil 3.1. Shotton ve diğerleri tarafından geliştirilen yöntemle ait tanımlı vücut parçaları ([7]'den alınmıştır)

Bu yöntem XBOX 360 oyun konsolu tarafında kullanılmaktadır. Microsoft Kinect sistemi [71], derinlik haritası üzerinden GPU'su yardımıyla 20 eklemli iskelet modeli oluşturma özelliğine sahip bir RGBD algılayıcıdır. Kinect sisteminin silüete ilişkin iskelet çıkarma ve takibini kızıl ötesi ışın yayan projektör yardımıyla gerçekleştirmektedir. Kızıl ötesi ışınlar insan gözünün fark edebileceği ışın bandında yer almadığı için cihazın çalışması sırasında fark edilmemektedir. Cihaz tarafından yayılan ışınlar CMOS algılayıcı tarafından toplanmakta ve derinlik haritası oluşturulmaktadır. Kızılötesi ışınlar kullanılarak derinlik haritası elde etmenin en büyük avantajı hesaplama maliyeti açısından düşük olması ve gerçek zamanlı uygulamalar geliştirilmesine olanak sağlamasıdır. Derinlik haritası oluşturmak için çoklu kamera sistemleri kullanılarak da elde edilebilir fakat bu işlemin hesaplama maliyeti oldukça fazladır.

Bu yöntemden elde edilen 20 eklemli insan iskeleti Şekil 3.2.'de görülmektedir.



Şekil 3.2. Elde edilen insan iskeleti

3.2. Kullanılan Veri Kümeleri

İlk aşamda yapılan çalışma kapsamında 4 farklı aktör tarafından 10 farklı harekete ilişkin toplamda 120 hareket verisi ile tez kapsamındaki veri kümesi tanımlanmıştır (HUN-3D). Her aktöre aynı hareket 3 defa yaptırılmıştır. Tanımlanan hareket kümesi el sallama, saate bakma, tekme atma, yumruk atma, kolları çapraz yapma, yürüme, kafa kaşıma, yerden bir şey alma, çömelme ve kendi ekseninde dönme hareketlerini

içermektedir. Hareketler belirlenirken XMAS [45] veri kümesindeki hareketler baz alınmıştır. XMAS veri kümesinde benzer hareketler çoklu kamera düzeneği karşısında yapıлып kaydedilmiştir. XMAS veri kümesi baz alınarak yapılan hareketlerden bazılarına ilişkin örnekler Şekil 3.3.'te gösterilmektedir. Hareket kümesinde aşağıdaki hareketler bulunmaktadır.

- | | |
|-------------------------|--------------------------|
| 1. Saate bakma | 6. Oturma |
| 2. Kolları çapraz yapma | 7. Kendi ekseninde dönme |
| 3. Tekme atma | 8. Kafa kaşıma |
| 4. Yerden birşey alma | 9. Yürüme |
| 5. Yumruk atma | 10.El sallama |

Ayrıca veri kümesi olarak, 2 farklı veri kümesinden daha faydalanılmıştır. Bunlar Microsoft Research Cambridge-12 (MSRC-12) ve MSR-Action3D veri kümeleridir [72,73]. Bu iki veri kümesi de, Microsoft SDK'sından elde edilen eklem iskeletine ilişkin hareket dizileri içermektedir.

MSRC-12 veri kümesi, 30 farklı insandan elde edilen 12 farklı harekete ilişkin 594 adet hareket dizisi içermektedir. Bazı insanların hareketleri tekrarlı sayıda bulunmaktadır. Hareketler kinect poz tanıma sürecine uygun 20 eklemli iskelet modeli şeklinde tanımlanmıştır ve 30 Hz lik bir örnekleme oranıyla elde edilmiştir. Hareket kümesinde aşağıdaki hareketler bulunmaktadır.

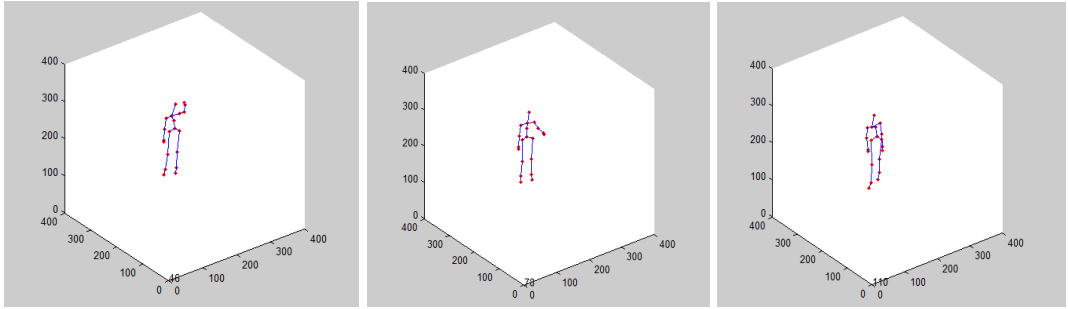
- | | |
|---------------------|---------------------------|
| 1. Eğil, saklan | 7. Müziğin sesini aç |
| 2. Silah ateşle | 8. Menüler arasında dolaş |
| 3. Bir nesne fırlat | 9. Müziği aç |
| 4. Silah değiştir | 10.Alkışlamak |
| 5. Tekme | 11.Müziğe İtiraz |
| 6. Gözlük giy | 12.Tempoyu yükselt |

MSR-Action3D veri kümesi de [73] yine eklem iskeletine ilişkin hareket dizileri içermektedir. Veri kümesi 10 farklı insandan elde edilen 20 farklı harekete ilişkin 567

hareket dizisi içermektedir. Bu veri kümesindeki hareketler daha ziyade günlük aktivitelere ilişkindir. Bu veri kümesinde yer alan hareketler:

- | | |
|----------------------|----------------------|
| 1- Yüksek el sallama | 11- İkili el sallama |
| 2- Yatay el sallama | 12- Eğilme |
| 3- Çekiç hareketi | 13- Boks |
| 4- El ile yakalama | 14- Tekme |
| 5- İleri yumruk | 15- Yan tekme |
| 6- Fırlatma | 16- Jogging |
| 7- X çizme | 17- Tenis servis |
| 8- Tik atma | 18- Tenis atışı |
| 9- Çember çizme | 19- Tutup fırlatma |
| 10- Alkış | 20- Golf atışı |

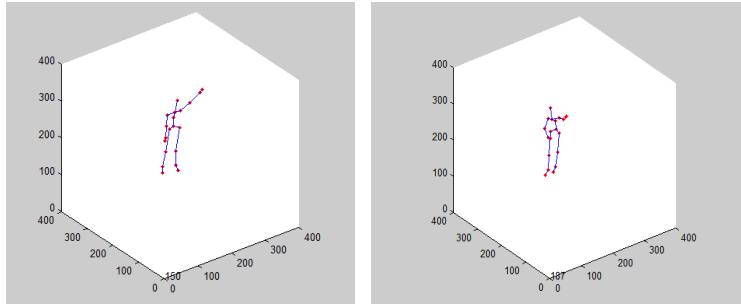
Bu veri kümeleri üzerinde öznitelik çıkarma yapıp, Support Vector Machine ve Random Forest yöntemleri kullanılarak hareket tanınma işlemi yapılmaya çalışılmıştır.



(a) Kafa kaşıma

(b) Saate bakma

(c) Kendi ekseninde dönme



(d) El sallama

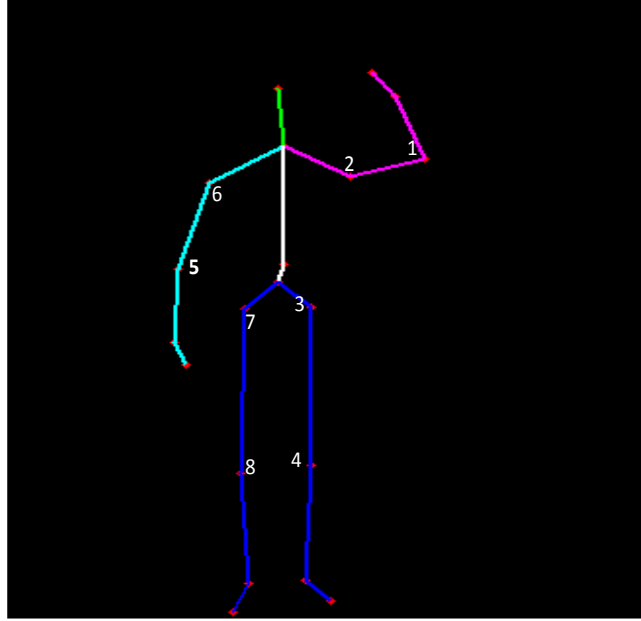
(f) Yumruk atma

Şekil 3.3. HUN-3D Veri kümesinde yer alan hareketlere ilişkin örnekler

3.3. Öznitelik çıkarma

Veri kümesi oluşturulduktan sonra öznitelik çıkarma aşamasına geçilmiştir. Öznitelik olarak ilk etapta eklemlere ilişkin açılar kullanılmıştır. Veri kümesindeki hareketleri tanımak için en uygun açılar olan omuz-dirsek-kol ve kalça-diz-ayak bileği arasındaki açı değerleri, hareket boyunca kaydedilip bu açı değerlerine ilişkin histogramlar oluşturulmuştur. Açı değerleri hesaplanırken, elde edilen iskelet modelinde dirsek, omuz, diz, kalça, el ve ayak bileklerine ait 3 boyutlu koordinat değerleri kullanılmıştır. Bu koordinat değerleri kullanılarak nokta ikilileri doğru parçaları olarak varsayılmış ve bu doğru parçaları arasındaki açı değerleri hesaplanmıştır.

Doğru parçası bir uzayda bir P_1 noktasından P_2 noktasına uzanan bir bölüngerdir (segment). Eğer yönlü bir doğru parçası ise P_1P_2 vektörüne karşılık gelir. Pozitif yön P_1 'den P_2 'ye doğrudur. Hesaplanacak olan eklem açıları Şekil 3.4.'te rakamlarla gösterilmektedir. Açılar hesaplanmadan önce açısı bulunacak eklemlere ilişkin nokta ikilileri, doğru parçaları olarak alınmıştır.



Şekil 3.4. Hesaplanan açı değerleri

İki doğru parçası arasındaki açı hesaplanmadan önce doğru parçalarının yönlü kosinüs (directed cosine) değerlerinin hesaplanması gerekmektedir. $P_1(x_1, y_1, z_1)$ ve

$P_2(x_2, y_2, z_2)$ P_1P_2 doğru parçasını tanımlayan iki nokta olsun. P_1P_2 doğru parçasına ait kosinüs değerleri aşağıdaki gibi verilmektedir.

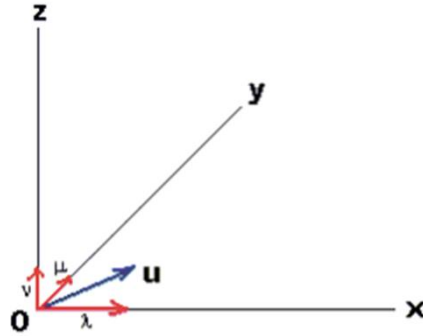
$$\cos \alpha = \frac{x_2 - x_1}{d} \quad \cos \beta = \frac{y_2 - y_1}{d} \quad \cos \gamma = \frac{z_2 - z_1}{d} \quad (3.2)$$

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (3.3)$$

α , β , γ sırasıyla x, y ve z eksenindeki eğim değerlerini belirtmektedir. Eşitlik 3.3'te belirtilen d değeri ise iki nokta arasındaki uzaklığı belirtmektedir. Bu değerlerin ikisi verildiği zaman üçüncüsü Eşitlik 3.4 ile bulunabilir.

$$\cos^2 \alpha + \cos^2 \beta + \cos^2 \gamma = 1 \quad (3.4)$$

Bir doğru parçasının yönlü kosinüs bilgilerini λ , μ ve ν olarak gösterebiliriz. Aynı şekilde bu değerler doğru parçasının x, y, z eksenleri üzerindeki birim vektör izdüşümlerine de karşılık gelmektedir. İlgili birim vektörler Şekil 3.5.'te belirtilmiştir. Bu vektörlerin birleşimi doğru parçasını vermektedir.



Şekil 3.5. Birim vektörler [74]

Uzaydaki iki doğru parçası arasındaki açı değeri, bu iki doğru parçası keşişse de keşişmese de bu iki doğru parçasının birim vektörlerinin aralarındaki açılara tekabül etmektedir.

$(\lambda_1, \mu_1, \nu_1)$ ve $(\lambda_2, \mu_2, \nu_2)$ olarak tanımlanan iki doğru parçası arasındaki Θ açısı Eşitlik 3.5 ile belirtilmektedir.

$$\cos \theta = \lambda_1 \lambda_2 + \mu_1 \mu_2 + v_1 v_2 \quad (3.5)$$

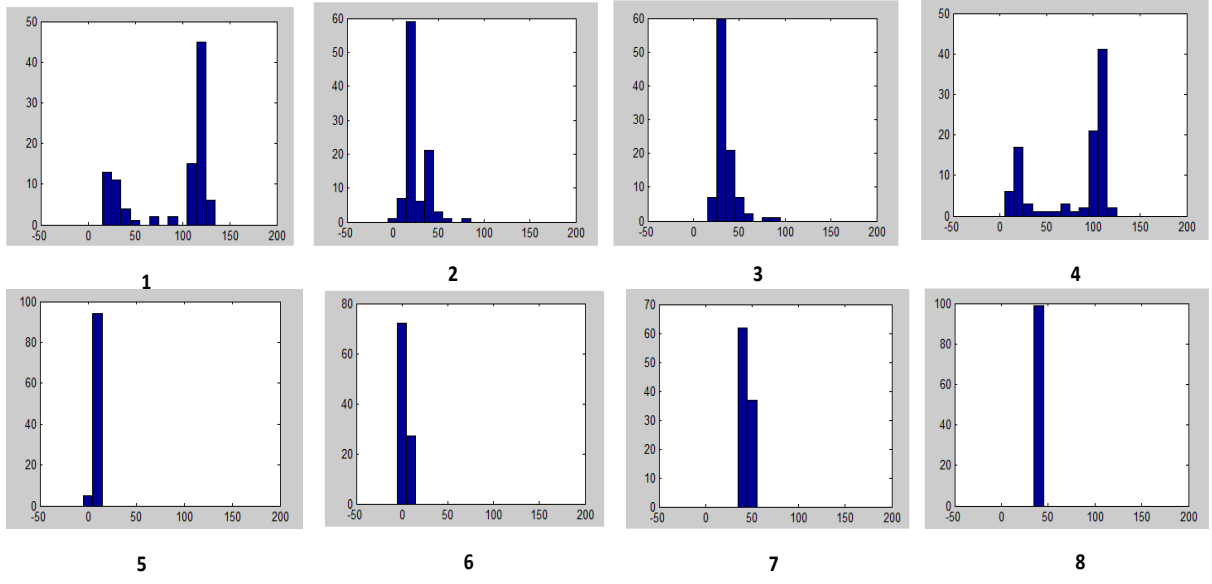
Böylelikle $\alpha_1, \beta_1, \gamma_1$ ve $\alpha_2, \beta_2, \gamma_2$ yön açılarına sahip iki doğru arasındaki açı aşağıdaki gibi de verilebilir.

$$\cos \theta = \cos a_1 \cos a_2 + \cos \beta_1 \cos \beta_2 + \cos \gamma_1 \cos \gamma_2 \quad (3.6)$$

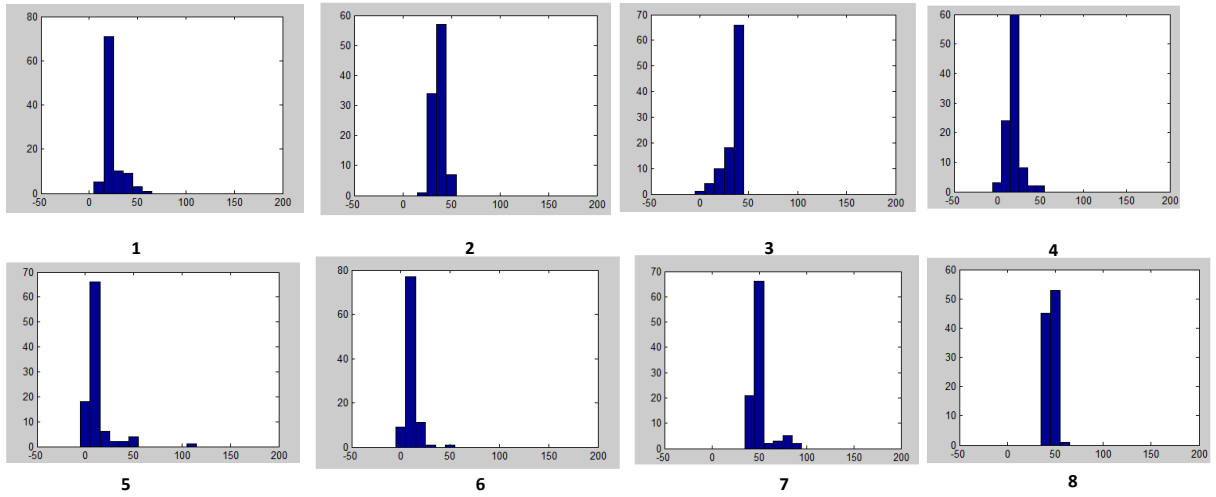
Şekil 3.4.'te belirtilen açıları hesaplamak için ilgili açıyı tanımlayan eklem noktalarına ilişkin koordinat değerleri kullanılmıştır. Her açı, $P_1P_2 P_2P_3$ şeklinde iki bölüme olarak tanımlanmıştır. Yani hesaplanacak her açı ortak noktaya sahip iki doğru parçası olarak ele alınmış ve hareket boyunca her çerçeve için bu açı değeri hesaplanmıştır. Açı değerlerinin hesaplanmasında Eşitlik 3.6 kullanılmıştır. Her çerçeve için açı değerleri hesaplandıktan sonra bu eklemlerdeki açıların histogramları oluşturulmuştur. Gözlenen 8 eklem hareket boyunca hangi açılara sahip olduğu bilgisi hareketin tanınmasında öznitelik olarak kullanılmıştır. Çalışma sürecinde, 10, 15, 20, 25, 30 gibi farklı sayıdaki kutulara sahip histogramlar hesaplanmıştır. En iyi sınıflama sonuçlarına 10 kutu ile ulaşıldığı için bu raporda sadece 10 kutuya ilişkin sonuçlar verilmiştir. Farklı açı aralıkları kullanılarak oluşturulan histogramlara ilişkin sınıflama doğrulukları Çizelge 3.6.'da yer almaktadır.

Her eklem için ayrı ayrı çıkarılan tüm histogramlar birleştirilmiştir ve bütünleşik bir gösterim elde edilmiştir. Bu işlem tüm eklemlere ilişkin histogramlarda ki bilgilerin tek boyutta bir dizi şeklinde peş peşe eklenmesiyle yapılmıştır. Ekleme işlemi yapılırken sıra olarak yukarıda Şekil 3.4.'te verilen açı sıralaması baz alınmıştır. Sıra bilgisi burada önem taşımaktadır çünkü hangi ekleme değişim olduğu bilgisi sınıflama algoritmasının seçiciliği açısından önemlidir.

Eklemlere ilişkin iki farklı hareketin açı histogramları aşağıda Şekil 3.6. ve 3.7.'de verilmiştir. Şekil 3.6.'da verilen ilk histogram grubu kolları çapraz yapma hareketine aittir. Şekil 3.7.'deki ikinci histogram grubu ise tekme hareketi sırasında ölçülen eklem açılarının histogramıdır.



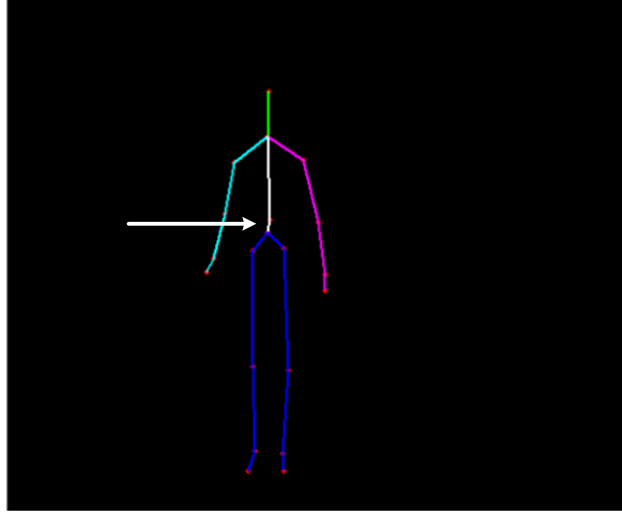
Şekil 3.6. Kolları çapraz tutma için histogramlar



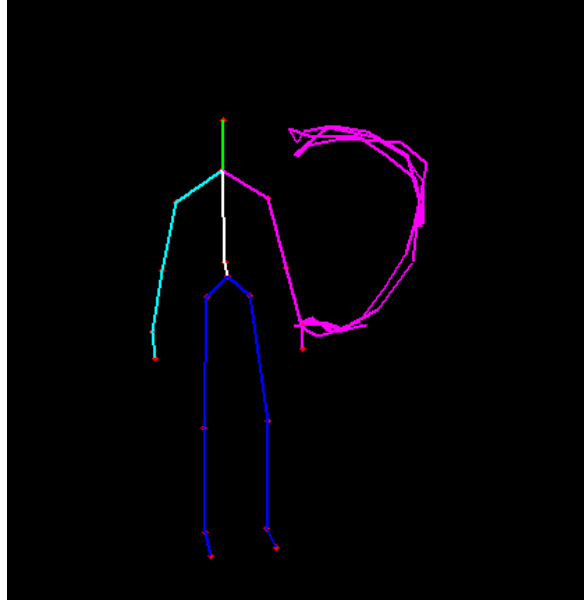
Şekil 3.7. Tekme için histogramlar

Açı değerleri her zaman tek başına yeterli olmayabilir. Özellikle sıçrama, oturma gibi hareketlerin ayırt edilmesinde açı değerlerinin yanı sıra eklem noktalarının yer değiştirme bilgisi de gerekmektedir. Sadece eklem açı bilgileri kullanılarak elde edilen başarımlar Çizelge 3.1.'de verilmiştir. Sadece eklem açıları kullanılarak %80 civarında bir sınıflama başarısı elde edilmiştir. Eklemlerin yer değiştirme bilgisi de eklenerek tanıma başarımlarını %90'lar düzeyine çıkarılmıştır. Öncelikle tüm noktaların

kalça noktasına görel koordinatları bulunmuştur. Kalça noktası Şekil 3.8.'de gösterilmektedir.



Şekil 3.8. İskelete ait kalça noktası



Şekil 3.9. El sallama sırasında el bileğinin yer deęiřtirmesi.

El ve ayak eklemleri dıřındaki tüm noktalar için görel koordinat deęerleri hesaplanmıřtır. Shotton ve diđerlerinin [7] algoritmasından elde edilen iskelette, el ve ayaklara karřılık gelen noktalar gürültü sebebiyle, yanlış konumlarda algılanabilirken,

el ve ayak bileklerinin konumu daha tutarlı bir şekilde algılanabilmektedir. Bu sorundan etkilenmemek ve daha tutarlı sonuçlar elde edebilmek amacıyla, el ve ayak bileklerinin yer deęiřtirme bilgisi kullanılmıřtır.

Görelü koordinatlar, görelü koordinatı hesaplanacak noktanın koordinat deęerlerinin kalça noktasının koordinatlarından çıkarılmasıyla hesaplanmaktadır. Daha sonra el ve ayak dıřında kalan tüm eklemlerin, kalça eklemine göre x, y ve z eksenlerindeki yer deęiřtirmeleri, ardıřık çerçeveller arasındaki konum farkları kullanılarak kümülatif olarak hesaplanmıřtır. Her üç eksendeki yer deęiřtirmelerin ayrı hesaplanmasının amacı, eksene göre farklılık gösteren hareketlerin ayırımının yapılabilmesidir. Örneęin el sallama ve yumruk atma düşünülürse bu iki hareketin ayırımında farklı eksenlerde hareket edilmektedir. Aynı zamanda tüm noktaların toplam yer deęiřtirmesi de hesaplanmıřtır. El sallama için el bileęinin aldıęı toplam yol Őekil 3.9.'da gösterilmektedir.

3.4. Hareketlerin Sınıflanması

Eklem açılarına ait histogramlar ve eklem yer deęiřtirme bilgileri elde edildikten sonra sınıflama yapılmıřtır. Sınıflama için Support Vector Machine (SVM), Random Forest (RF) yöntemleri kullanılmıřtır.

3.4.1. Support Vector Machine

SVM, makine öğrenmesi yöntemleri arasında sık kullanılmaya bařlayan yöntemlerin bařında gelmektedir [75]. Bu yöntemde sınıflama iřlemi lineer ve lineer olmayan ayırıcı fonksiyonlar yardımıyla yapılmaktadır. SVM yöntemi, eğitim kümesinde bulunan noktaları ayıran ve bu noktalara en uzak mesafede bulunan düzlemi bulmaya çalıřır. Günümüzde SVM yönteminin kullanımı giderek yaygınlařmaktadır [75]. Yapay nöron aęlarının kullanımında dramatik bir düşüş yařanırken, SVM kullanılan çalıřmaların sayısında ciddi bir artış görölmektedir. Kısa bir süre sonra en yaygın kullanılan yöntemlerden biri olabilir [75].

Bu yöntem veriyi en doęru şekilde ayıran fonksiyonu bulma ilkesine dayanır. Lineer olarak ayrılabilen bir veri kümesinde iki kümeyi $y=\{-1,1\}$ olarak gösterelim. Aynı sınıftaki gözlemleri aynı tarafta bırakan H_0 düzlemini bulmak istiyoruz. İki kümeyi ayıran birden fazla düzlem olabilir. Bunlardan kümelere en uzak ikisini H_1 ve H_2

olarak tanımlayalım. H_0 düzlemi bu iki düzlemin ortasında kalacak olan düzlemdir. Şekil 3.10'da H_0 , H_1 ve H_2 gösterilmektedir. H_0, H_1 ve H_2 aşağıdaki şekilde tanımlanır.

$$H_0 : W^T X + b = 0 \quad (3.7)$$

$$H_1 : W^T X + b = 1 \quad (3.8)$$

$$H_2 : W^T X + b = -1 \quad (3.9)$$

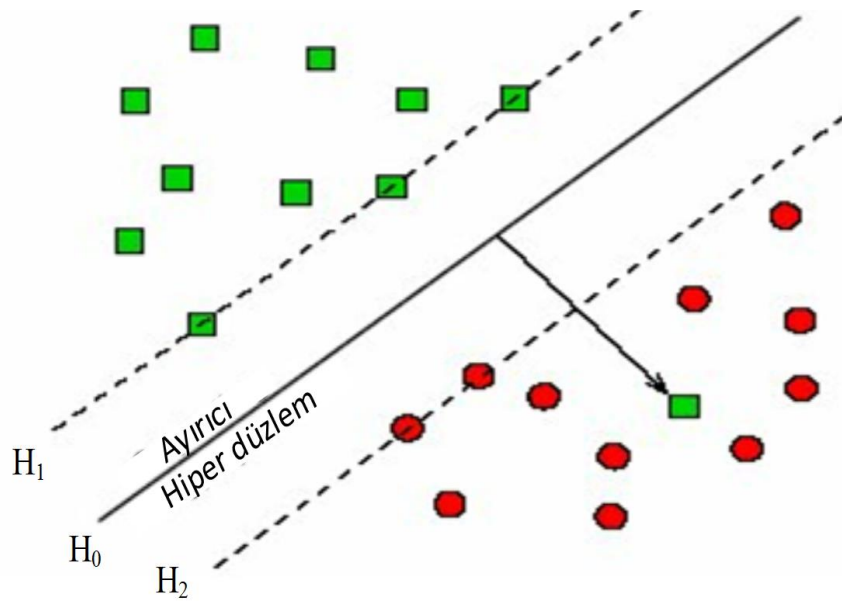
Bu eşitliklerde W ağırlık vektörünü $W = \{w_1, w_2, \dots, w_n\}$, n öznelik sayısını ve b sabit bir değeri belirtmektedir. X eğitim kümesindeki gözlemleri belirtmektedir. H_0 düzleminin altında kalan noktalar Eşitlik 3.9, üstünde kalan noktalar Eşitlik 3.11 ile tanımlanır.

$$W^T X + b < 0 \quad (3.10)$$

$$W^T X + b > 0 \quad (3.11)$$

Eşitlik 29 ve 30 'un birleşiminden Eşitlik 3.12 elde edilir. H_1 ve H_2 üzerinde yer alan noktalar destek vektörü olarak isimlendirilir.

$$Y_i(W^T X + b) - 1 \geq 0 \quad (3.12)$$



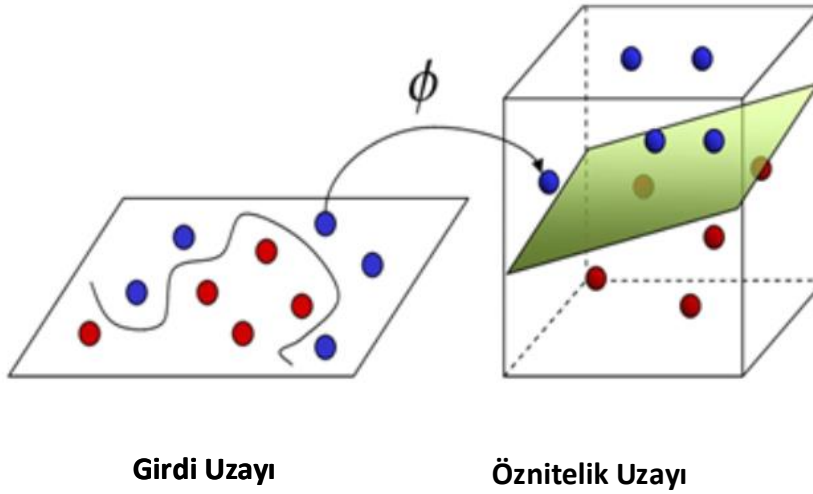
Şekil 3.10. Düzlemler ve destek vektörler

Destek vektöründe olan noktaların H_0 düzlemine olan uzaklığı Eşitlik 3.13. ile bulunabilir.

$$d = \frac{|Wx_p \pm b|}{\|w\|} \quad (3.13)$$

$d=1$ olarak alınırsa iki düzlem arasındaki m uzaklığı $m=2d=2/\|w\|$ olarak tanımlanabilir. Bu eşitlikten $\|w\|$ değerini en küçük yapan minimizasyon problemi çözülerek H_0 düzlemi bulunabilir.

Doğrusal (*linear*) durum, doğrusal olmayan durumlara da kolayca uyarlanabilir. Bu durumda öznelik kümesinin boyutu artırılarak iki sınıfı ayıran düzlem bulunmaya çalışılır ve böylelikle doğrusal olmayan problem, doğrusal sınıflama problemi durumuna getirilir. Boyut artırma işlemi Şekil 3.11.'de gösterilmiştir. Bu örnekte doğrusal olarak ayrılamayan 2 boyutlu örneklem üzerine yeni bir öznelik boyutu eklenerek, örneklem doğrusal olarak ayrılabilir bir hale getirilmiştir.



Şekil 3.11. Öznelik uzayında boyut genişletme

3.4.2. Random Forest

Random Forest [76] birçok karar ağacından oluşan bir yapıdır. Bu yöntemle analiz sırasında, Random Forest'teki her bir ağaç için bootstrap yöntemiyle veri setinden örneklem seçilir ve seçilen verilerin 2/3'ü ağaç oluşturmak için kullanılır ve bir sınıflama yapılır. Bu sınıflamalar "oy" (*vote*) alır. Random Forest algoritması ise "Forest" içindeki tüm ağaçlardan en çok oy alanı seçer ve onun sınıflamasını kullanır.

Düşük hata oranına sahip ağaç daha iyi sınıflayıcıdır. Random Forest algoritmasında hata oranı iki şeye bağlıdır: Ağaçlar arasındaki korelasyona ve her ağacın kendi hata oranına. Random Forest'in en önemli özelliği aşırı uyuma (*over fitting*) el vermemesi, hızlı çalışması ve oluşturulan karar ağacı kümesinin diğer veri kümeleri içinde kullanılmak üzere saklanabilmesidir. Random forest algoritmasının çalışma ilkesi aşağıdaki şekilde özetlenmiştir:

1. Bootstrap yöntemiyle veriden örneklem al. Kalan veri kümesini "out-of-bag" olarak ayır.
2. Her düğümünde m tane seçilmiş değişkeni kullanan, en iyi ayrımın yapıldığı bir ağaç oluştur.
3. Out-of-bag veriyi tahmin etmek için oluşturulan karar ağacını kullan.
4. Out-of-bag veri kümesi üzerindeki tahminleri kullanarak oy dağılımını belirle.
5. Yukarıdaki adımları N defa tekrar ederek N ağacın birleşiminden oluşan bir sınıflandırıcı oluştur. Oy çokluğuna göre test verisi üzerinde tahminler yap.

3.5. Sınıflandırma Sonuçları

Öznitelikler, SVM ve Random Forest (RF) algoritmaları kullanılarak hareketler sınıflandırılmıştır. Sınıflamada öncelikle sadece eklem açıları kullanarak sınıflamanın başarısı değerlendirilmiştir. Daha sonra eklem açılarına ek olarak eklemelerin yer değiştirmeleri de öznitelik olarak alınıp sınıflama işlemi tekrar yapılmıştır. Sadece eklem açıları kullanılarak yapılan sınıflamaya ilişkin sonuçlar Çizelge 3.1.'de; eklem açıları ve yer değiştirme özniteliklerinin birlikte kullanılarak yapılan sınıflama sonuçları Çizelge 3.2.'de yer almaktadır. CA (*Classification Accuracy*) doğru sınıflama oranını, AUC (*Area Under Curve*) ROC eğrisi altında kalan alanı, Sens duyarlılığı (*Sensitivity*) belirtmektedir. Eşitlikler 3.14'te CA değerlerinin hesaplanmasında kullanılan eşitlik verilmiştir.

$$CA = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.14)$$

SVM sınıflandırıcısı için çeşitli parametreler denenmiştir. Çekirdek olarak RBF, doğrusal, sigmoid ve polinom çekirdekleri çalışılmıştır. En başarılı sonuçlar doğrusal çekirdek ile elde edilmiştir. Tolerans değeri olarak 0,0010, cost parametresi ise 32,0 olarak seçilmiştir. Bu parametrelerin bulunmasında Orange'ın bir özelliği olan otomatik parametre tespiti de denenmiştir. Bu yöntem de lineer çekirdeği seçmiştir. Zaten RBF ve doğrusal çekirdekler, SVM sınıflayıcısı ile en çok kullanılan çekirdek türleridir [75].

Random Forest'a ilişkin parametrelerin seçiminde ağaç sayısı esastır. Ağaç sayısı ise 100 olarak alınmıştır. Farklı sayıda ağaçlara sahip modeller denenmiştir. Fakat iyi bir model, genele uygulanacak kadar sade ve özeldir başarılı olacak kadar da geniş olması gerektiğinden, ağaç sayısı 100'de bırakılmıştır. Ağaç sayısının fazla olması kullanılan veri kümesi üzerinde başarıyı arttırmasına karşın bu modelin genele uygulanabilir olması için mümkün olduğunca dar olması gerekmektedir. 50-100 ağaç arasında yakın sonuçlar elde edilmiştir. 100-200 arasında ağaç kullanımında ise sonuçlar arasında bir değişiklik görülmemiştir. 200 den sonra ise az da olsa düşme görülmüştür. 100 sayısı ayrıca zaman karmaşıklığı açısından da uygun olduğu için seçilmiştir. Ağaç sayısının çok fazla artması eğitim ve sınıflama sürelerini ciddi oranda arttırmaktadır.

3.5.1. HUN-3D Üzerinde Sınıflama Sonuçları

İlk deneyde kendi oluşturduğumuz HUN-3D veri kümesi üzerinde testler sadece açığı histogramlarını kullanarak yapılmıştır. Çizelge 3.1.'de yapılan çalışmalara ait sonuçlar belirtilmiştir. 2. Deneyde ise açığı histogramları ve eklem noktalarına ait yer değiştirmeler kullanılmıştır. 2. Deneye ait sonuçlar Çizelge 3.2.'de belirtilmektedir. Sınıflayıcıların test edilmesinde literatürde yer alan Test-1, Test-2 ve Cross-Subject Test yöntemleri kullanılmıştır.

Test-1: Bu yöntemde örneklem kümesi 3 eşit parçaya ayrılmakta parçalardan 1 ile sınıflayıcı eğitilip kalan parçalarla test edilmektedir.

Test-2: Bu yöntem ise Test-1 in tam tersi bir durum mevcuttur, Test-2 de parçaların 2 i ile eğitilen sınıflayıcı kalan parça ile test edilmektedir.

Cross-Subject Test: Bu test yönteminde ise örnekler kendisini gerçekleştiren aktöre göre gruplanıp bir grup aktörün gerçekleştirdiği hareketler ile eğitilen sınıflayıcı diğer bir grup aktör tarafından gerçekleştirilen hareketler ile test edilmektedir. Aktörler eşit sayıda 2 gruba ayrılmış ve 1. grup eğitim 2. grup test grubu olarak kullanılmıştır.

HUN-3D üzerinde farklı özniteliklerin sınıflama üzerindeki etkisini görmek amacıyla Sadece yer değiştirme ve açı bilgilerinin kullanıldığı deneylerde yapılmıştır. Bu deneylere ait sonuçlar Çizelge 3.1. ve Çizelge 3.2.'de görülmektedir. Çizelgelerde sütunlar yapılan testleri, satırlar ise kullanılan sınıflandırma yöntemlerini belirtmektedir.

Çizelge 3.1. HUN-3D üzerinde Eklem açıları ile sınıflama sonuçları

| | Test-1 | Test-2 |
|------------|--------|--------|
| SVM | 63.26 | 74.75 |
| RF | 77.66 | 84.12 |

Çizelge 3.2. HUN-3D üzerinde Yer değiştirme bilgisi ile elde edilen sonuçlar

| | Test-1 | Test-2 |
|------------|--------|--------|
| SVM | 61.26 | 74.33 |
| RF | 66.95 | 76.47 |

Bu sınıflama işlerimin sonucunda, eklem açıları ve eklem noktalarına ait yer değiştirme bilgisi öznitelikleri tek başlarına yeterince iyi sınıflama sonuçları üretmemektedir. İkisinin birlikte kullanımı daha yüksek ve kayda değer bir sınıflama başarısı vermektedir. Bu sebeple bundan sonraki deneylerde bu iki öznitelik kümesi birlikte kullanılmıştır. İki öznitelik kümesinin birlikte kullanıldığı sınıflama sonuçları Çizelge 3.3.'te verilmektedir.

Çizelge 3.3. Eklem açıları ve eklem yer deęiřtirme bilgisinin birlikte kullanıldıęı sınıflama sonuçları

| | Test-1 | Test-2 |
|------------|--------|--------|
| SVM | 82.60 | 86.50 |
| RF | 91.82 | 93.50 |

Sadece açı histogramları kullanılması durumunda bazı hareketler sınıflama sırasında birbirleriyle karıřtırılmaktadır. Örneęin, el sallama ve yumruk hareketleri benzer açı deęerleri üretmektedir. Bu sebepten sadece açı deęerleri ile bu iki hareketin ayrıřtırılmasında zorlanılmaktadır. Aynı durum sadece yer deęiřtirme bilgisinin kullanıldıęı durumlar içinde geçerlidir. Mesala kafa kařıma ve el sallama hareketinde olduęu gibi. Eklem açıları ve eklemlerin yer deęiřtirme bilgisinin birleřtirilmesi sınıflama başarımını ciddi oranda arttırmıřtır. Çizelge 3.1.-3.3.'te verilen sonuçlara göre RF modeli SVM modeline göre daha başarılı sonuçlar üretmektedir. Tüm özniteliklerin kullanılması durumunda RF modeli Test-2'de 93.5% sınıflama başarımına ulaşmaktadır.

Deneylere iliřkin hata matrisleri yanlıř sınıflamaların sebebini anlamak için de gözlemlenir. Hata matrislerinde kolonlar tahmin edilen sınıflara ait etiketlere, satırlar ise gerçek sınıflara iliřkin etiket deęerlerine karřılık gelmektedir. Doęru tahminler hata matrisinin diagonalinde yer almaktadır. SVM modele iliřkin hata matrisi Çizelge 3.4.'te verilmiřtir. Hata matrisinde diagonal üzerinde yer alan deęerler doęru sınıflama oranını belirtmektedir. Satırlar gerçek sınıf etiketlerine, sütunlar ise sınıflandırıcı tarafından atanan sınıf etiketlerine karřılık gelmektedir.

Çizelge 3.4. HUN3D’de Test-2 de SVM modeli için hata matrisi

| | Saate Bak | El Salla | Kolları Çaprazlama | Tekme | Alma | Yumruk | Oturma | Dönme | Kafa Kaşımaya | Yürüme |
|--------------------|-----------|----------|--------------------|-------|------|--------|--------|-------|---------------|--------|
| Saate Bak | 71.8 | 0 | 28.2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| El Salla | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Kolları Çaprazlama | 29.3 | 0 | 70.7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Tekme | 0 | 0 | 0 | 97.5 | 2.5 | 0 | 0 | 0 | 0 | 0 |
| Alma | 0 | 0 | 0 | 2.3 | 86 | 0 | 11.6 | 0 | 0 | 0 |
| Yumruk | 0 | 0 | 0 | 0 | 4.8 | 95.2 | 0 | 0 | 0 | 0 |
| Oturma | 0 | 0 | 0 | 0 | 0 | 0 | 96.8 | 3.2 | 0 | 0 |
| Dönme | 0 | 0 | 0 | 0 | 0 | 0 | 11.9 | 81 | 7.1 | 0 |
| Kafa Kaşımaya | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11.9 | 78.6 | 9.5 |
| Yürüme | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 90 |

SVM modelinde, hatalı tahminler özellikle saate bak ve kolları çapraz yapma hareketleri arasında yapılmıştır. Vücut uzuvlarının birbirleri üstüne bindikleri hareketlerde eklem iskeleti hatalı üretilmektedir. Bu durum ve bu tür hareketler arasındaki benzerlikler sınıflamadaki hata oranını artırmaktadır. Dönme ve kafa kaşımaya hareketlerindeki hatalı tanımalardaki artış gürültüden kaynaklanmaktadır. Bu hareketler için Shotton ve diğerleri [7] tarafından geliştirilen yöntem hatalı iskelet modelleri üretebilmekte ve buda hatalı sınıflamalara neden olmaktadır.

Test-2 de RF modeli için hata matrisi Çizelge 3.5.’te verilmiştir. Çizelge 3.4. ile karşılaştırıldığında RF modeli SVM modeline göre daha başarılı sonuçlar üretmektedir. Kolları çapraz yapma ve saate bakma hareketlerindeki hatalı sınıflama oranı düşmüştür fakat bazı saate bakma ve kolları çapraz yapma hareketleri arasında karışıklık olmaya devam etmektedir. Dönme ve kafa kaşımaya hareketlerindeki hatalı

sınıflama hala gürültüden ötürü devam etmektedir. Diğer hareketler içinse daha yüksek sınıflama oranlarına erişilmiştir.

Çizelge 3.5. RF Modeli için Test-2 de hata matrisi

| | Saate Bak | El Salla | Kolları Çaprazlama | Tekme | Alma | Yumruk | Oturma | Dönme | Kafa Kaşıma | Yürüme |
|--------------------|-----------|----------|--------------------|-------|------|--------|--------|-------|-------------|--------|
| Saate Bak | 85.1 | 0 | 14.9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| El Salla | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Kolları Çaprazlama | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Tekme | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| Alma | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 | 0 | 0 |
| Yumruk | 0 | 0 | 0 | 0 | 4.8 | 95.2 | 0 | 0 | 0 | 0 |
| Oturma | 0 | 0 | 0 | 0 | 0 | 0 | 100 | 0 | 0 | 0 |
| Dönme | 0 | 0 | 0 | 0 | 0 | 0 | 5.3 | 86.8 | 7.9 | 0 |
| Kafa Kaşıma | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14.9 | 76.6 | 8.5 |
| Yürüme | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2.7 | 97.3 |

Çizelge 3.1.-3.5. arasında yer alan sonuçlar 10 çubuklu (bin) histogramlar kullanılarak elde edilmiştir. Farklı çubuk sayısı kullanımının sınıflama üzerindeki başarımını görmek için, 5, 10, 15, 20, 15 ve 30 çubuklu histogramlarla da testler yapılmıştır. Çizelge 3.6.'da farklı çubuk sayıları kullanılarak elde edilen sonuçlar gösterilmektedir. Çizelge 3.6.'da yer alan sonuçlar Test-2 ile elde edilmiştir. Çizelge 3.6 incelendiğinden 5 ve 10 çubuklu histogramlar yakın sonuçlar üretmelerine rağmen, en başarılı sonuçlar 10 çubuklu histogramlar ile elde edilmiştir.

Çizelge 3.6. Test-2 de farklı çubuk sayıları için sınıflama sonuçları

| #Bin | 5 | 10 | 15 | 20 | 25 | 30 |
|------|-------|-------|-------|-------|-------|-------|
| SVM | 86.27 | 86.30 | 86.33 | 85.63 | 86.11 | 84.44 |
| RF | 93.40 | 93.50 | 88.89 | 89.44 | 89.44 | 89.33 |

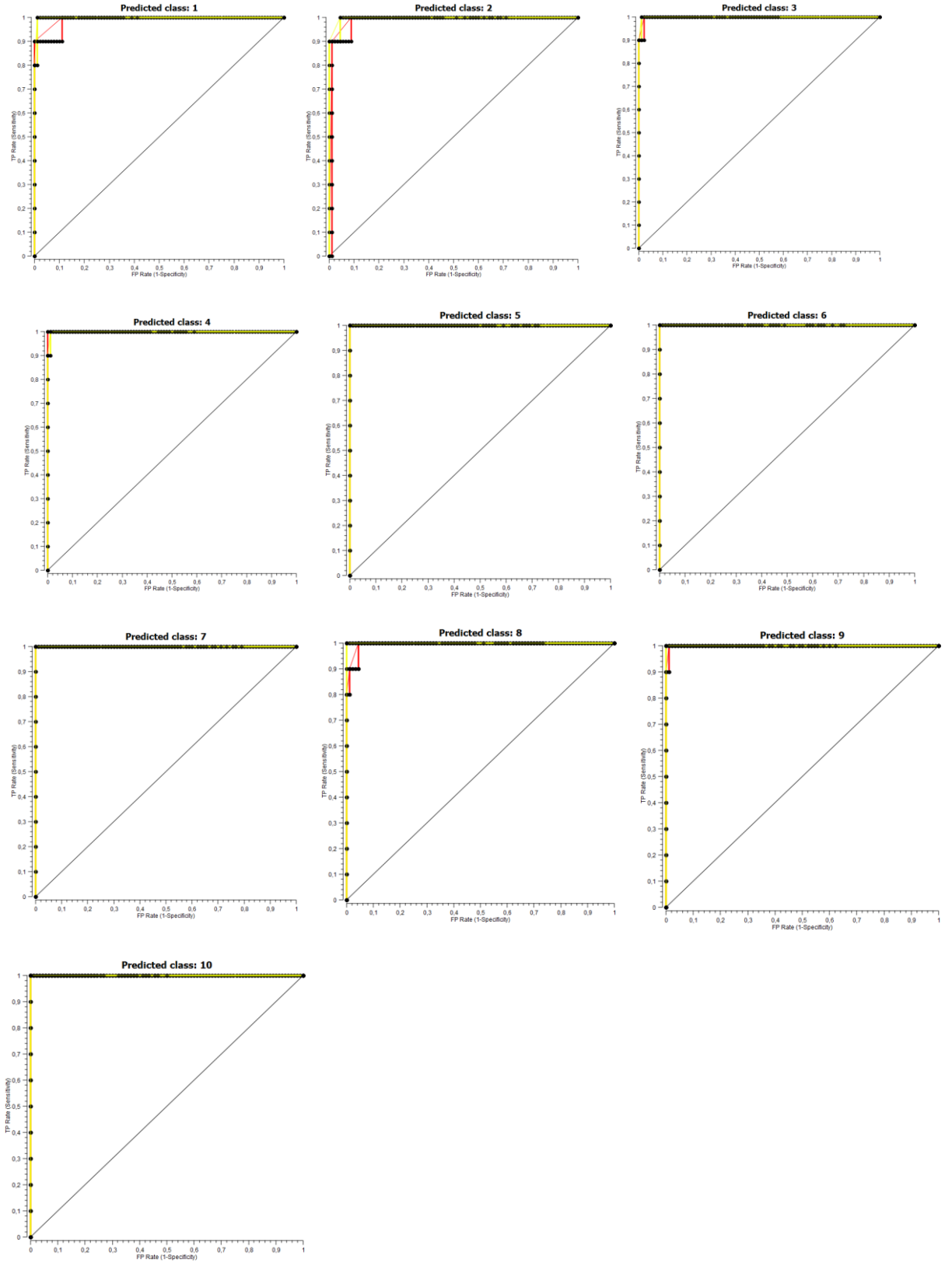
Çıkarılan öznitelikler üzerinde Sparse Multinomial Logistic Regression via Bayesian L1 Regularisation yöntemi [77] kullanılarak boyut indirilmesi de yapılmıştır. Boyut indirgeme işlemiyle işlem performansını arttırmaya çalışılmıştır. 102 sütundan oluşan öğrenme verisinden 27 sütunluk bir öğrenme verisi elde edilmiştir. Eğitim verisinin küçülmesi modelin eğitim süresi üzerinde iyileştirme sağlamıştır. Sınıflama başarımında ise düşme görülmüştür. Boyut indirgeme yapıldıktan sonra elde edilen sınıflama değerleri Çizelge 3.7.'de verilmiştir.

Çizelge 3.7. Öznitelikler üzerinde boyut indirgeme yapıldıktan sonra sınıflama sonuçları

| | CA | Sens | AUC |
|------------|------|------|------|
| SVM | 81.0 | 96.0 | 98.0 |
| RF | 92.0 | 97.0 | 99.0 |

Sınıflama sonuçlarına ait ROC eğrileride incelenmiştir. ROC eğrisi, sınıflama testinin kendi doğruluğunu tanımlaması ve testler arasında güvenilir bir karşılaştırma yapmaya olanak sağlaması açısından sıklıkla kullanılmaktadır. Sürekli sayıların kullanıldığı ölçümlerde sınıfları ayırma çözümlemeyi karmaşık hale getirir ve hata olasılığını yükseltir. ROC analizi çeşitli durumlarda optimum eşik değerini ve yapısında var olan değerlendirme dışı bırakılacak olan değerleri (duyarlılık ve özgüllük arasında yer alan) belirler. ROC eğrisinin grafiksel yaklaşımı ölçümlerin duyarlılığı ve özgüllüğü arasındaki ilişkileri kavramayı kolay kılar.

HUN3D üzerinde Test-2 de RF ve SVM için ROC eğrileri Şekil 3.12.'de verilmiştir. ROC eğrileri incelendiğinde HUN-3D üzerinde sınıflama başarımının yüksek olduğu görülmektedir. Bunun sebebi bu veri kümesinde aktör çeşitliliğinin az olmasıdır. Bir kişinin aynı hareketleri yaparken fazla benzerlik göstermesi başarımı arttırmıştır. ROC eğrilerindeki sınıf numaraları Kesim 3.2 de verilen hareket numaralarıdır.



Şekil 3.12. SVM ve RF yöntemleri için ROC eğrileri.

3.5.2. MSRC-12 ve MSR-Action 3D Veri Kümeleri İçin Deney Sonuçları

HUN-3D veri kümesinde yapılan testlerden sonra, geliştirilen model Microsoft tarafından da yayınlanan MSRC-12 ve MSR-Action 3D veri kümeleri ile de test edilmiştir. Testler sırasında hem eklem açılarına ait histogramlar hem de eklem noktalarının yer değiştirme bilgisi öznitelik olarak kullanılmıştır. Bu testlerde Test-1 ve Test-2 ye ek olarak Cross-Subject Test de kullanılmıştır. Bu testte model farklı kişilere ait hareketler ile eğitilip, eğitim kümesi dışında kalan kişilere ait hareketlerle test edilmektedir. Böylece modelin kişiye bağlı olup olmadığı test edilmektedir. MSRC-12 veri kümesine ait sınıflama sonuçları Çizelge 3.8.'de gösterilmektedir. Bu test sırasında 15 kişiye ait hareketler eğitim, geri kalan 15 kişiye ait hareketler ise test sırasında kullanılmıştır. Veri kümesinde 30 kişi yer alması ve 30'un 15'li kombinasyonlarının sayıca büyüklüğü nedeniyle tüm kişi kombinasyonları için test yapılamamıştır. Cross-Subject Test sadece rastgele seçilmiş 500 kombinasyon için RF ve SVM modelleri için yapılmıştır. Bu testte yüksek başarı oranlarına erişmek kişilerin aynı hareketlerde gösterdikleri farklılıklar nedeniyle oldukça zordur. Test sırasında erişilen ortalama başarılı sınıflama oranları ve standart sapma değerleri Çizelge 3.8.'de belirtilmiştir. Tez çalışmalarının bu ve ilerleyen kesimlerinde de literatürde yer alan yöntemlerle karşılaştırılabilir olması sebebiyle MSRC-12 ve MSRAction 3D yöntemleri üzerinde testler yapılmıştır.

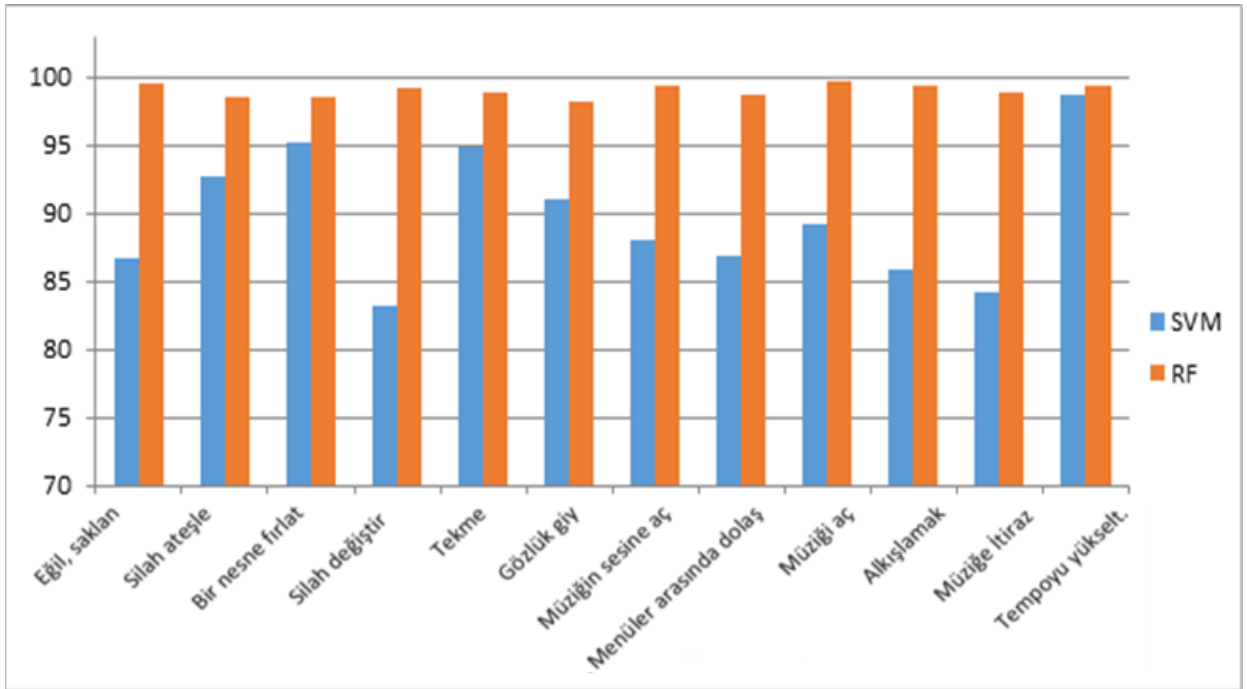
Çizelge 3.8. MSRC-12 veri kümesi için RF ve SM modellerinde sınıflama sonuçları

| | Test-1 | Test-2 | Cross-Subject |
|-----|--------|--------|---------------|
| SVM | 89.68 | 93.24 | 73.30±9.71 |
| RF | 97.84 | 98.75 | 83.03±8.44 |

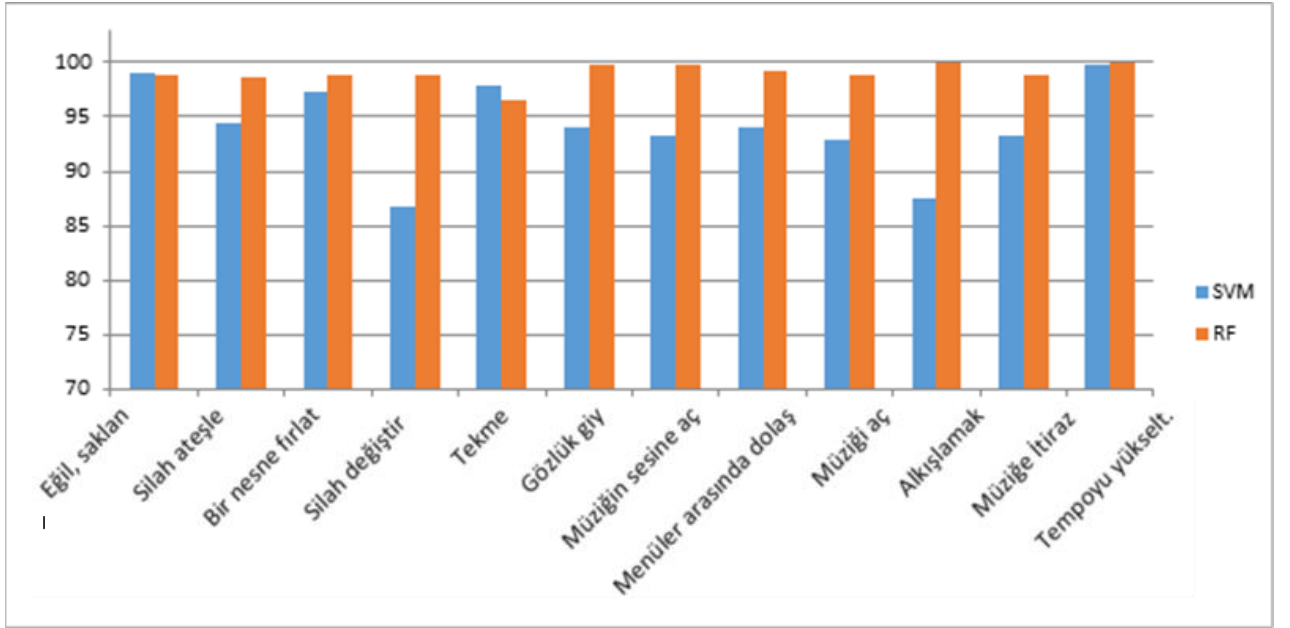
HUN-3D ile kıyaslandığında Test-1 ve Test-2 de doğru sınıflama oranında bir iyileşme görülmektedir. Bunun nedeni HUN-3D de yer alan gürültülü örnekler ve Shotton ve diğerleri [7] tarafından geliştirilen yöntemin bu veri kümesinde yer alan bazı hareketler için yanlış iskelet modelleri üretmesidir ki buda sınıflama oranını düşürmüştür. MSRC-12 ve MSR-Action 3D veri kümeleri büyük çoğunlukla daha gürültüsüz ve hatasız örnekler içermektedir. Bu durumda bu veri kümeleri ile daha yüksek sınıflama oranlarının yakalanmasına neden olmaktadır. Ayrıca HUN-3D de

yer alan hareket sayısının azlığı modelin eğitilmesi için yeterince büyük bir eğitim kümesi oluşturmaya engel olmaktadır.

Çizelge 3.8.'de görüldüğü üzere doğru sınıflama oranı RF modeli için Test-1 ve Test-2 de 97.8% ve 98.7% oranına ulaşmıştır. SVM modelinin performansı kabul edilebilir seviyede olmasına karşın RF modelinden geri kalmıştır. Şekil 3.13. ve 3.14.'te RF ve SVM sınıflayıcılarının MSRC-12 de ki hareketler üzerindeki sınıflama sonuçları gösterilmektedir. Kırmızı çubuklar hareketler üzerinde RF ile elde edilen doğru sınıflama oranını mavi çubuklar ise SVM ile elde edilen doğru sınıflama oranını belirtmektedir. Şekil 3.13. ve Şekil 3.14. de görüldüğü üzere RF modelinde MSRC-12 için Test-1 ve Test-2 de 95% üzerine çıkmıştır. Test-1 de SVM modelinde fırlatma, müziği açma hareketlerinde sınıflama oranı düşük olmasına karşın 80% düzeyindedir.



Şekil 3.13. MSRC-12 de Test-1 için sınıflama oranları



Şekil 3.14. MSRC-12 de Test-2 için sınıflama oranları

Cross-Subject Test te doğru sınıflama oranı Test-1 ve Test-2 ye kıyasla düşmektedir. Bu beklenen bir durum olmakla birlikte kişilerin hareketleri gerçekleştirirkenki karakteristik özelliklerinden kaynaklanmaktadır. Bu sebeple eğitim kümesinde kullanılan hareketler test kümesinden farklı özneliklere sahip olabilmektedir ve bu durumda modelin hareketi iyi ifade edebilmesi engel teşkil edebilmektedir. Bu testler kişiler arası farklılıkların ciddi oranda hataya neden olabildiğini göstermiştir.

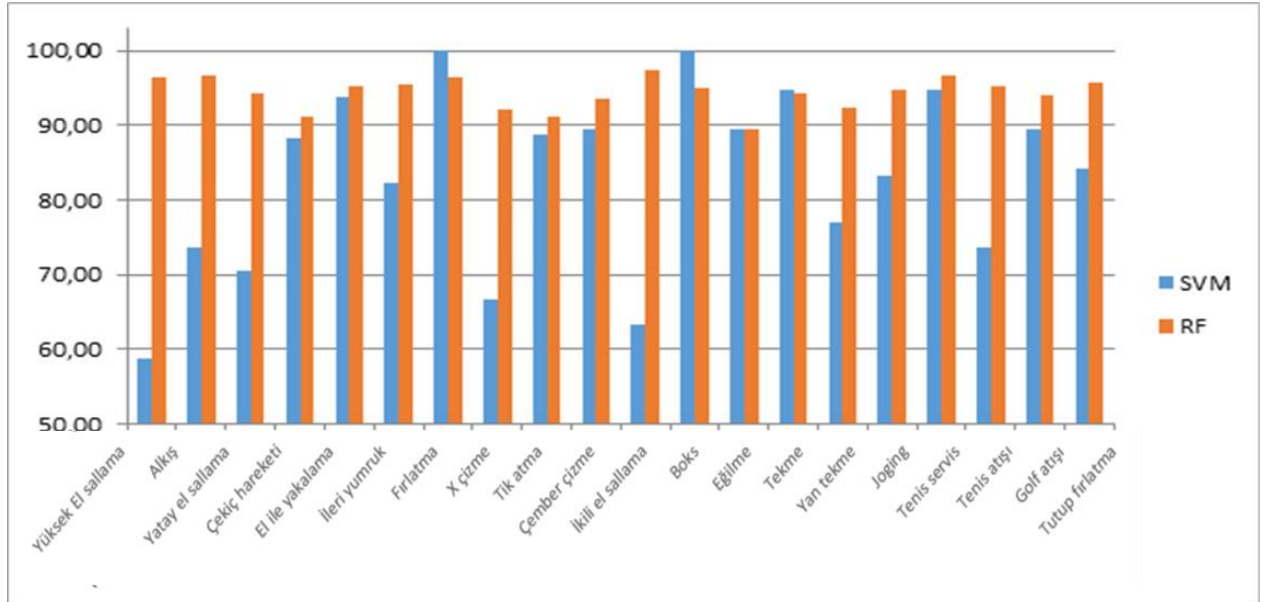
MSR-Action 3D için test sonuçları Çizelge 3.9.'da belirtilmiştir. Cross-Subject Test sırasında kişilerin C(10,5) kombinasyonlarının tamamı kullanılmıştır. MSRC-12 karşılaştırıldığında doğru sınıflama oranı Test-1 ve Test-2 için düşmüştür. Bunun nedeni bu veri kümesinin daha fazla türden hareket içermesi ve bazı hareket türlerinin de birbirlerine daha yakın olmasıdır. Bu sebeple doğru sınıflama oranı düşmüştür. Fakat Cross-Subject Test'te ise MSRC-12 ye göre daha başarılı sonuçlar elde edilmiştir. Bunun nedeni MSRC-12 de yer alan farklı kişilere ait aynı hareketler arasındaki farklılık düzeyinin fazla olmasıdır. MSR-Action 3D veri kümesinde yer alan kişiler aynı hareketleri benzer şekilde uygulanmaktadır. Bu durum Cross Subject Test'te daha başarılı sonuçlar elde edilmesine olanak sağlamaktadır. MSRC-12 de

ise bu farklılıklar öznelilikler üzerinde yüksek sapmalara neden olmakta ve doğru sınıflama oranını düşürmektedir.

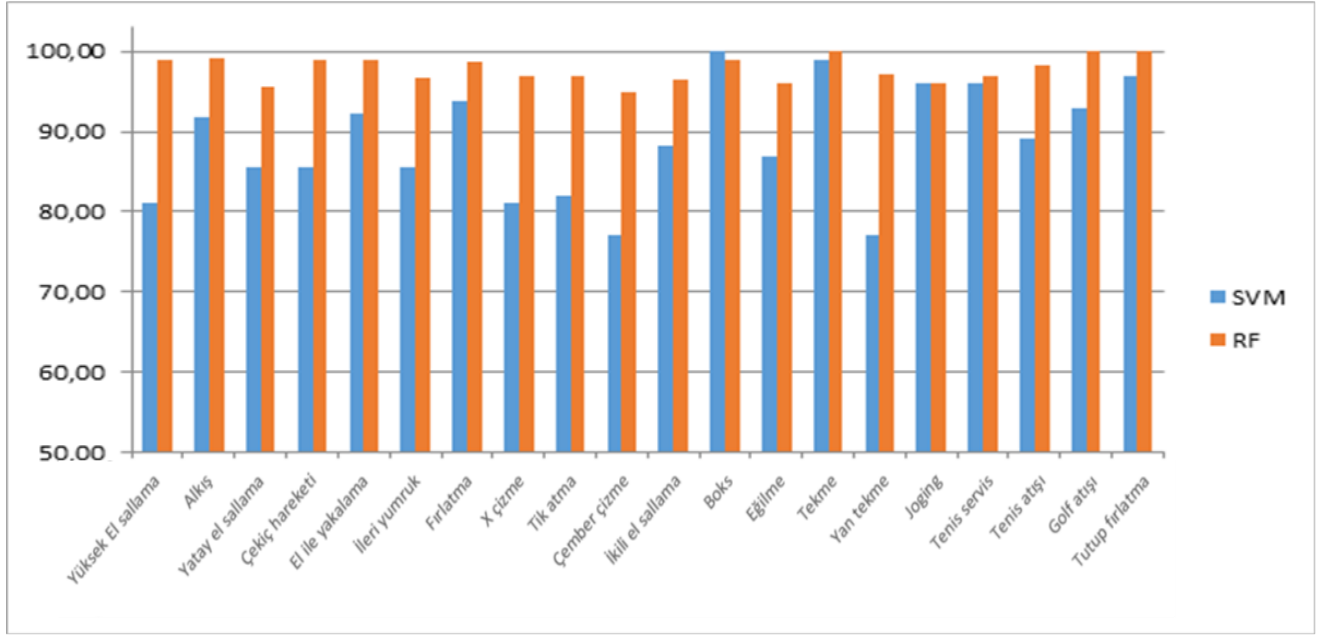
Çizelge 3.9. MSR-Action 3D veri kümesi için RF ve SVM modellerinde sınıflama sonuçları

| | Test-1 | Test-2 | Cross-Subject |
|------------|--------|--------|---------------|
| SVM | 83.19 | 89.12 | 81.17±8.47 |
| RF | 95.35 | 97.77 | 88.19±7.39 |

Şekil 3.15. ve Şekil 3.16.'da MSR-Action 3D için Test-1 ve Test-2 ile de edilen sonuçlar gösterilmektedir. Yine kırmızı çubuklar RF mavi çubuklar SVM modeli ile elde edilen sınıflama sonuçlarını göstermektedir. Tüm hareketler için ayrı ayrı düşünüldüğünde bile RF modeli halen daha başarılı sonuçlar üretmektedir. Tüm hareketlerde RF modeli için başarı düzeyi 90% ve 95% seviyelerindedir. Fakat SVM modelinde bazı hareketlerin sınıflama oranı Test-1 de 60%, 65% oranına kadar gerileyebilmektedir. Test-2 de SVM modeli daha başarılı sonuçlar üretmesine karşın halen RF modelin gerisinde kalmaktadır.



Şekil 3.15. MSRACTION-3D de Test-1 için sınıflama oranları



Şekil 3.16. MSR-Action 3D de Test-2 için sınıflama oranları

Çizelge 3.10. Geliştirilen Yöntemin Literatürde yer alan diğer yöntemler ile karşılaştırılması

| | Test-1 | Test-2 | Cross-Subject Test |
|----------------------|-------------|-------------|--------------------|
| Bag of 3D Points [3] | 91.6 | 94.2 | 74.7* |
| HOJ3D [78] | 96.2 | 97.2 | 79.0* |
| EigenJoints [68] | 95.8 | 97.8 | 83.3* |
| Yang et al. [64] | N/A | N/A | 85.5* |
| Wang et al. [70] | N/A | N/A | 88.2* |
| HON4D [66] | N/A | N/A | 82.15±4.18 |
| Bu çalışmada | 95.3 | 97.7 | 88.2±7.39 |

RF Modelin literatürde yer alan diğer yöntemler ile karşılaştırılması Çizelge 3.10.'da verilmiştir. Çizelgede ilgili yöntemlerin MSR-Action 3D ile elde edilen sonuçları belirtilmektedir. Pek çok çalışmada kişilerin yarısı test yarısı eğitim amacıyla kullanılmıştır. Yani hareketlere ait sadece tek kombinasyon alınmıştır. Bu çalışmalar ‘*’ işareti ile çizelgede gösterilmiştir. ‘**’ işareti bulunmayanlar ise tüm kombinasyonları kullanan çalışmalardır. Test ve eğitim kümelerinin farklı kişilerin kombinasyonlarından oluşması geliştirilen modelin eğitim verisinden bağımsız ve yanlı olmadığını gösterir. Yukarıdaki sonuçlara göre, SVM ve RF algoritmaları içinde, RF algoritması sınıflama doğruluğu ile daha başarılı bir sonuç vermiştir. Eklem açılarının ve eklem yer değiştirmelerinin öznitelik olarak birlikte kullanımı daha başarılı sonuçlar vermektedir.

3.6 Sonuçlar

Sonuç olarak, Microsoft Kinect algılayıcı kullanılarak, eklem iskelet modeli elde edilmiş ve modelden öznitelik çıkarılarak hareket tanınmaya çalışılmıştır. Öznitelik olarak eklemlerin açılarına ilişkin histogramlar ve eklem noktalarının yer değiştirme bilgileri kullanılmıştır.

Hareketleri sınıflama için SVM ve RF modelleri oluşturulmuştur. Bu modeller basit ve hesaplaması kolay öznitelikler ile başarılı sonuçlar üretmişlerdir. Test-2’de HUN-3D, MSRC-12 ve MSR-Action 3D için 93.5%, 98.7% ve 97.7% başarı oranlarına erişilmiştir. Yöntemimizin başarılı sonuç üretmesindeki etkenlerden biri, Shotton ve diğerlerinin [7] yöntemindeki eğitilmiş bir modelden faydalanarak öznitelik çıkarılıyor olmasıdır. İskelet modelinde tüm noktaları değil de, sadece eklemlerle ilgili noktaları takip edip bu noktalar ile ilgili öznitelikler çıkarılması başarıyı arttırmaktadır. Öznitelik kümesi elde edildikten sonra tahmin etmeye yönelik modeller oluşturulmuştur. Model oluşturulması adımlarında SVM ve RF yöntemleri kullanılmıştır. RF kullanılarak elde edilen model daha başarılı sonuçlar üretmektedir. İskelet modelinden elde edilen niteliklere ek olarak hareketlerin sınıflanmasında zaman bilgisini de kullanmak amacıyla ilerleyen adımlarda, Saklı Markov Modelleri (*Hidden Markov Model*) ve zaman serileri kullanılarak hareketler tanınmaya çalışılmıştır. Bu iki yaklaşımda zamansal örüntülerin (*temporal pattern*) tanınmasında literatürde yaygın olarak kullanılan yöntemlerdir. Bu kesimde kullanılan açı histogramları belirli düzeyde zaman bilgisini de işin içine katmaktadır. Fakat hareketi tümüyle zamansal bir örüntü

olarak ele alan yaklaşımların sınıflama başarımına etkisini anlamak amacıyla bu yöntemler çalışılmıştır.

4. Saklı Markov Modelleri İle Hareket Tanıma

Bu kesimde, Saklı Markov Modelleri kullanılarak iskelet modeli üzerinden çıkarılan öznitelikler ile insan hareketlerinin tanınması için yapılan çalışılmalar açıklanmaktadır. Saklı Markov Modelleri hareket tanıma çalışmalarında sık kullanılan yöntemlerdendir. Özellikle değişken uzunluktaki serileri tanımada sıkça başvurulan yöntemlerdendir. Bu bölümde ilk olarak Saklı Markov Modellerine ilişkin temel kavramlar anlatılmış daha sonra elde edilen öznitelikler ile yapılan tanıma işlemi ve deneylere değinilmiştir.

Saklı Markov Modelleri (SMM), bir örüntüye ait çerçevelerin izgesel özellikleri modellemede yaygın olarak kullanılan yöntemlerden biridir [79]. SMM (veya diğer istatistiksel yöntemler) bir örneklemede bulunan elemanların diziliminin birbirine bağımlı olduğu durumlarda kullanılır. Örneğin bir dilde harfler belli bir sırada yer almaktadır. Yine aynı şekilde DNA gen dizileri de rastgele bir sırada değildir. Bu tür özelliklere sahip verilerin sınıflandırılmasında SMM yaygın olarak kullanılmaktadır. SMM modelleri gelecek durumların geçiş durumlarından bağımsız olduğu Markov zincirlerini temel almaktadırlar.

4.1. Markov Zincirleri

Markov zincirleri ayrık zamanlı, durumlar arası geçişin sadece bulunulan duruma bağlı olarak tanımlandığı süreçlerdir. Markov zincirleri model λ ile tanımlanırsa aşağıdaki şekilde ifade edilmektedir.

$$\lambda = \{S, A, \pi\}$$

λ modelinde

S : N durumdan oluşan durum kümesi

A : NxN'lik durumlar arası geçiş matrisi. Bir durumdan diğer duruma geçme olasılığını belirtmektedir.

π : N boyutlu başlangıç durum vektörü, başlangıç anında herhangi bir durumda olma olasılığı olarak tanımlanmaktadır. Model, $Q = \{q_1, q_2, q_3 \dots, q_T\}$ T adet durum gözlem sırası olmak üzere Eşitlikler 4.1, 4.2 ile verilen koşulları sağlar.

$$P(q_t = S_j | q_{t-1} = S_i | q_{t-1} = S_k, \dots) = P(q_t = S_j | q_{t-1} = S_i) \quad (4.1)$$

$$P(q_t = S_i | q_{t-1} = S_j) = P(q_{t+l} = S_i | q_{t+l-1} = S_j) \quad (4.2)$$

Durum geçiş matrisi A, eşitlikler (4.3-4.6) ile ifade edilmektedir [79].

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) \quad 1 \leq i, j \leq N \quad (4.3)$$

$$A = [a_{ij}] \quad (4.4)$$

$$a_{ij} \geq 0 \quad \forall i, j \quad (4.5)$$

$$\sum_{j=1}^N a_{ij} = 1 \quad \forall i \quad (4.6)$$

Geçiş matrisinin tanımladığı geçişlere göre farklı çeşitlerde Markov zincirleri bulunmaktadır. Geçiş matrisinin değişimine göre temel 3 çeşit geçiş Markov zinciri bulunmaktadır. Bunlar ergodik, soldan sağa ve sağdan sola markov zincirleridir. Başlangıç durum vektörü π_i ise Eşitlik 4.7- 4.8 ile ifade edilmektedir.

$$\pi_i = P(q_1 = S_i) \quad 1 \leq i \leq N \quad (4.7)$$

$$\pi_i \geq 0 \quad \forall i \quad (4.8)$$

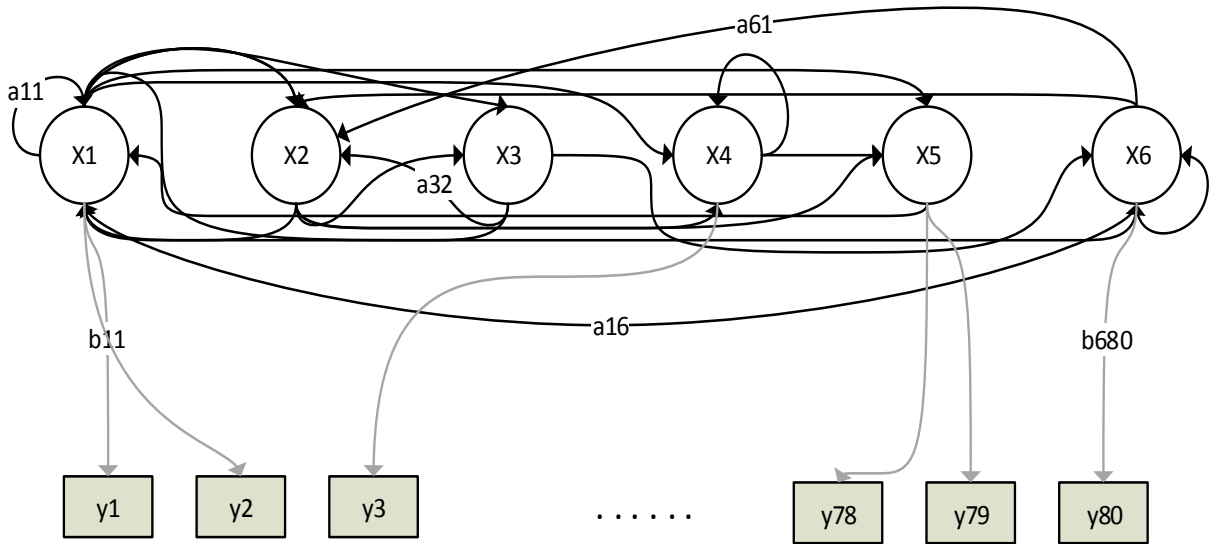
$$\sum_{i=1}^N \pi_i = 1 \quad (4.9)$$

Yukarıdaki eşitliklerden hareketle bir $Q = \{q_1, q_2, q_3, \dots, q_T\}$ gözlem dizisi verildiğinde bu dizinin gerçekleşme olasılığı Eşitlik 4.10 ile hesaplanmaktadır.

Burada A'dan sonra B'nin gözlenme olasılığı belirtilmektedir. Bu olasılık hesabı tüm dizi boyunca ardışık olarak hesaplanmaktadır. Bu hesaplama eşitlik 10 ile koşullu olasılık kuralı doğrultusunda, tek tek parçalara ayrılarak ve her adımda koşullu

olasılıklar, birinci dereceden Markov zinciri özelliği uygulanarak durumlar arası geçiş olasılıkları indirgenilmektedir. $P(q_1)$ ise başlangıç durumu olasılığını belirtmektedir (Eşitlik 4.10). MSRC-12 veri kümesi üzerinde kullanılan SMM'lerden birine ait durumlar, durumlara ait çıktı olasılıkları ve geçiş olasılıkları Şekil 4.1.'de gösterilmektedir. Gözlem olasılığı, belirli bir durumda iken belirli bir gözlemin görülme olasılığını; geçiş olasılığı ise bir durumdan diğer başka bir duruma geçme olasılığını belirtmektedir. Şekil 4.1.'de durumlar X_i , gözlem olasılıkları b_{ij} , gözlemler Y_k ve geçiş olasılıkları a_{ij} ile gösterilmiştir.

$$\begin{aligned}
 P(Q) &= P(q_1, q_2, q_3, \dots, q_T) & (4.10) \\
 &= P(q_T | q_1, q_2, q_3, \dots, q_{T-1}) P(q_1, q_2, q_3, \dots, q_{T-1}) \\
 &= P(q_T | q_{T-1}) P(q_1, q_2, q_3, \dots, q_{T-1}) \\
 &= P(q_T | q_{T-1}) (q_{T-1} | q_{T-2}) P(q_1, q_2, q_3, \dots, q_{T-2}) \\
 &\vdots \\
 &= P(q_T | q_{T-1}) (q_{T-1} | q_{T-2}) \dots P(q_2 | q_1) P(q_1) \\
 &= \pi_{i=q_1} \prod a_{q_{T-1} q_T}
 \end{aligned}$$



Şekil 4.1. MSRC-12 için kullanılan bir SMM, X'ler durumlar a'lar geçiş olasılıkları ve b'ler gözlem olasılıklarıdır.

4.2. Saklı Markov Modeli

Markov zincirlerinden farklı olarak Saklı Markov Modeli'nde durumlar doğrudan gözlenebilir değildir; gözlemlerin gerçek durumların bir olasılık fonksiyonu olarak gerçekleştiği varsayılır. Saklı Markov Modeli $\lambda = \{S, V, A, B, \pi\}$ ile tanımlanabilir ve Saklı Markov Modeli'nde gözlemler, durumlar, durumlar arası geçişler ve saklı durumlar yer almaktadır.

Durum kümesi, $S = \{S_1, S_2, S_3, \dots, S_N\}$ ile gösterilmekte ve t anında bulunulan durum q_t ile ifade edilmektedir. M: kesikli gözlemlerini ifade etmektedir. Gözlem sembolleri modelin çıktılarını ifade etmektedir. Gözlem kümesi ise (*Emission*) $V = \{V_1, V_2, V_3, \dots, V_M\}$ ile gösterilmektedir. O_t t anında üretilen gözlemi belirtmektedir. Durumlar arası geçiş matrisi A bir durumdayken diğer bir duruma geçme olasılığını ifade etmektedir. Diğer bir deyişle t anında i. durumda bulunulup t+1 anında j. durumda bulunma olasılığını ifade eder. $A = [a_{ij}]$ Eşitlik 4.3-4.6 ile ifade edilmektedir. B: Gözlem sembolleri olasılık dağılımıdır. t anında j. durumda t. gözleminde k sembolünün üretilme olasılığıdır ve Eşitlik 4.11 ile ifade edilmektedir.

$$b_j(k) = P(O_t = v_k | q_t = S_j) \quad \begin{array}{l} 1 \leq i \leq N \\ 1 \leq k \leq M \end{array} \quad (4.11)$$

Doğrudan gözlenemeyen durumların birer olasılık dağılımı olarak meydana getirdiği gözlem sembollerinin gerçekleşme olasılıkları $b_j(k)$ Eşitlik 4.12-4.13 ile tanımlanır.

$$b_j(k) \geq 0 \quad \forall j, k \quad (4.12)$$

$$\sum_{k=1}^M b_j(k) = 1 \quad \forall k \quad (4.13)$$

π : Başlangıç durumu olasılık vektörüdür. Sözel olarak i. durumun ilk durum olma olasılığıdır.

$\pi = [\pi_i]$ Eşitlik 4.14 ile tanımlanmıştır.

$$\pi = [\pi_i] = P(q_1 = S_i) \quad 1 \leq i \leq N \quad (4.14)$$

Saklı Markov Modeli için üç problem tanımlanabilir: Verilen $O = O_1, O_2, O_3, \dots, O_T$ gözlem sırası ve $\lambda = \{A, B, \pi\}$ modeli için $P(O|\lambda)$ gözlem sırasının λ modeli ile gerçekleşme olasılığının hesaplanması, $O = O_1, O_2, O_3, \dots, O_T$ gözlem sırası ve $\lambda = \{A, B, \pi\}$ modeli için en anlamlı $Q = q_1, q_2, q_3, \dots, q_T$ durum sırasının hesaplanması ve aynı gözlem sırası ve $\lambda = \{A, B, \pi\}$ modeli için $P(O|\lambda)$ olasılığını maksimize edecek şekilde model parametrelerinin tahmin edilmesi. Bu problemlerden ilki olan gözlem dizisini modelin gerçekleştirme olasılığı İleri/Geri algoritmasıyla hesaplanmaktadır.

4.2.1. İleri/Geri Algoritması

Gözlem dizisinin belirli modeller tarafından gerçekleştirilmiş olması olasılığını hesaplamaktadır. Bu algoritma için $\alpha_t(i)$ değişkeni, t anına kadarki k gözlem sırası sonucunda S_i durumuna varılması olasılığını ifade etmektedir. $\alpha_t(i)$ değişkeni, eşitlik 4.15 ile de ifade edilmektedir. Bu algoritma kısaca özetleyecek olursak 3 temel adımdan oluşmaktadır. Bunlar ileri olasılıkların hesaplanması, geri olasılıkların hesaplanması ve hesaplanan değerlerin düzeltilmesidir.

Tüm durumlar arasındaki geçişlerin olduğu modellerde geçiş matrisinin hesaplanması için gereken işlem karmaşıklığı yüksek olacaktır. Karmaşıklığı fazla olduğu durumlar için uygulamaya bağlı olarak model belirlenirken geçişlerin sınırlı olduğu topolojiler seçilmelidir. Böylelikle ileri/geri algoritmasının karmaşıklığı düşürülmüş olacaktır. Örneğin ses uygulamalarında soldan sağa Markov modelleri kullanılmaktadır. Bu modellerde girdi üzerinde hareket edildikçe aynı durumda kalınmakta yada bir sonraki duruma geçilmektedir. Sonraki durumlara geçiş olasılıkları Eşitlik 4.15 ile belirtilmektedir

$$\alpha_t(i) = P(O_1, O_2, O_3, \dots, O_t, q_t = S_i | \lambda) \quad (4.15)$$

İleri algoritmasının adımları aşağıdaki eşitlikler (4.16,.4.17,4.18) ile ifade edilmektedir

$$\text{Başlangıç : } \alpha_1(i) = \pi_i b_i(O_1) \quad (4.16)$$

$$\text{Çıkarım: } \alpha_{t+1}(j) = \sum_{i=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \quad (4.17)$$

$$\text{Bitiş: } P(O|\lambda) = \sum_{i=1}^N \alpha_T(i) \quad (4.18)$$

İleri algoritması ile i durumunun başlangıç durumu olması ve bu durumun gözlenen O_1 sembolünü üretmesi ihtimali $\alpha_1(i)$ değeri ile ifade edilir. Bir sonraki adımda tüm durumlardan j durumuna geçiş ve bu yeni durumda O_{t+1} gözlem sembolünün üretilme olasılığı hesaplanır. En nihayetinde ise gözlem sırasının gerçekleşme olasılığı $\alpha_T(i)$ tüm durumlar üzerinden kümülatif olarak toplanarak hesaplanır. Geri algoritmasında ise $\beta_t(i)$ nin t anında S_i durumunda bulunulup $t+1$ anından itibaren sonrası için gözlem dizisinin gerçekleşme olasılığı olarak tanımlanır ve Eşitlik 4.19 ile ifade edilir.

$$\beta_t(i) = P(O_{t+1}, O_{t+2}, \dots, O_T | q_t = S_i, \lambda) \quad (4.19)$$

Geri algoritması aşağıda bahsedilen adımlarla gerçekleştirilir :

$$\text{Başlangıç} \quad : \beta_T(i) = 1 \quad 1 \leq i \leq N \quad (4.20)$$

$$\begin{aligned} \text{Çıkarım} \quad : \sum_{j=1}^N a_{ij} b_i(O_{t+1}) \beta_{t+1}(j) \quad 1 \leq i \leq N \quad (4.21) \\ 1 \leq t \leq T - 1 \end{aligned}$$

4.2.2. Viterbi Algoritması

Saklı Markov Modellerinde verilen bir girdiyi türeten durum dizisinin çıkarılmasında yaygın olarak kullanılan yöntemdir. Bu algoritma ile verilen bir $O = \{O_1, O_2, O_3, \dots, O_T\}$ gözlem dizisini üretme olasılığı en fazla olan durum sırası $Q = q_1, q_2, q_3, \dots, q_T$ elde edilir. Kısaca diziyi üretebilecek durum dizilerinden en olası olan seçilir.

$$\delta_t(i) = \max(P(q_1, q_2, q_3, \dots, q_t = i, O_1, O_2, O_3, \dots, O_t | \lambda)) \quad (4.22)$$

Eşitlik 4.22 ile O_1 'den O_t 'ye kadar olan gözlem sembollerini sağlayan ve t anında S_i durumunda bulunma olasılığı tanımlanmıştır.

$$\delta_{t+1}(j) = \max (\delta_t(i) a_{ij}) b_i(O_{t+1}) \quad (4.23)$$

Eşitlik 4.23 ile j durumuna hangi i . durumdan geçişin en yüksek olasılıkla gerçekleşebileceği hesaplanarak j durumunda O_{t+1} gözlem sembolü üretilme olasılığı δ_{t+1} olarak bulunur.

Durumlar arası en olası geçişler $\psi_t(j)$ ile tanımlanır ve son adımdan itibaren geriye doğru takip edilerek istenen durum dizisi elde edilir.

Viterbi algoritması aşağıdaki adımlardan oluşmaktadır [80].

Başlangıç:

$$\delta_1(i) = \pi_i b_i(O_1) \quad 1 \leq i \leq N \quad (4.24)$$

$$\psi_t(j) = 0 \quad (4.25)$$

Yineleme:

$$\delta_t(j) = \max (\delta_{t-1}(i) a_{ij}) b_i(O_t) \quad \begin{array}{l} 1 \leq i \leq N \\ 2 \leq t \leq T \\ 1 \leq j \leq N \end{array} \quad (4.26)$$

$$\psi_t(j) = \arg \max (\delta_{t-1}(i) a_{ij}) \quad \begin{array}{l} 1 \leq i \leq N \\ 2 \leq t \leq T \\ 1 \leq j \leq N \end{array} \quad (4.27)$$

Bitiş:

$$P^* = \max \delta_t(i) \quad 1 \leq i \leq N \quad (4.28)$$

$$q_T^* = \operatorname{argmax} \delta_t(i) \quad 1 \leq i \leq N \quad (4.29)$$

Yol Takibi:

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \quad t = T - 1, T - 2, \dots, 1 \quad (4.30)$$

4.2.3. Baum-Welch Algoritması

Baum-Welch algoritması Expectation maximization algoritmasının bir türevidir ve verilen bir gözlem dizisinden $x = \{O^k\}_{k=1}^K$ model parametrelerinin öğrenilmesinde kullanılır. İteratif olarak çalışan bu yöntemde her iterasyonda o anki $\lambda = \{N, M, A, B, \pi\}$ değerlerinden faydalanılarak t anında S_i durumunda olma ve t+1 anında S_j durumuna geçme olasılığı $\varphi_t(i, j)$ ve S_i durumundan olası tüm durumlara geçme olasılığı $\gamma_t(i)$ hesaplanır. $\varphi_t(i, j)$ ara değeri ve $\gamma_t(i)$ Eşitlik 4.31-4.32 ile hesaplanmaktadır.

$$\varphi_t(i, j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^M \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \quad (4.31)$$

$$\gamma_t(i) = \sum_{j=1}^N \varphi_t(i, j) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{i=1}^N \alpha_t(i) \beta_t(i)} \quad (4.32)$$

Markov modeli saklı olmasaydı, $\varphi_t(i, j)$ ve $\gamma_t(i)$ değerleri 0 yada 1 olurdu ve bilinirdi. Bu iki değer hesaplandıktan sonra bu değerleri kullanılarak λ değerini kestirmeye çalışılmıştır. Geçiş matrisini Eşitlik 4.33 ile tahmin edilmektedir.

$$\overline{a}_{ij} = \frac{\sum \varphi_t(i, j)}{\sum \gamma_t(i)} \quad (4.33)$$

Saklı markov modelinin diğer bir bileşeni olan gözlem olasılıkları matrisi Eşitlik 4.34 ile tahmin edilmektedir.

$$b_j(k) = \frac{\sum_{t=1, O_t=v_k}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(j)} \quad (4.34)$$

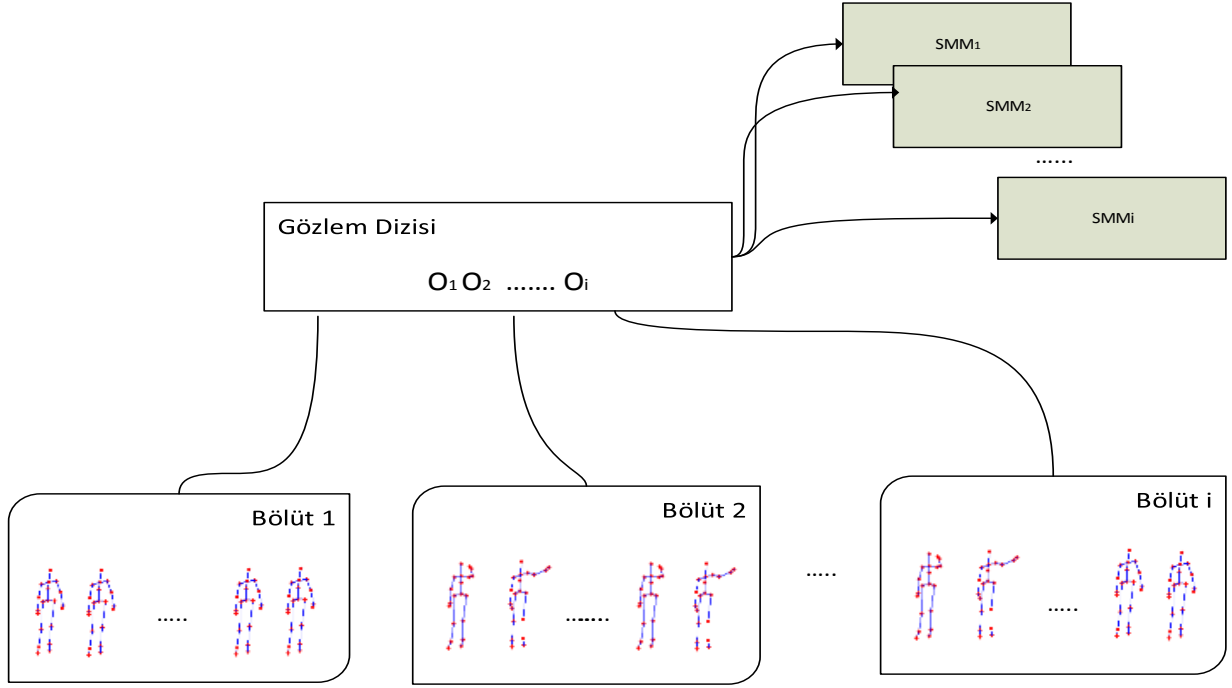
Belirtilen eşitlikler kullanılarak model parametreleri öğrenilir. Kısaca verilen gözlem dizisi için $P(X)$ değerini çıktı olarak veren parametreler bulunmuş olur.

4.3. SMM ile Hareket Tanıma

Öznitelik olarak Kesim 3'de kullanılan özniteliklerden olan eklem açıları ve eklemlerin yer değiştirme bilgileri kullanılmıştır. Omuz-dirsek-kol ve kalça-diz-ayak bileği arasındaki açı değerleri hareket boyunca tutulup bu açı değerlerine ilişkin histogramlar oluşturulmuştur. Açı değerleri hesaplanırken, elde edilen iskelet modelinde dirsek, omuz, diz, kalça, el ve ayak bileklerine ait 3 boyutlu koordinat değerleri kullanılmıştır. Bu koordinat değerleri kullanılarak nokta ikilileri doğru parçaları olarak varsayılmış ve bu doğru parçaları arasındaki açı değerleri hesaplanmıştır. Yer değiştirme bilgisi de yine Kesim 3 'te olduğu üzere eklemlerin 3 boyutta ayrı ayrı kattığı uzaklıklar ve toplam yer değiştirmesidir.

Kesikli bir SMM'de kesikli zamana ait diziler, durumları gözlenemeyen bir Markov sürecinin çıktıları olarak tanımlanır. Tanıma işleminde SMM kullanabilmek için öncelikle veri kümesini gözlem dizisi haline getirmek gerekmektedir. Bir SMM, N tane durumdan $S=\{S_1, S_2, S_3 \dots S_n\}$ ve M tane çıktı sembolünden $Y=\{Y_1, Y_2, Y_3 \dots S_m\}$ oluşur. Bu amaçla ilk olarak veri kümesinde yer alan hareketler zaman dilimlerine (time slice) bölümlenmiştir. Bölümleme işleminde eşit uzunlukta ve 10 çerçeveden oluşan bölütler (time slices) kullanılmıştır. Bölüt uzunluğu farklı uzunluklardaki bölütler test edilerek seçilmiştir. Çok uzun bölütlerin aynı kümelere düşme olasılığı oldukça düşüktür. Bölütler kısa tutulduğu zaman da ayırt edici özelliği azalmakta ve aynı kümeye düşme olasılığı artmaktadır. Bölüt içerisinden öznitelik çıkarılabilmesi için bölütlerin yine belirli bir uzunlukta olması gerekmektedir. Çok kısa bölütlerden ayırt edici öznitelikler çıkarılması mümkün olamayabilmektedir. Bu durumlar göz önüne alınarak en optimal sonuçlara 10 çerçevelik bölütler ile ulaşılmıştır. Elde edilen bölütler üzerinde yukarıda bahsettiğimiz eklem açılarına ait histogram ve yer değiştirme bilgileri çıkarılmıştır. Bölütler üzerinde bu öznitelikler çıkarıldıktan sonra bu

bölütlerin belli bir gözlem dizisi haline getirilmesi gerekmektedir. Bu amaçla bölütler arasında kümeleme işlemi yapılmıştır. K-means [81] yöntemi kullanılarak bölütler kümelendirilmiştir. Kümeleme işleminin amacı, yakın öznelik vektörlerini aynı kümeye yerleştirerek hareketi bir gözlem dizisi haline getirmektir. Hareketlerin bölütlenerek gözlem dizisi haline getirilmesi ve sonrasında her harekete ilişkin bir SMM ile bu dizinin gözlemlenme olasılığının hesaplanması Şekil 4.2. de gösterilmiştir.



Şekil 4.2. SMM ile hareket tanıma

4.3.1 K-means Kümeleme

K-means algoritması gözetimsiz (*unsupervised*) sınıflandırma algoritmaları içerisinde bilinen en basit kümeleme yöntemidir. İlk olarak 1967 yılında J.B. MacQueen tarafından ortaya atılmıştır [81]. Burada amaç, özellik çıkarımı yapılmış n adet gözlemin k tane kümeye ayrılmasıdır. Bu işlem ile amaçlanan küme içi benzerliğin maksimum kümeler arası benzerliğin ise minimum olduğu kümeleri bulmaktır. Yöntemin başarımını küme sayısı, küme merkezlerinin başlangıç değerleri ve kümeleme sırasında kullanılacak olan uzaklık ölçütü belirlemektedir.

K-means algoritması her verinin sadece bir kümeye ait olmasına izin vermekte ve merkez noktaların kümeleri temsil etme mekanizmasına dayanmaktadır. Küme sayısı kullanıcı tarafından verilirken başlangıç küme merkezleri rastgele seçilebilir.

Algoritma temelde dört adımdan oluşmaktadır. Algoritmayı oluşturan adımlar aşağıda verilmiştir:

1. Küme merkezlerinin belirlenmesi
2. Tüm nesnelerin kendilerine en yakın kümeye atanması
3. Küme merkezlerinin tekrar belirlenmesi
4. 2. ve 3. adımların merkez noktalar değişmeyene kadar tekrarlanması

K-means algoritmasının çalışma biçimi aşağıdaki örneklerle daha açık bir şekilde anlatılmıştır.

İlk olarak küme merkezleri belirlenir. Küme merkezleri rastgele seçilebileceği gibi nesnelerin ortalaması alınarak da belirlenebilir. Tez kapsamındaki çalışmalarda küme merkezleri rastgele seçilmiştir. İlk olarak tüm örnekler kendilerine en yakın olan kümeye atanır. Daha sonra yeni küme merkezleri belirlenerek kümeleme işlemine devam edilir. Bu işleme yeni küme merkezleri bir önceki adımda bulunan merkezlerle aynı oluncaya kadar devam edilir.

Bu yöntemin en büyük avantajı hızlı ve basit olmasıdır. Bu sebepten ötürü çok büyük veri kümeleri üzerinde rahatça kullanılabilir. Bu yöntemin en büyük eksikliği k değerini tespit edememesidir. Yöntemin diğer bir olumsuz tarafı ise her işletildiğinde farklı sonuç üretebilmesidir. Bunun nedeni, küme merkezlerinin birinci adımda rastgele belirlenmesidir.

K-means algoritmasında örneklerin küme merkezlerine olan uzaklığının hesaplanmasında farklı yöntemler kullanılabilir. Bu yöntemlerin en bilinenleri Euclidean, Manhattan ve Chebychev yöntemleridir. İki nokta arasındaki Euclidean uzaklığı Eşitlik 4.35 ile hesaplanmaktadır. Euclidean yöntemi en yaygın kullanılan uzaklık hesaplama yöntemidir. X_i ve Y_i 2 ayrı noktayı temsil etmektedir.

$$\sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad (4.35)$$

İki nokta arasındaki Manhattan uzaklığı, ise Eşitlik 4.36 ile hesaplanmaktadır. Manhattan uzaklığı boyutlar arasındaki ortalama farka eşittir.

$$\sum_{i=1}^n |X_i - Y_i| \quad (4.36)$$

Chebychev uzaklığı ise iki nesne arasındaki maksimum boyut farkına eşittir. Chebychev uzaklığı Eşitlik 4.37 ile hesaplanmaktadır.

$$\max |X_i - Y_i| \quad (4.37)$$

K-means yönteminin değerlendirilmesinde, karesel hata kriteri SSE de (*sum square error*) yaygın olarak kullanılır. En düşük SSE değerine sahip kümeleme en iyi sonucu verir. SSE Eşitlik 4.38 ile hesaplanmaktadır. x kümeleme işleminde yer alan bir örneği, m_i ise küme merkezlerini belirtmektedir [103]'den alınmıştır.

$$SSE = \sum_{i=1}^K \sum_{x \in C_i} \text{dist}^2(m_i, x) \quad (4.38)$$

4.3.2. Sınıflama

Her biri 10 çerçeveden oluşan bölütler kümelere ayrılırken, küme sayısı olarak 80 kullanılmış ve veri kümesi, 80 kelimelik (kodluk) bir sözlük haline getirilmiştir. Kümeleme işleminden sonra artık tüm bölütler 1-80 arasında etiket değerleri ile ifade edilmektedir. Böylece tüm hareketler, 1-80 arasında kodlar içeren diziler haline gelmiştir. Tüm hareketler etiket dizisi haline getirildikten sonra sınıflama işlemine geçilmiştir. Sınıflama yapılabilmesi için her hareket için ayrı bir SMM tanımlanmıştır. Küme sayısı yani sözlük boyu farklı deneme ve gözlemler sonucunda seçilmiştir. Küme sayısı 50 den az olduğunda gözlem dizileri arasındaki benzerlikler fazla olmakta, 100-200 arasında olması durumunda ise aynı harekete ilişkin diziler arasındaki farklılıklar fazla olmaktadır. Yapılan deneyler sonucunda optimal sözlük boyu 80 olarak belirlenerek, hem ayırt edicilik bakımından geniş hemde genele uygulanabilirlik açısından dar bir sözlük uzunluğu sağlanmıştır.

Eğitim aşamasında veri kümesi test ve eğitim olarak ikiye bölünmüştür. Kesikli diziler haline getirilen hareketler üzerinde Test-1, Test-2 ve Cross-Subject Testleri yapılmıştır. Eğitim dizileri kullanılarak saklı markov modellerine ait geçiş ve gözlem matrisleri tahmin edilmiştir. Bu matrisler, SMM modellerinde tahmin sırasında kullanılacaklardır. Daha sonrasında elde edilen matrisler kullanılarak test kümesinde yer alan dizilerin bu modeller tarafından oluşturulabilme olasılıkları hesaplanmış ve en yakın hangi modelin oluşturulması olasıysa o harekete ait etikete atanmıştır.

Bu yaklaşıma göre, MSR-Action 3D veri kümesi için 20 SMM modeli eğitilerek, Test-2 de %93 civarında bir doğru sınıflama oranına erişilmiştir. Sınıflama işlemine ilişkin hata matrisleri Çizelge 4.1. ve Çizelge 4.2. de görülmektedir. Çizelgeler incelendiğinde karışan hareketlerin daha ziyade 4 ve 5 numaralı hareketler olan yumruk ve ileri fırlatma hareketleri olduğu görülmektedir (Hareket numaraları 3. Kesimdeki sıra

numaralarıdır). Bu hareketler sağ el kullanılarak yapıldığından yakın gözlem dizileri üretmiş ve bu sebeple yanlış sınıflama oranları görece daha yüksek çıkmıştır. Bir sonraki adımda MSRC-12 veri kümesi üzerinde yapılan çalışmada, her bir hareketi temsil eden 12 farklı Saklı Markov Modeli eğitilmiştir. Test-2’de %88 civarında bir doğru sınıflama oranına erişilmiştir. MSRC-12 veri kümesi kullanılarak yapılan Test-1 ve Test-2 ile yapılan sınıflamaya ilişkin hata matrisi Çizelge 4.3.’de ve Çizelge 4.4.’te gösterilmektedir. MSRC-12 de Test-1’de hareketler arası yanlış sınıflamaya eğitim kümesinin kısalığı bu kümedeki hareketlerin benzerliği sebepleriyle yeterli bir gözlem sözlüğü oluşturulamaması sebep olmuştur. Çizelge 4.5.’te ise Test-1, Test-2 ve Cross-Subject Test yöntemleri ile elde edilen doğru sınıflama oranları verilmektedir. Bu sonuçlardan da anlaşılacağı üzere, SMM ile geliştirilen yöntem Kesim 3’de geliştirilen yöntemle göre daha başarısız olmuştur.

Çizelge 4.1. MSR-Action 3D nin Test-1 de SMM ile sınıflandırılması sonucu elde edilen hata matrisi

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|----|-------|------|-------|------|------|------|-------|------|------|-------|-------|------|------|-------|------|------|------|------|------|-------|
| 1 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 10.0 | 90.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | 0.0 | 0.0 | 0.0 | 72.0 | 27.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5 | 0.0 | 0.0 | 0.0 | 8.0 | 92.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 94.0 | 6.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.0 | 90.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 94.0 | 6.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 10 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 11 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 12 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 95.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 13 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 79.0 | 21.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 14 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 15 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 10.0 | 90.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 16 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 68.0 | 32.0 | 0.0 | 0.0 | 0.0 |
| 17 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 82.0 | 18.0 | 0.0 | 0.0 |
| 18 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 94.0 | 6.0 | 0.0 |
| 19 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 10.0 | 72.0 | 18.0 |
| 20 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 |

Çizelge 4.2. MSR-Action 3D nin Test-2 de SMM ile sınıflandırılması sonucu elde edilen hata matrisi

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|----|------|-------|------|------|-------|------|------|------|------|-------|------|-------|-------|-------|-------|------|------|------|------|-------|
| 1 | 82.0 | 18.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 2 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.0 | 6.0 | 82.0 | 12.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | 0.0 | 0.0 | 0.0 | 84.0 | 0.0 | 0.0 | 16.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 6 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 81.0 | 0.0 | 19.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 92.0 | 8.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 10.0 | 90.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 9 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 10.0 | 82.0 | 17.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 10 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 11 | 0.0 | 11.1 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 88.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 12 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.8 | 0.0 | 100.0 | 0.0 | 0.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 13 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 14 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 15 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 16 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 13.0 | 0.0 | 0.0 | 0.0 | 0.0 | 87.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 17 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 7.0 | 93.0 | 0.0 | 0.0 | 0.0 |
| 18 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 7.0 | 93.0 | 0.0 | 0.0 |
| 19 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 6.0 | 94.0 | 0.0 |
| 20 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 |

Çizelge 4.3. MSRC-12 nin Test-1 de SMM ile sınıflandırılması sonucu elde edilen hata matrisi

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|------|------|-------|------|------|------|------|------|------|------|------|------|
| 1 | 82.0 | 6.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 12.0 |
| 2 | 12.0 | 82.0 | 6.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | 0.0 | 0.0 | 7.0 | 86.0 | 7.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5 | 0.0 | 0.0 | 0.0 | 11.0 | 68.0 | 21.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 6 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 76.0 | 15.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 6.0 | 76.0 | 17.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 7.0 | 86.0 | 7.0 | 0.0 | 0.0 | 0.0 |
| 9 | 0.0 | 9.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 91.0 | 0.0 | 0.0 | 0.0 |
| 10 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 20.0 | 80.0 | 0.0 | 0.0 |
| 11 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 75.0 | 16.0 |
| 12 | 25.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 23.0 | 77.0 |

Çizelge 4.4 MSRC-12 nin Test-2 de SMM ile sınıflandırılması sonucu elde edilen hata matrisi

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|------|------|-------|-------|------|------|------|------|------|-------|------|------|
| 1 | 83.0 | 6.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 10.0 |
| 2 | 1.0 | 94.0 | 5.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 3 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 4 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 5 | 0.0 | 0.0 | 0.0 | 0.0 | 80.0 | 20.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 6 | 0.0 | 0.0 | 0.0 | 0.0 | 8.0 | 92.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 7 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 30.0 | 70.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 35.0 | 55.0 | 10.0 | 0.0 | 0.0 | 0.0 |
| 9 | 0.0 | 4.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 12.5 | 83.0 | 12.5 | 0.0 | 0.0 |
| 10 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 100.0 | 0.0 | 0.0 |
| 11 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 66.0 | 33.0 |
| 12 | 10.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 5.0 | 85.0 |

Çizelge 4.5. SMM İle Elde Edilen Test Sonuçları

| | Test-1 | Test-2 | Cross-Subject |
|----------------------|--------|--------|---------------|
| MSRC-12 | 82.68 | 88.24 | 61.30±4.71 |
| MSR-Action 3D | 87.82 | 93.75 | 68.03±5.44 |

4.4. Sonular

Bu kesimdeki alıřmalar kapsamında srekli gzlemlerin sınıflandırılmasında kullanılan Saklı Markov modelleri ile hareket sınıflama yapılmıř ve belirli bir başarı düzeyine eriřilmiřtir. Test-1 yntemiyle yapılan iřlemlerde eđitim kmesindeki rnek sayısının az olması burdaki başarıyı Test-2 ye gre daha dřk kılmaktadır. Cross-Subject Test iin sonuların dřklđ ise bu test sırasında farklı kiřilerin aynı hareketleri gerekleřtirmesindeki farklılıklardan kaynaklanmıřtır. Sonular incelendiđinde başarı oranı Kesim 3'deki ynteme gre dřk ıkmıřtır. zellikle, cross-subject testlerinde dřř daha fazla olmuřtur. SMM modeli, kiřiler arasındaki farklılıkları ele almada daha başarısız olmuřtur. Yine SMM ile elde edilen sonularda, MSRC-12 veri kmesinde MSR-Action 3D veri kmesine gre daha başarısız sonular elde edilmiřtir. Bunun sebebi bu veri kmesindeki hareketlerin ayakta durarak yapılan hareketler olması sebebiyle birbirine daha benzer olması ve kmeleme iřlemi sonucunda birbirine yakın gzlem dizileri retmesidir. Saklı Markov Modellerinin geliřtirilmesinden sonra, zamansal rntlerin tanınmasında diđer bir yaklařım olan zaman serileri kullanılarak hareket tanıma yapılmaya alıřılmıřtır. zneliklerin zaman eksenindeki deđiřimlerinden faydalanılarak tanıma amalanmıřtır.

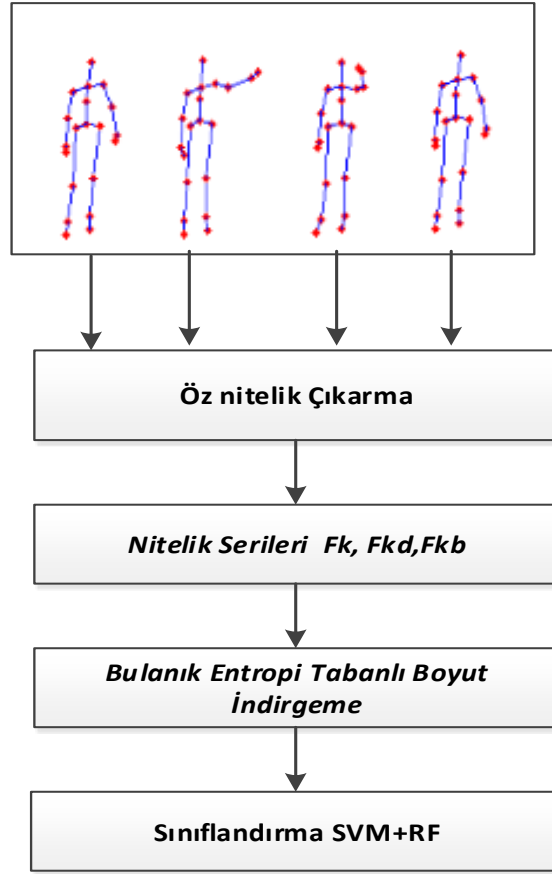
5. Zaman Serilerinden Öznitelik Çıkarımı İle Hareket Tanıma

Bu kesimde, derinlik bilgisi kullanarak insan hareketlerini seri olarak ele alan ve bu seriler üzerinden boyut indirgeme ile tanımaya çalışan bir yaklaşım sunulmaktadır. Derinlik haritaları, Shotton ve diğerleri [7] tarafından önerilen yöntemle işlenmiş ve eklem iskeleti elde edilmiştir. Eklem iskeletinden elde edilen öznitelikler zaman serileri olarak ele alınmış ve boyut indirgemeyle her hareket için sabit uzunlukta öznitelik vektörleri oluşturulmuştur. Öznitelikler eklem açıları, eklem noktalarının görece konumları, bu görece konumların değişimi ve görece eklem konularının başlangıç konumuna göre farklıdır. Elde edilen öznitelikler kullanılarak SVM ve Random Forest (RF) modelleri eğitilmiş ve sınıflama için kullanılmıştır

Hareket dizisinden elde edilen özniteliklerden sabit boyutlu bir öznitelik kümesi çıkarmada Khusbaba ve diğerleri [82] tarafından geliştirilen Wavelet Packet Transform (WPT) tabanlı yöntem kullanılmıştır. Kullanılan boyut indirgeme yöntemine ilişkin aşamalar ilerleyen kesimde açıklanmıştır. Bu yönteme ilişkin alt adımlar ve akış şeması ise Şekil 5.1.'de belirtilmiştir. Eklem serileri üzerinden önce öznitelikler çıkarılmış daha sonra öznitelik serileri üzerinden boyut indirgeme ile farklı uzunluktaki serilerden eşit uzunlukta öznitelik vektörleri elde edilmiştir. Son olarak da sınıflayıcılar eğitilerek farklı test kümeleri üzerinde test edilmiştir.

5.1. Zaman Serisinden Öznitelik Çıkarımı

Yapılan çalışmalar kapsamında hareket tanıma için çıkarılan öznitelikler farklı uzunluklardaki zaman serilerinden oluşmaktadır. Hareket dizisinden elde edilen özniteliklerden sabit boyutlu bir öznitelik kümesi çıkarmada bulanık entropi tabanlı bir yöntem kullanılmıştır [82]. Bu yöntem içerisinde Wavelet Packet Transform, Bulanık c-means ve PCA [83] alt adımlarını içermektedir.



Şekil 5.1. Yöntemin Akış Çizelgesi

5.1.1. Wavelet Packet Transform

Wavelet Packet Transform bir V_j sinyal uzayının daha düşük çözünürlüklü bir V_{j+1} uzayı ve bir W_{j+1} detay uzayına ayrıştırılmasıdır. Bu ayrıştırma işlemi V_{j+1} dikey tabanı olan $\{\phi_j(t - 2^j n)\}$ 'nin 2 yeni dikey temele ayrıştırılmasıdır. Bu iki yeni dikey temel W_{j+1} ve V_{j+1} 'in dikey temelleridir. Bu iki yeni dikey temelde yer alan ϕ_j fonksiyonları ölçek ve wavelet fonksiyonlarıdır. Bu işlem bir filtre bankası kullanılarak gerçekleştirilmekte ve sinyal 2 ayrı alt banda ayrılmaktadır. Öznitelikler genellikle wavelet sabitlerine ilişkin enerji değerlerinin hesaplanmasıyla bulunmaktadır. Bu işlem durağan olmayan sinyallerin analizinde kullanımı daha dikkat çekici ve kullanışlıdır. Fakat bu yöntemin uygulanması sonucu öznitelik uzayında büyük ölçekli bir boyutluluk oluşmaktadır. Sınıflandırma işlemi sırasında öznitelik boyutunun fazla olması sınıflayıcının öğrenme ve test aşamalarında yoğun işlem yapmasına neden

olmaktadır. Bu nedenle öznitelik uzayı üzerinde boyut indirgemesi yapılması gerekmektedir. Bu amaçla 2 aşamalı bir boyut indirme yöntemi izlenmiştir. İlk olarak fuzzy entropy yöntemi daha sonra ise PCA uygulanarak boyut indirgemesi yapılmıştır.

5.1.2 Bulanık Entropi Tabanlı Boyut İndirgeme

Bulanık entropi bir değişken üzerindeki rastgeleliğin ölçümünü veren bir yöntemdir. Bu kavrama ilişkin pek çok tanım şimdiye kadar verilmiştir [84,85]. Genel olarak entropy olasılıkların histogramından elde edilen bir kavramdır. X , M farklı durumu $[x_1, x_2, x_3 \dots x_m]$ olan bir kesikli değişken olsun. Durumlarda bulunma olasılıkları da $[p_1, p_2, p_3 \dots p_m]$ olarak ifade edilirse X 'in entropisi Eşitlik-5.1 ile hesaplanabilir.

$$H(X) = \sum_{i=1}^m P(x_i) \log P(x_i) \quad (5.1)$$

Entropi değerlerinin hesaplanmasından sonra bulanık c-means (FCM) tabanlı bir yöntem [86] kullanılarak ilk boyut indirgeme işlemi yapılmıştır. Kümeleme işlemi örnekler üzerinde değil öznitelikler üzerinde yapılmıştır.

Bulanık c-means algoritması bulanık kümeleme tekniklerinin en iyi bilineni ve en yaygın kullanılanıdır. Bulanık c-means algoritması 1973 yılında Dunn tarafından ortaya atılmış ve 1981'de Bezdek tarafından geliştirilmiştir [86]. Bu yöntemde veri kümesinde yer alan her nesne belirli üyelik dereceleriyle bir kümeye atanır. Bulanık c-means algoritmasında nesnelerin iki veya daha fazla kümeye üye olmalarına izin verilir. Bulanık mantık prensipleri uyarınca her nesne belirli bir kümeye $[0,1]$ arasında yer alan bir üyelik derecesiyle üyedir. Bir nesnenin tüm kümelere olan üyeliklerinin toplamı ise 1 olmalıdır ve her nesne kendisine en yakın olan kümeye en büyük üyelik derecesiyle bağlıdır.

Bulanık c-means algoritması Eşitlik 5.2'de belirtilen amaç fonksiyonu iterasyonlar ile minimize etmek için çalışır. Bu amaç fonksiyon aynı zamanda en küçük kareler yönteminin de bir genellemesidir.

$$J_m = \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|x_i - c_j\|^2 \quad (5.2)$$

Burada u_{ij} , x_i nesnesinin c_j merkezli kümeye olan üyelik derecesini göstermektedir. Küme sayısı C , nesne sayısı N , m ise oluşan kümelerde bulanıklığın derecesini belirtmektedir.

Algoritma, küme sayısının belirlenmesi ve u üyelik matrisinin rastgele atanmasıyla başlatılır. Yeni küme merkezleri Eşitlik 5.3 ile hesaplanır. Eşitlikten de anlaşılacağı üzere küme merkezleri içerdikleri nesnelere üyelik derecesi ağırlıklı ortalamasıyla bulunur.

$$C_j = \frac{\sum_{i=1}^N u_{ij}^m x_i}{\sum_{i=1}^N u_{ij}^m} \quad (5.3)$$

Yeni küme merkezlerinin hesaplanmasının ardından u üyelik matrisi Eşitlik 5.4 kullanılarak tekrar hesaplanır.

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left\{ \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right\}^{2/(m-1)}} \quad (5.4)$$

Son olarak yeni hesaplanan üyelik matrisi önceki adımda hesaplanan üyelik matrisi ile karşılaştırılır. Eğer aradaki fark belirlenen bir ε değerinden daha küçük ise algoritma sonlandırılır. Bu değer aynı zamanda amaç fonksiyonunun çıktısı olan değerdir. Bu yöntem K-means algoritmasından daha kararlı sonuçlar üretmektedir fakat K-means algoritmasına göre daha yavaş çalışan bir algoritmadır. Algoritmanın performansı küme merkezlerinin ilk belirlenmesi aşamasına bağlıdır. Uygun küme merkezlerinin belirlenmesi algoritmanın performansını ve başarımını önemli ölçüde arttırmaktadır.

Bulanık kümeleme işleminden sonra, bulanık entropi değerlerinin hesaplanması yapılmıştır. Sınıflara ilişkin bulanık entropi değerleri Eşitlik 5.5 ile hesaplanmaktadır.

$$FE_c = -D_c \log D_c \quad (5.5)$$

D_c c kümesinde yer alan örneklerin bu kümeye üyelik dereceleri toplamının tüm kümelerdeki örneklerin c kümesine üyelik dereceleri toplamına oranını belirtmektedir. Özniteliğe ilişkin bulanık entropi ise Eşitlik 5.6 ile hesaplanmaktadır.

$$FE = \sum_{c=1}^C FE_c \quad (5.6)$$

Özniteliklerin entropi değerlerinin hesaplanmasından sonra yüksek entropi değerlerine sahip öznitelikler elenmiştir. Entropi o öznitelikteki dağınıklığı temsil ettiği için düşük entropi değerine sahip öznitelikler daha fazla bilgi taşımaktadır. Son adım olarak da öznitelik kümesi üzerinde PCA [83] uygulanarak boyut indirgeme yapılmıştır. Bu yöntem sayesinde 16 seriden oluşan bir öznitelik kümesi 256 öznitelikle belirtilebilmektedir. Bu öznitelik sayısına PCA işleminde sonra ilk 16 bileşen alınarak erişilmektedir.

5.2 Öznitelik Çıkarma ve Sınıflama

Bu kesimdeki çalışmalar kapsamında zaman serisilerinden öznitelikler çıkarılıp WPT tabanlı öznitelik çıkarma yöntemi kullanılarak MSR-Action 3D ve MSRC-12 hareket kümeleri üzerinde hareket tanıma işlemleri gerçekleştirilmeye çalışılmıştır.

Öznitelik olarak eklemlere ilişkin açı ve eklem noktalarının görelî konumları, eklem noktalarının görelî konularının türevi ve eklem noktalarının ilk pozisyona göre farkları kullanılmıştır.

Eklemlere ilişkin açı değerleri doğrudan alınmayarak belirli aralık değerlerine atanarak alınmıştır. Açı değerlerinin alabileceği değerler 0-180 arasında 10 parçaya bölünerek ayrıklaştırılmıştır. Örnek olarak: 0-18 arasındaki açı değerlerin 1, 18-36 arasındaki açı değerlerine de 2 atanmıştır. Her eklem için ayrıklaştırılmış açı serileri öznitelik alt grubu olarak alınmıştır.

Diğer öznitelik grubumuz F_k ise eklem noktalarının görelî konumlarıdır. Eklem noktalarının görelî konumları seri olarak alınmıştır.

$$F_k = \{X_1, \dots, X_n\} \mid X_i = \{J_1, J_2 \dots, J_{20}\} \mid J_i = \{x_i, y_i, z_i\} \quad (5.7)$$

Doğrudan görelî konum değerlerine ek olarak konum değerlerinin değişimi F_{kd} de öznitelik olarak çıkarılmıştır. Konum değişimlerini hesaplanması için konum bilgilerini türevi alınmıştır (Eşitlik 5.8). Her 3 eksen içinde alınıp birleştirilmiştir.

$$F_{kd} = \frac{d}{dx} F_k \parallel \frac{d}{dy} F_k \parallel \frac{d}{dz} F_k \quad (5.8)$$

Son olarak ta konum bilgilerinin ilk konuma göre olan farkları alınmıştır (Eşitlik 5.9).

$$F_{kb} = \{X_i^c - X \mid X_i^c \in X; X_j^i \in X_i\} \quad (5.9)$$

Kullanılan bu öznitelikler üzerinde önceki kesimlerde belirtilen WPT tabanlı boyut indirgeme yöntemi uygulanmıştır. Boyut indirgeme yapılırken o özneliğe ait entropi tüm sınıflar için önce ayrı ayrı hesaplanmış ve tüm sınıflar için hesaplanan entropi değerleri toplanarak özneliğe ilişkin toplam entropi değeri hesaplanmıştır. Öznelik seçme işlemi bu toplam entropiye göre yapılmıştır. Bu yöntemin uygulanmasıyla her seri için 16 öznelik oluşturulmuştur. İskelet modelinde toplamda 20 nokta bulunmaktadır. Bu 20 noktaya ait konum bilgileri her 3 ekseninde toplam 60 zaman serisine karşılık gelmektedir. Yine bu noktaların başlangıç pozisyonuna göre farkları ve türevleride 60 zaman serisine yani toplamda 180 seriye karşılık gelmiştir. Her seride 16 bileşen ile ifade edildiğinden dolayı toplamda her hareket için 2880 öznelik oluşturulmuştur. Elde edilen öznelik kümesi kullanılarak önceki çalışmalarda olduğu gibi Random Forest ve SVM sınıflayıcıları eğitilmiştir.

Sınıflayıcıların eğitilmesi ve test edilmesi sırasında MSR-Action 3D ve MSRC-12 veri kümeleri kullanılmıştır. Sınıflayıcıların test edilmesinde önceki kesimlerdeki gibi Test-1, Test-2 ve Cross-Subject Test yöntemleri kullanılmıştır. Çizelge 5.1.'de geliştirilen yöntemin MSR-Action 3D üzerindeki sonuçları belirtilmiştir. Cross-Subject Test için farklı kişi kümeleri üzerinde testler yapıldığından bu testlerde elde edilen ortalama sonuçlar ve bu sonuçlara ait standart sapma değerleri belirtilmiştir. Bu çizelgede kolonlar test yöntemlerini, satırlar ise kullanılan sınıflayıcıları belirtmektedir. Test-1 ve Test-2 de elde edilen hata matrisleri ise Çizelge 5.2., 5.3., 5.4. ve 5.5.'te verilmiştir. MSRC-12 veri kümesi üzerinde elde edilen sonuçlar ise Çizelge 5.6.'da verilmiştir. Yine MSRC-12 ile elde edilen hata matrisleri Çizelge 5.7., 5.8., 5.9. ve 5.10.'da verilmiştir.

Sınıflayıcıların test edilmesinde literatürde yer alan Test-1, Test-2 ve Cross-Subject test yöntemleri kullanılmıştır. MSR-Action 3D veri setinde 10 deneğe ilişkin örnekler bulunduğundan 10'un 5'li kombinasyonlarının tamamı denenmiş ve ortalaması alınmıştır. Fakat önceki belirtilen testlerde MSRC-12 de 30 denek olması sebebiyle 30'un 15'li kombinasyonlarının sayısının fazla olması nedeniyle 2000 elemanlı bir örnekleme alınıp kullanılmıştır.

Çizelge 5.1. Zaman serileri kullanılan model için MSR-Action 3D Sınıflama Başarımı

| | Test-1 | Test-2 | Cross-Subject |
|------------|--------|--------|---------------|
| SVM | 89.39 | 93.79 | 82.42±7.81 |
| RF | 88.38 | 91.26 | 82.09±9.24 |

Test-1:

Çizelge 5.2. Test-1 için MSR-Action 3D Random Forest Hata Matrisi

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|----|----|----|------|------|----|----|----|----|----|-----|----|-----|----|----|-----|-----|----|----|----|----|
| 1 | 75 | 25 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2 | 00 | 95 | 05 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 3 | 00 | 00 | 1,00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 4 | 00 | 00 | 00 | 1,00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 5 | 00 | 00 | 00 | 10 | 70 | 10 | 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 6 | 00 | 00 | 00 | 00 | 16 | 84 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 7 | 00 | 00 | 00 | 00 | 05 | 10 | 70 | 00 | 05 | 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 8 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 90 | 05 | 00 | 05 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 9 | 00 | 00 | 00 | 00 | 00 | 00 | 30 | 15 | 55 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 05 | 95 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 12 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 13 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 15 | 00 | 00 | 85 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 14 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 90 | 10 | 00 | 00 | 00 | 00 | 00 |
| 15 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 100 | 00 | 00 | 00 | 00 | 00 |
| 16 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 100 | 00 | 00 | 00 | 00 |
| 17 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 05 | 90 | 00 | 05 | 00 |
| 18 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 10 | 00 | 85 | 00 | 05 |
| 19 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 05 | 90 | 05 |
| 20 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 05 | 00 | 95 |

Hata matrisi incelendiğinde 7 ve 9 numaralı hareketler arasında yanlış sınıflamanın daha fazla olduğu görülmektedir. Bunu sebebi 2 hareketinde sağ el ve kol kullanılarak yapılması ve hareket gerçekleştirilirken bu uzuvların yakın biribirine görelî yakın bölgelerde gezdirilmesidir. Aynı durum Çizelge 5.3. içinde geçerlidir. SVM sınıflayıcı ilede benzer durum yaşanmıştır.

Çizelge 5.3. Test-1 için MSR-Action 3D SVM Hata Matrisi

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|----|----|------|----|----|----|------|-----|
| 1 | 95 | 05 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2 | 05 | 90 | 05 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 3 | 00 | 00 | 95 | 05 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 4 | 00 | 00 | 00 | 90 | 05 | 05 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 5 | 00 | 00 | 00 | 05 | 70 | 10 | 15 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 6 | 00 | 00 | 00 | 00 | 16 | 84 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 7 | 00 | 00 | 00 | 05 | 05 | 00 | 55 | 00 | 35 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 8 | 00 | 00 | 00 | 00 | 00 | 00 | 05 | 90 | 05 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 9 | 00 | 00 | 00 | 00 | 00 | 00 | 20 | 00 | 80 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 12 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 13 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 15 | 75 | 00 | 05 | 05 | 00 | 00 | 00 | 00 |
| 14 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 95 | 05 | 00 | 00 | 00 | 00 | 00 |
| 15 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 1,00 | 00 | 00 | 00 | 00 | 00 |
| 16 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 05 | 95 | 00 | 00 | 00 | 00 |
| 17 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 10 | 90 | 00 | 00 | 00 |
| 18 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 05 | 00 | 85 | 05 | 05 |
| 19 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 1,00 | 00 |
| 20 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 100 |

Çizelge 5.4. Test-2 için MSR-Action 3D Random Forest Hata Matrisi

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|----|----|----|----|----|----|----|----|----|----|-----|----|-----|----|----|----|----|-----|----|----|----|
| 1 | 91 | 02 | 00 | 01 | 00 | 00 | 04 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2 | 08 | 94 | 03 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 3 | 00 | 04 | 97 | 14 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 4 | 00 | 00 | 00 | 82 | 06 | 00 | 04 | 00 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 5 | 00 | 00 | 00 | 00 | 86 | 03 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 6 | 00 | 00 | 00 | 03 | 07 | 97 | 03 | 00 | 01 | 00 | 02 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 7 | 00 | 00 | 00 | 00 | 01 | 00 | 64 | 03 | 05 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 8 | 00 | 00 | 00 | 00 | 00 | 00 | 05 | 92 | 17 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 9 | 00 | 00 | 00 | 00 | 00 | 00 | 14 | 02 | 75 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 04 | 00 | 01 | 100 | 01 | 00 | 07 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 03 | 00 | 00 | 97 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 12 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 13 | 01 | 00 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 00 | 84 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 14 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 05 | 95 | 00 | 00 | 00 | 01 | 00 | 00 |
| 15 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 01 | 02 | 97 | 02 | 00 | 00 | 01 | 00 |
| 16 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 01 | 03 | 03 | 98 | 00 | 00 | 00 | 00 |
| 17 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 100 | 04 | 00 | 00 |
| 18 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 93 | 05 | 04 |
| 19 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 02 | 90 | 01 |
| 20 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 00 | 04 | 95 |

Çizelge 5.5. Test-2 için MSR-Action 3D SVM Hata Matrisi

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|----|----|----|----|----|----|----|----|
| 1 | 98 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 2 | 02 | 95 | 03 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 3 | 00 | 05 | 97 | 03 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 4 | 00 | 00 | 00 | 92 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 5 | 00 | 00 | 00 | 02 | 91 | 07 | 02 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 6 | 00 | 00 | 00 | 03 | 04 | 93 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 7 | 00 | 00 | 00 | 00 | 04 | 00 | 74 | 05 | 12 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 8 | 00 | 00 | 00 | 00 | 00 | 00 | 08 | 95 | 05 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 9 | 00 | 00 | 00 | 00 | 00 | 00 | 10 | 00 | 83 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 12 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 100 | 02 | 00 | 00 | 00 | 00 | 01 | 00 | 00 |
| 13 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 87 | 00 | 00 | 00 | 00 | 00 | 00 | 02 |
| 14 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 95 | 00 | 01 | 00 | 00 | 00 | 00 |
| 15 | 00 | 00 | 00 | 00 | 00 | 00 | 03 | 00 | 00 | 00 | 00 | 00 | 03 | 00 | 97 | 04 | 00 | 00 | 00 | 03 |
| 16 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 05 | 04 | 03 | 95 | 00 | 00 | 00 | 00 |
| 17 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 99 | 05 | 00 | 00 |
| 18 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 01 | 94 | 00 | 03 |
| 19 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 99 | 00 |
| 20 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 03 | 01 | 00 | 00 | 00 | 00 | 01 | 92 |

Test-1 de Çizelge 5.3. ve 5.4. üzerinde gözlemlenen durum Test-2 de azda olsa görülmektedir. Düşüşün sebebi Test-2 de daha geniş bir eğitim kümesi ve daha dar bir test kümesi kullanılmasıdır.

Çizelge.5.6. MSRC-12 Sonuçları

| | Test-1 | Test-2 | Cross-Subject |
|-----|--------|--------|---------------|
| SVM | 86.26 | 93.34 | 68.11±6.70 |
| RF | 96.02 | 97.22 | 79.34±10.07 |

Test-1

Çizelge 5.7. Test-1 için MSRC-12 SVM Hata Matrisi

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|----|----|-----|----|----|----|----|----|----|----|----|----|
| 1 | 97 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 |
| 2 | 01 | 95 | 03 | 00 | 00 | 00 | 00 | 00 | 01 | 00 | 00 | 00 |
| 3 | 00 | 00 | 100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 4 | 00 | 00 | 03 | 84 | 13 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 5 | 00 | 00 | 00 | 03 | 97 | 01 | 00 | 00 | 00 | 00 | 00 | 00 |
| 6 | 00 | 00 | 00 | 00 | 03 | 92 | 05 | 00 | 00 | 00 | 00 | 00 |
| 7 | 00 | 00 | 00 | 00 | 00 | 05 | 94 | 01 | 00 | 00 | 00 | 00 |
| 8 | 00 | 00 | 00 | 00 | 00 | 00 | 03 | 95 | 02 | 00 | 00 | 00 |
| 9 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 08 | 90 | 02 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 03 | 94 | 03 | 00 |
| 11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 03 | 94 | 03 |
| 12 | 05 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 05 | 90 |

Çizelge 5.8. Test-1 için MSRC-12 Random Forest Hata Matrisi

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|----|----|----|----|-----|----|----|----|----|----|----|----|
| 1 | 97 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 03 |
| 2 | 02 | 93 | 01 | 00 | 00 | 00 | 01 | 01 | 02 | 00 | 00 | 00 |
| 3 | 00 | 05 | 95 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 4 | 00 | 00 | 01 | 98 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 5 | 00 | 00 | 00 | 00 | 100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 6 | 00 | 00 | 00 | 00 | 03 | 93 | 05 | 00 | 00 | 00 | 00 | 00 |
| 7 | 00 | 00 | 00 | 00 | 00 | 03 | 95 | 01 | 01 | 00 | 00 | 00 |
| 8 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 97 | 00 | 00 | 00 | 00 |
| 9 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 06 | 93 | 01 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 04 | 96 | 00 | 00 |
| 11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 97 | 02 |
| 12 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 00 | 00 | 01 | 96 |

Test-2

Çizelge 5.9. Test-2 için MSRC-12 SVM Hata Matrisi

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 98 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 01 |
| 2 | 03 | 94 | 02 | 00 | 00 | 01 | 00 | 00 | 00 | 00 | 00 | 01 |
| 3 | 00 | 03 | 96 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 4 | 00 | 00 | 02 | 91 | 07 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 5 | 00 | 00 | 00 | 03 | 96 | 01 | 00 | 00 | 00 | 00 | 00 | 00 |
| 6 | 00 | 00 | 00 | 02 | 03 | 85 | 09 | 01 | 00 | 00 | 00 | 00 |
| 7 | 00 | 00 | 00 | 00 | 00 | 09 | 89 | 02 | 00 | 00 | 00 | 00 |
| 8 | 00 | 00 | 00 | 00 | 00 | 00 | 03 | 94 | 03 | 00 | 00 | 00 |
| 9 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 05 | 90 | 04 | 00 | 01 |
| 10 | 00 | 00 | 02 | 00 | 00 | 00 | 00 | 00 | 05 | 89 | 04 | 00 |
| 11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 04 | 96 | 01 |
| 12 | 04 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 06 | 90 |

Çizelge 5.10. Test-2 için MSRC-12 Random Forest Hata Matrisi

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----|----|----|----|----|-----|----|----|----|----|----|----|----|
| 1 | 99 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 01 |
| 2 | 04 | 96 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 3 | 00 | 03 | 96 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 4 | 00 | 00 | 01 | 98 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 5 | 00 | 00 | 00 | 00 | 100 | 00 | 00 | 00 | 00 | 00 | 00 | 00 |
| 6 | 00 | 00 | 00 | 00 | 03 | 97 | 00 | 00 | 00 | 00 | 00 | 00 |
| 7 | 00 | 00 | 00 | 00 | 00 | 01 | 99 | 00 | 00 | 00 | 00 | 00 |
| 8 | 00 | 00 | 00 | 00 | 00 | 01 | 00 | 98 | 01 | 00 | 00 | 00 |
| 9 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 04 | 95 | 02 | 00 | 00 |
| 10 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 04 | 96 | 01 | 00 |
| 11 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 98 | 00 |
| 12 | 01 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 00 | 02 | 97 |

Geliştirilen yöntemin diğer yöntemlerle karşılaştırılması Çizelge 5.11.'de verilmiştir. Çizelgede yöntemlerin MSR-Action 3D ile test edilmesi sonucunda elde edilen sonuçlar gösterilmektedir. Kesim 3.6'da olduğu gibi '*' işareti bulunmayanlar tüm kombinasyonları kullanan çalışmalardır. Karşılaştırmada daha yüksek sonuç veren SVM modeli kullanılmıştır.

Çizelge 5.11. Geliştirilen yöntemin MSR-Action 3D veri kümesi için diğer yöntemlerle karşılaştırılması

| | Test-1 | Test-2 | Cross-Subject Test |
|------------------------|--------|--------|--------------------|
| Bag of 3D Points [3] | 91.6 | 94.2 | 74.7* |
| HOJ3D [78] | 96.2 | 97.2 | 79.0* |
| EigenJoints [68] | 95.8 | 97.8 | 83.3* |
| Yang ve diğ. [64] | N/A | N/A | 85.5* |
| Wang ve diğ. [70] | N/A | N/A | 88.2* |
| Xia ve diğ. [65] | 93.5 | 96.7 | 89.3* |
| HON4D [66] | N/A | N/A | 89.15±4.18 |
| Önerilen Yöntem | 88.3 | 91.2 | 82.42±7.81 |

5.3. Sonuçlar

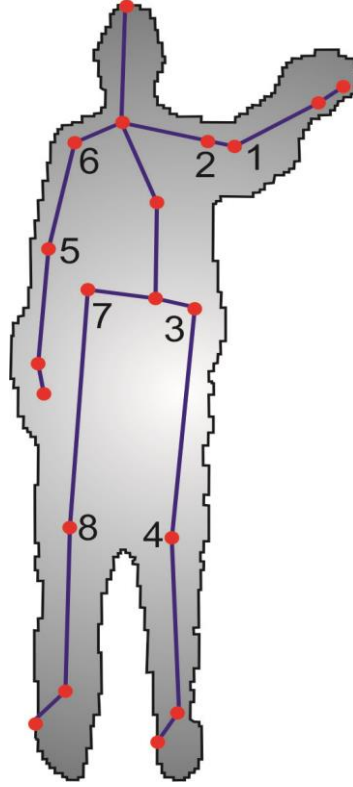
Sonuç olarak, zaman serileri üzerinden öznitelik çıkarılarak oluşturulan modeller, MSRC-12 ve MSR-Action 3D veri kümeleri ile test edilmiş ve literatürde yer alan çalışmalara yakın karşılaştırılabilir sonuçlar elde edilmiştir. Geliştirilen yöntem, Kesim 3'teki yöntem nazaran daha az başarılı olsa da, Kesim 7'deki sonuçlara göre bu yöntem hareketin tümünü görmeden sonuç üretebilmede daha başarılı olmaktadır. Bu nedenle zaman serilerinin, düşük gecikmeli hareket tanıma da yararlı olabileceği için çalışılması anlamlıdır.

Bir sonraki kesimde ise iskelet modeli haricinde ham derinlik bilgisi kullanılarak da hareketler tanınmaya çalışılmıştır. İskelet modelinden elde edilen nitelikler ham derinlik bilgisinden elde edilen nitelikler ile birleştirilip daha ayırt edici ve başarımı daha yüksek modeller geliştirilmesi amaçlanmıştır.

6. Ham Derinlik Bilgisinden Faydalanılarak Çok Modlu Hareket Tanıma

Bu kesimde, iskelet modelinden elde edilen özniteliklerden sonra, ham derinlik bilgisinden elde edilen öznitelikleri kullanarak hareket tanıma başarımı artırılmaya çalışılmıştır. Bu kapsamda denenen yaklaşımlar, video tabanlı yöntemlerde özellikle de silüet tabanlı çalışmalarda kullanılan yöntemlerdir. Arka plan ve insan bedeni arasındaki belirgin ayırım sayesinde, derinlik haritalarından silüet çıkarma işlemi daha kolaylıkla yapılabilmektedir. Buradan hareket ile, video ve görüntü işlemede sık sık başvurulan koordinat sistemi değiştirme, ilgi noktaları (*interest points*) çıkarma ve Histogram of Oriented Gradients (HOG) yaklaşımları derinlik haritaları üzerinde de uygulanabilmektedir. Ham derinlik bilgisinden çıkarılan öznitelikler, daha önceki çalışmalarda elde edilen öznitelikler ile birleştirilerek Cross-Subject Test başarımı artırılması mümkündür.

Bu kesimde, derinlik bilgisi kullanarak tanıma kapsamında, Shotton ve diğerleri [7] tarafından geliştirilen modelden (Şekil 6.1.) elde edilen öznitelikler yanında ham derinlik verisinden de öznitelikler çıkartılmaya çalışılmıştır. Bu kapsamda, polar koordinat sistemleri, SURF öznitelikleri, Derinlik Geçmiş Haritası (DGH) ve Derinlik Gradyan Görüntüsü (GG) üzerine yöntemler çalışılmıştır. Çıkarılan yeni öznitelikler önceki çalışmalardan elde edilen öznitelikler ile birleştirilerek mevcut başarımlar artırılmaya çalışılmıştır. Geliştirilen yöntemlerin test edilmesinde MSR-Action 3D veri kümesi kullanılmıştır. Bu veri kümesinde derinlik ve iskelet bilgileri senkron bir şekilde yer almaktadır. Bu sayede iskelet modelinden çıkarılan öznitelikler yanında, derinlik haritasından çıkarılan öznitelikler birlikte kullanılabilmiştir. Daha önceki aşamalarda kullanılan MSRC-12 veri kümesinde senkron olarak derinlik ve eklem iskeleti verileri bulunmadığı için, bu kesimdeki çalışmalarda bu veri kümesi kullanılmamıştır.



Şekil 6.1. Shotton ve diğerleri [7] tarafından geliştirilen iskelet modeli ve derinlik haritası

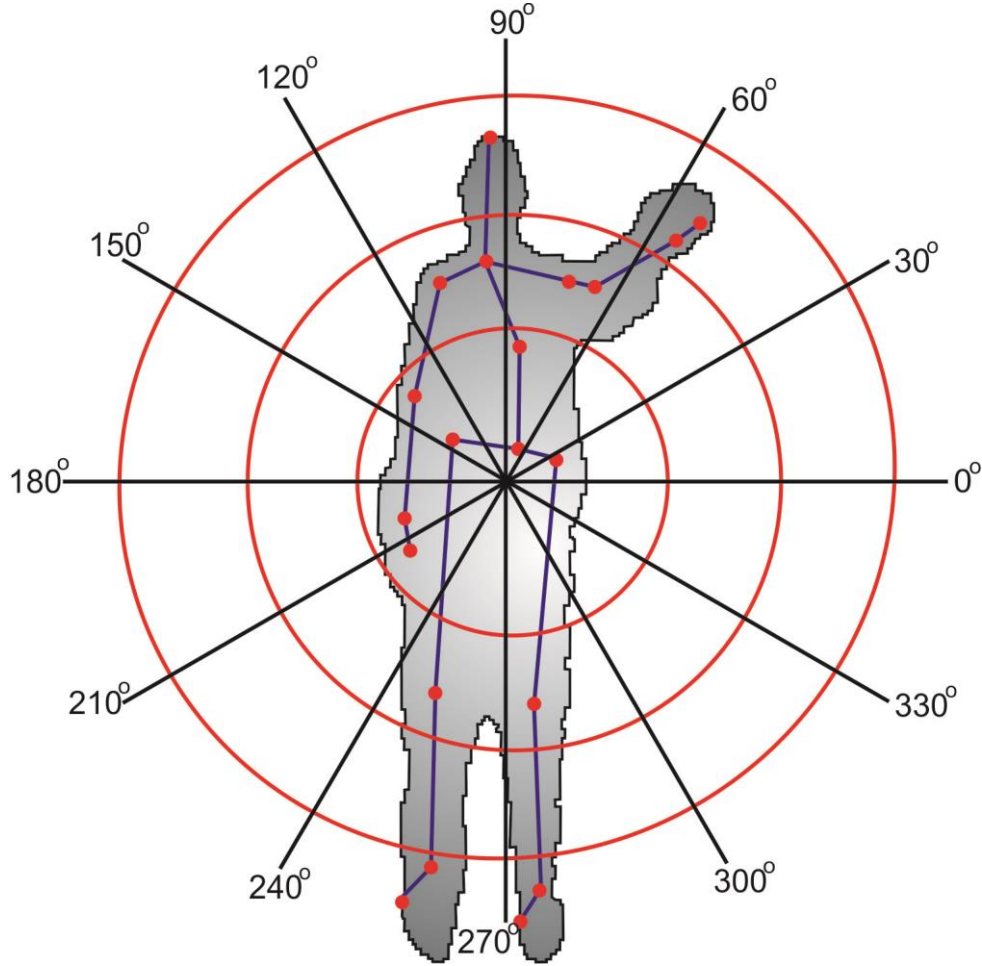
6.1. Derinlik Haritası Üzerinde Polar Koordinat Sistemi Tanımlama

Çalışmalar kapsamında, derinlik haritasındaki silüet üzerine polar bir koordinat sistemi oturtulup parçalara bölünmüş ve bu parçalarda yer alan silüete ilişkin noktaların sayılmasıyla bir histogram oluşturulmuştur. Oluşturulan histogram öznitelik olarak kullanılarak sınıflama yapılmıştır.

Polar koordinat sistemine dönüştürme Eşitlik 6.1, 6.2 kullanılarak gerçekleştirilmiştir. x_i , y_i derinlik haritasında yer alan noktanın koordinatlarını belirtmektedir. x_c , y_c ise silüet merkezini belirtmektedir. x_c , y_c koordinatları ortalaması alınarak bulunmuştur. Oluşturulan koordinat sistemi Şekil 6.2.'de gösterilmiştir. Koordinat sisteminin merkezi derinlik haritasının ağırlık merkezidir. Hesaplanan r_i ve θ_i değerleri ile bir noktanın hangi bölgeye düştüğü tespit edilmektedir. Sözü geçen bölgeler Şekil 6.2.'de kırmızı renki çemberler ve siyah doğru parçaları arasındaki bölümlerdir.

$$r_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \quad (6.1)$$

$$\theta_i = \tan^{-1} \frac{y_i - y_c}{x_i - x_c} \quad (6.2)$$



Şekil 6.2. Oluşturulan koordinat sistemi

Koordinat sistemi oluşturulduktan sonra, m tane açı ve n tane çaptan oluşan K tane hücreye bölünmüş ve her hücrede yer alan noktalar sayılarak histogramlar oluşturulmuştur. Çalışmalar sırasında m değeri 12, n değeri 3 olarak alınmıştır. Bir noktanın hangi hücreye ait olduğu Eşitlik 6.3 ve 6.4 kullanılarak hesaplanmaktadır. İlgili noktanın polar koordinat sisteminde düştüğü kesim (m,n) bulunduktan sonra ilgili kesimlere ait nokta sayıları artırılmıştır. Bu işlem tüm çerçeveler için yapılmış olup hareket sonunda hangi hücreden kaç nokta geçtiğine ilişkin histogram oluşturulmuştur. Eşitlik 6.3'te θ_i ve Eşitlik 6.4'te r_j derinlik haritasında yer alan bir noktanın polar koordinatlarını belirtmektedir.

$$\frac{2\pi i}{m} \leq \theta_i \leq \frac{2\pi(i+1)}{m} \quad i=0,1,2,\dots,m-1 \quad (6.3)$$

$$\frac{j}{n}r_{max} \leq r_j \leq \frac{1}{n}(j+1)r_{max} \quad j=0,1,2,\dots,n-1 \quad (6.4)$$

$$k = n * (m - 1) \quad (6.5)$$

Histogramların önceki kesimde çalıştığımız öznitelikler (eklem açıları ve yer değiştirme öznitelikler) ile birleştirilerek sınıflamada kullanılmasıyla elde edilen sonuçlar Çizelge 6.1'de verilmiştir. Cross-Subject Test üzerinde iyileştirici bir etki yapmamıştır. Sınıflama için daha önceki aşamalarda olduğu gibi SVM ve RF yöntemleri kullanılmıştır.

Çizelge 6.1. Polar koordinat sistemi ile MSR-Action 3D üzerinde elde edilen sonuçlar

| | Test-1 | Test-2 | Cross-Subject Test |
|-----|--------|--------|--------------------|
| SVM | 88.12 | 93.21 | 78.16±8.18 |
| RF | 92.07 | 95.07 | 88.04±7.63 |

6.2. Surf İlgili Noktaları Kullanımı

Çalışmalar kapsamında diğer bir alternatif olarak derinlik haritası üzerinde Surf öznitelikleri çıkartılarak hareketler tanınmaya çalışılmıştır. Bu amaçla ilk olarak derinlik çerçeveleri üzerinde surf noktaları bulunup bu surf noktalarına ilişkin tanımlayıcılar (*surf descriptors*) [36] hesaplanmıştır. Hesaplanan bu tanımlayıcılar K-means algoritması ile kümelenecek 80 kelimedenden oluşan bir sözlük (*codebook*) oluşturulmuştur. Sözlük oluşturulmasında sadece eğitim kümesinde yer alan derinlik haritaları kullanılmıştır. Daha sonra oluşturulan bu sözlük kullanılarak her harekette yer alan sözcüklere ilişkin bir histogram tanımlanmıştır. Eğitim kümesinde yer alan her hareket serisi için bir histogram tanımlanmıştır. Oluşturulan bu histogramlar ile önceki kesimlerde tanımladığımız iskelet modeli öznitelikleri (açı histogramları ve yer değiştirmeler) birleştirilerek SVM ve RF modelleri eğitilmiştir.

Test aşamasında, test örnekleri içinde surf tanımlayıcıları bulunup eğitim aşamasında tanımlanan kümelerden en yakın olanlarına atanmış ve histogramları

oluşturulmuştur. Fakat, testler sonucunda geliştirilen bu yöntemin başarımlar açısından bir katkı sağlamadığı ve başarımların Test-2 için %50'ler düzeyinde kaldığı gözlemlenmiştir. Test-2 için başarımların düşük olması sebebiyle Cross-Subject Test yapılmamıştır.

6.3. Yıldız İskelet Modeli

Bu kesim kapsamında kalça noktasının merkez alındığı diğer eklem noktalarının ise çeper noktaları kabul edildiği, Şekil 2.3.'tekine benzer bir model geliştirilmiştir. Bu modelde öznitelik olarak merkez noktadan eklemlere vektörler kullanılmıştır. Merkez nokta ve eklem noktaları vektörler olarak ele alınmış ve vektörlerin 3 düzlem ile yaptıkları açıların histogramları çıkarılmıştır. Sonrasında elde edilen bu öznitelikler daha önceki adımlarda olduğu gibi SVM ve RF yöntemleri kullanılarak sınıflandırılmıştır. Sonuçlar daha önce önerdiğimiz yöntemlerden daha başarılı çıkmamıştır. MSR-Action 3D kümesi üzerinde yeterince iyi sonuç vermediği için bu yöntem üzerinde fazla durulmamıştır. MSR-Action 3D kümesi üzerinde Test-2 ile %80 civarında bir başarı elde edilmiştir. Başarımların düşük kalması sebebiyle bu yaklaşım üzerine daha fazla gidilmemiştir.

6.4. Derinlik Geçmiş Haritası

İskelet modelinden elde vücut şekil ve hareket özniteliklerine ek olarak, ham derinlik görüntü serilerinden yeni öznitelikler çıkarılmıştır. Bu kapsamda her noktası hareket ve derinlik frekansının bir fonksiyonu olan Derinlik Geçmiş Haritası oluşturulmuştur. DGH, statik bir görüntü şablonudur. DGH, derinlik haritası serisinde yer alan derinlik görüntülerinin ağırlık toplamları alınarak hesaplanmaktadır. Ağırlık değerleri seride yer alan çerçevelerin indis numarasıdır. Bu sayede sonda yer alan hareketler daha açık renkler ile ifade edilmektedir. DGH hesaplamasında kullanılan yöntem Eşitlik 6.6 da gösterilmektedir. Eşitlik 6.6'da D_i i numaralı derinlik haritasını ve N serisinde yer alan toplam derinlik haritası sayısını belirtmektedir.

$$DGH = \sum_{i=1}^N (D_i * i) \quad (6.6)$$

Oluşturulan DGH üzerinden şekil ve duruşa ilişkin öznitelikleri çıkarılması amacıyla HOG betimleyicileri (*descriptors*) hesaplanmıştır. Histogram of Oriented Gradient (HOG) algoritması, Shashua ve diğerleri [87] ve Dalal ve Triggs [55] tarafından yaya

tanıma sistemlerinde kullanılmak için önerilen ve algoritma ve sınıflandırıcıların performansını arttırmak için sıkça kullanılan yöntemlerden biridir.

HOG yöntemi görüntü işleme ve bilgisayarlı görüde sıkça kullanılan bir öznitelik çıkarma yöntemidir. Yöntem görüntü üzerindeki bölgelerde gradyan oryantasyonlarının bulunmasına dayanmaktadır. Yöntem eşit aralıklarla dağıtılmış hücrelerde kontrast normalizasyonu yapılarak gradyanların bulunmasına dayanmaktadır.

İlk olarak Dalal ve Triggs [55] tarafından ortaya atılan bu yöntemde, görüntü üzerindeki kişilerin tespiti için kullanılmıştır. Yöntemin özünde yatan fikir nesnelere şekil, gradyan ve yoğunluk (*intensity*) değerlerinin dağılımının yerel olarak ifade edilebilmesidir.

Yöntemde ilk olarak gradyan hesaplamaları yapılmaktadır. Bu işlem aşağıda belirtilen tek boyutlu filtrelerin görüntü üzerinde dolaştırılması ve konvulasyona tabi tutulmasıyla yapılmaktadır. Hesaplama öncesi yapılan yumuşatma (*smoothing*) ön işlemleri daha iyi sonuçlar elde edilmesine yardımcı olmaktadır.

$$[-1,0,1] \quad [-1,0,1]^T$$

Belirtilen filtreler dışında kenar bulmada kullanılan Sobel filtreleri'de gradyan bulmada kullanılabilir.

İkinci adım olarak hücre histogramları hesaplanmaktadır. Hücreler içinde yer alan her piksel için gradyan değeri kullanılarak histogramlar oluşturulur. Histogramdaki her çubuk belirli bir gradyan büyüklük aralığını ifade etmektedir. Her piksele ait değerler histogram çubukları için bir oya tekabül etmektedir. Histogramlar oluşturulurken doğrudan gradyan büyüklüğü yerine bununla ilişkili bir fonksiyon kullanılabilir. Hücre histogramları oluşturulduktan sonra blok tanımlayıcılarının hesaplanmasına geçilmiştir.

Kontrast ve aydınlanmadaki (*illumination*) değişikliklerden kaynaklı gürültülerin azaltılması için hücreler yerel olarak gruplanarak normalize edilmektedir. HOG tanımlayıcıları bu normalize edilmiş bloklardan elde edilen hücre histogramlarının birleştirilmesiyle oluşan vektörlerdir. Bu bloklar üst üste gelen (*overlap*) ve dairesel veya dikdörtgen şekilli olabilir. Blokların kesişmesinden dolayı bir hücre farklı

bloklardan elde edilen tanımlayıcılarda yer alabilir. Blok normalizasyon işlemlerinde kullanılan belli başlı yöntemler aşağıda Eşitlik 6.7-6.10'da listelenmiştir. Eşitliklerde v bir blok içerisinde yer alan normalize edilmemiş histogramları belirtmektedir. $\|v\|_k$ $k = 1,2$ v 'nin k -norm'unu, e ise sabiti belirtmektedir.

$$L2 - norm: f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \quad (6.7)$$

$$L2hys - norm: f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \quad \max v = 0.2 \quad (6.8)$$

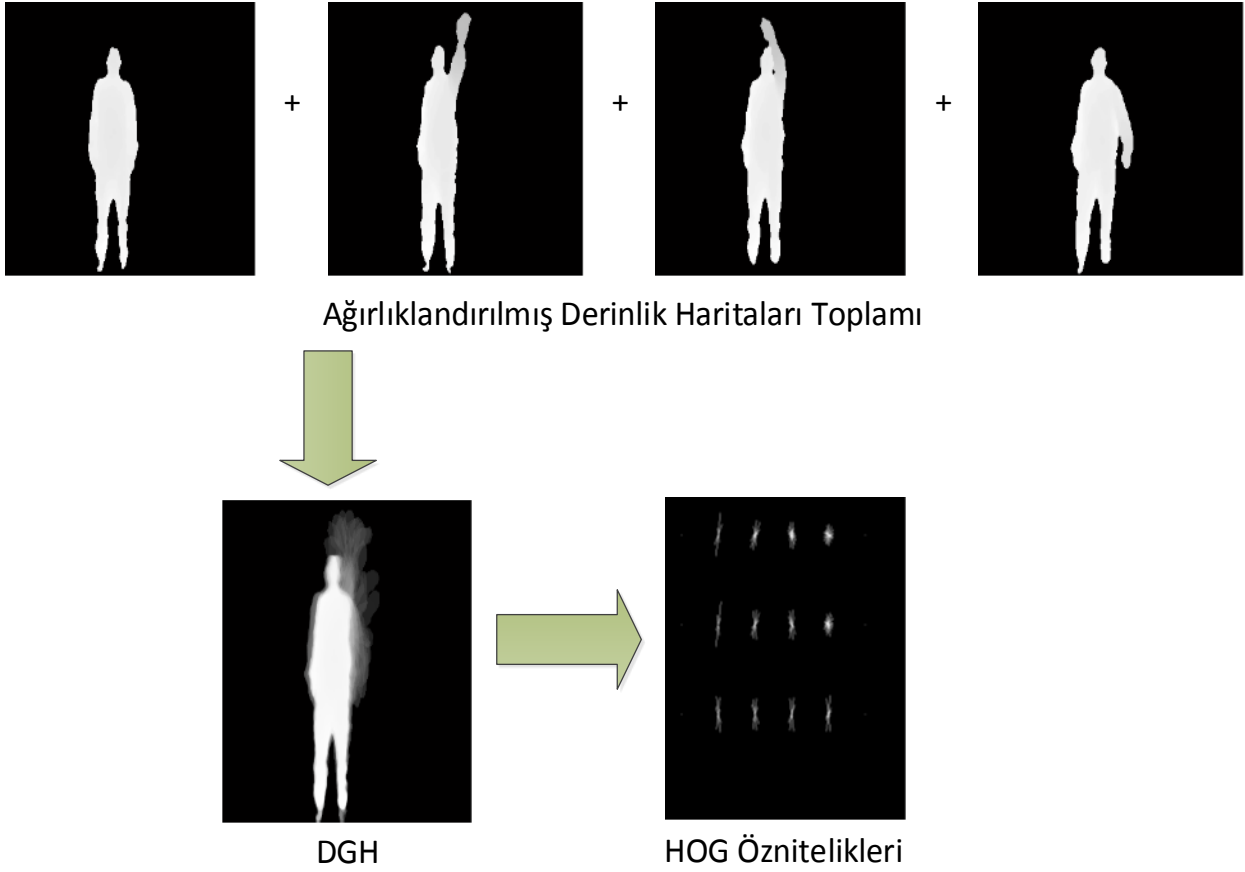
$$L1 - norm: f = \frac{v}{\|v\|_1 + e} \quad (6.9)$$

$$L1 - sqrt: f = \sqrt{\frac{v}{\|v\|_1 + e}} \quad (6.10)$$

Belirtilen bu normalizasyon yaklaşımlarından L1-norm dışındakiler yakın sonuçlar üretmektedir. L1-norm yöntemi ise diğer yöntemlerden daha düşük sonuçlar üretmektedir.

Görüntüye ait HOG tanımlayıcıları oluşturulduktan sonra, bu tanımlayıcılar öz nitelik vektörleri olarak kullanılarak sınıflama yapılabilmektedir.

HOG öznitelikleri hesaplaması 64x64 lük kare şeklinde bloklar, 32x32 lik üst üste gelmeyen hücreler ve 8 gradyan yönelemi kullanılarak yapılmıştır. HOG öznitelikleri genelde kontrast ve aydınlanmadaki değişiklikleri belirtmekte kullanılır. Fakat bu durumda derinlik üzerindeki değişikliklerin belirtilmesinde kullanılmaktadır. HOG özniteliklerinin çıkartılmasında ilk adım gradyanların hesaplanmasıdır. Gradyanlar filtreler yardımıyla dikey ve yatay olarak hesaplanır. Sonra gradyanlara ait yönlerin histogram haline getirilmesi hücreler düzeyinde yapılmaktadır. Bir sonraki adımda blok betimleyicileri hücre histogramlarının bir araya getirilmesiyle elde edilir. Son olarak da HOG betimleyicileri normalleştirilmiş hücre histogramlarının bir vektörüdür. Örnek bir DGH hesaplanmasına ilişkin adımlar Şekil 6.3.'te belirtilmektedir. DGH öznitelikleri ile yapılan deney sonuçları karşılaştırma amacıyla kesim 6.5.'de verilmiştir.



Şekil 6.3. El sallama hareketine ilişkin DGH hesaplaması

6.5. Derinlik Gradyan Görüntüsü

Derinlik haritasından öznitelik çıkarma kapsamında geliştirilen bir diğer yöntem ise gradyan görüntüsüdür. Bu yöntemde ilk olarak derinlik görüntüsünden 3 boyutlu bir hacim oluşturulmuştur. Oluşturulan hacim üzerinde 3 boyut (x,y,z) içinde Eşitlik-7 kullanılarak gradyan hesaplaması yapılmıştır. Eşitlik 6.11'de D oluşturulan 3 boyutlu hacmi, N derinlik haritası sayısını (çerçeve sayısı) belirtmektedir. 3 eksen içinde gradyanların hesaplanmasıyla 3 eksenindeki değişimi ifade eden bileşenler bulunmuştur. Bu bileşenler Eşitlik 6.12, 6.13, 6.14'te belirtildiği şekilde kümülatif olarak toplanmıştır. Son olarak elde edilen bu toplamlar toplanarak gradyan görüntüsü (GG) oluşturulmuştur. El sallama hareketi için GG oluşturulmasında izlenen adımlar Şekil 6.4. ile gösterilmiştir.

$$[F_x, F_y, F_z] = \nabla(D) = \frac{\partial F}{\partial x} \hat{i} + \frac{\partial F}{\partial y} \hat{j} + \frac{\partial F}{\partial z} \hat{k} \quad (6.11)$$

$$F_{x_{total}} = \sum_{i=1}^N F_{x_i} \quad (6.12)$$

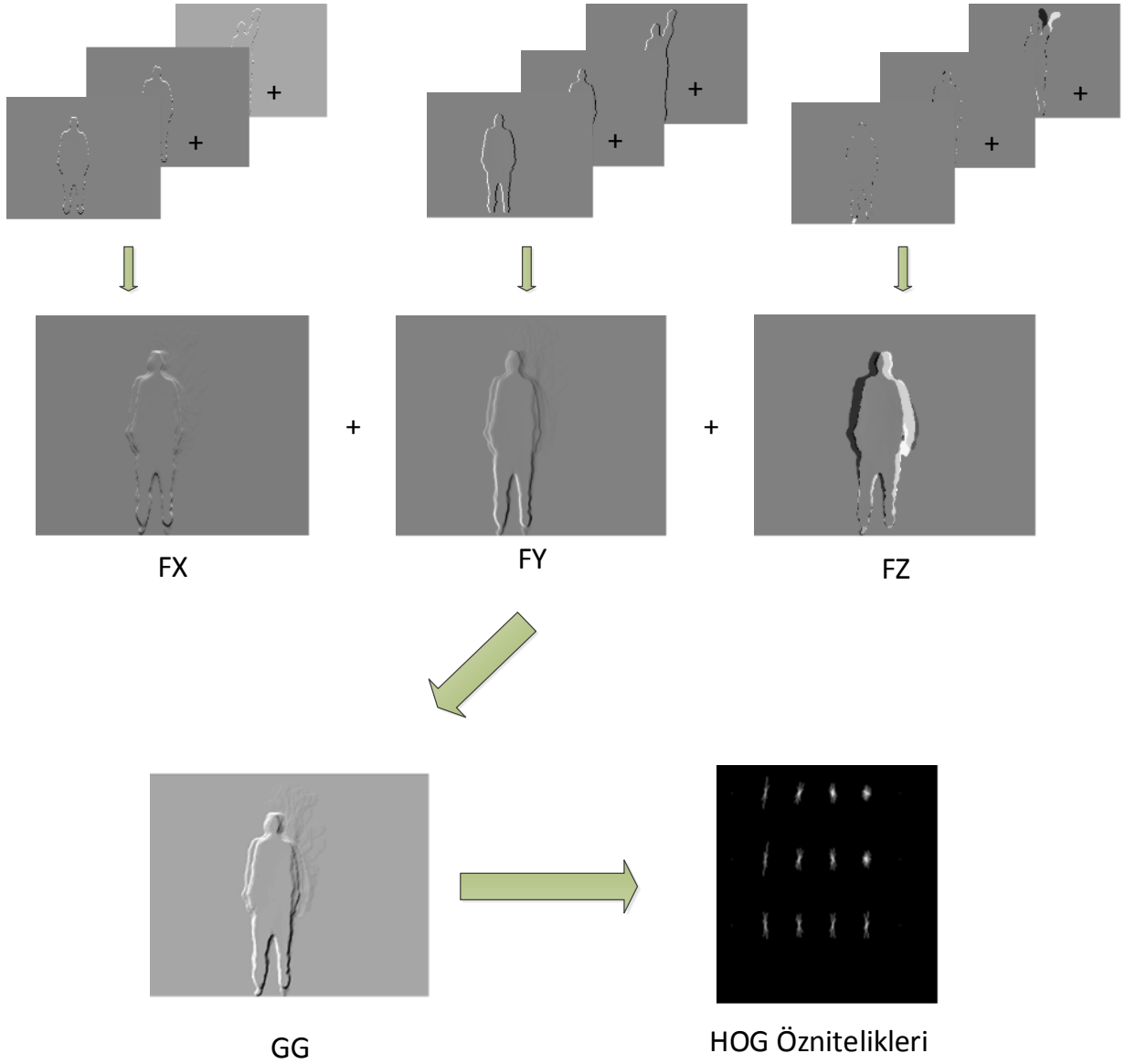
$$F_{y_{total}} = \sum_{i=1}^N F_{y_i} \quad (6.13)$$

$$F_{z_{total}} = \sum_{i=1}^N F_{z_i} \quad (6.14)$$

$$GG = F_{x_{total}} + F_{y_{total}} + F_{z_{total}} \quad (6.15)$$

GG elde dilmesinden sonra, GG üzerinde HOG öznitelikleri çıkarılıp, iskelet modelinden elde edilen öznitelikler ile birleştirilmiştir. DGH ile aynı deney düzeneği kullanılarak HOG öznitelikleri çıkarılmıştır.

Her ekseninde yer alan toplam gradyanlardan ayrı ayrı HOG öznitelikleri kullanılarak da hareketler sınıflanmaya çalışılmış fakat sınıflama sonucuna herhangi bir etkisi olmamıştır. Bu 3 toplamın toplamları alınarak elde edilen görüntü üzerinden hareketle mevcut sınıflama başarımında %1 oranında bir artış sağlanmıştır.



Şekil 6.4. El sallama hareketine ilişkin GG hesaplaması

6.6. Sınıflama Deneyleri

Ham derinlik bilgisinden de faydalanılarak geliştirilen yöntemlerin , literatürde yer alan diğer yöntemlerle ve güncel diğer yöntemler ile karşılaştırılmalı tablosu Çizelge 6.2. de gösterilmektedir. Çizelge 6.2.'de yöntemlere ilişkin daha önceki testler sırasında da kullanılan Test-1, Test-2 ve Cross-Subject Test yöntemleri yer almaktadır. Sınıflandırıcı olarak Random Forest Yöntemi ve veri kümesi olarak da MSR-Action 3D kullanılmıştır. Sınıflama işlemi ham derinlik bilgisinden elde edilen nitelikler ile önceki kesimlerde kullanılan eklem açısı histogramları ve eklem yer değiştirme bilgisi birleştirilerek elde edilmiştir. GG ve DGH üzerinden elde edilen HOG öz nitelikleri $1 \times N$

uzunluğunda vektörlerdir. Diğer grup öznelikler ile peş peşe eklenerek birleştirilmiştir. Sınıflayıcı parametreleri önceki kesimlerle aynıdır. Geliştirilen yöntemlerden başarıma katkı sağlamış olanlar tabloda yer almaktadır. Ham derinlik bilgisi kullanılmasıyla başarıma önceki adımlara göre %1.2 oranında artmıştır. Yanında * bulunan çalışmalarda, deneklerin bir yarısı seçilip eğitim yapıldıktan sonra diğer yarısı ile test edilmiş, fakat bütün denek kombinasyonları denenmemiştir. Yani sadece bir alt küme kombinasyonu ile Cross-Subject Test yapılmıştır. Farklı alt kümeler, başarıma değiştirebileceği için, bu çalışmada bütün alt küme kombinasyonlarını denenmiştir. Kesim 3.6'da olduğu gibi '*' işareti bulunmayanlar tüm kombinasyonları kullanan çalışmalardır. Bütün kombinasyonlar denendikten sonra, elde edilen sınıflama başarıma sonuçlarının ortalama ve standart sapması hesaplanmış ve tabloda verilmiştir. Bu çalışmaya benzer bir Cross-Subject Test yöntemini HON4D [66] çalışması uygulamıştır. Çizelge 6.3.'te ise SVM sınıflayıcısı ile elde edilen sonuçlar belirtilmiştir. Çizelgelerden anlaşılacağı gibi RF sınıflayıcısı daha başarılı sonuçlar üretmiştir.

Çizelge 6.2. Geliştirilen Yöntemlerin Literatürde yöntemlerle MSR-Action 3D üzerinde RF sınıflayıcısı ile kıyaslaması.

| | Test-1 | Test-2 | Cross-Subject Test |
|--------------------------|--------|--------|--------------------|
| Bag of 3D Points [3] | 91.6 | 94.2 | 74.7* |
| HOJ3D [78] | 96.2 | 97.2 | 79.0* |
| EigenJoints [68] | 95.8 | 97.8 | 83.3* |
| Yang ve diğ. [64] | N/A | N/A | 85.5* |
| Wang ve diğ. [70] | N/A | N/A | 88.2* |
| Xia ve diğ. [65] | 93.5 | 96.7 | 89.3* |
| HON4D [66] | N/A | N/A | 82.15±4.18 |
| Açı+Yer değiştirme | 95.3 | 97.7 | 88.2±7.39 |
| Açı + Yer değiştirme +GG | 93.1 | 97.7 | 89.0±7.06 |
| Açı+Yer değiştirme +DGH | 95.4 | 98.0 | 89.4±6.86 |

Çizelge 6.3. Ham Derinlik Bilgisi İle Elde Edilen Test Sonuçları, SVM için

| | Test-1 | Test-2 | Cross-Subject Test (Ort.) |
|--------------------------|--------|--------|---------------------------|
| Açı+Yer değiştirme | 83.19 | 89.12 | 81.17±7.39 |
| Açı + Yer değiştirme +GG | 85.4 | 89.0 | 78.0±5.28 |
| Açı+Yer değiştirme +DGH | 87.4 | 94.3 | 81.4±6.28 |

6.7. Sonuçlar

Bu kesimde ham derinlik bilgisi üzerinden öznitelikler çıkarılarak hareketler tanınmaya çalışılmıştır. Yapılan çalışmalarda ham derinlik verisiyle sınıflamada istenen düzeye erişilememiş ve bu öznitelikler daha önceki kesimlerde kullanılan ve iskelet modelinden elde edilen öznitelikler ile birleştirilerek mevcut sınıflama başarımı arttırılmaya çalışılmıştır. Yapılan deneylerde GG ve DGH yöntemlerinde RF ile sınıflama yapılması sınıflama başarımı üzerinde %1 düzeyinde bir iyileştirme sağlamıştır. Literatürde yer alan güncel çalışmalardan az bir miktar da olsa, daha iyi sınıflama sonuçları elde edilmiştir. SVM ile yapılan deneylerde başarımlar artışı sağlanamamıştır. Yapılan çalışmada MSR-Action 3D kullanılmasının nedeni, bu veri kümesinde iskelet ve derinlik bilgisinin senkron olmasıdır. Diğer veri kümelerinde iskelet ve derinlik bilgisi aynı anda bulunmamakta veya senkron olmamaktadır.

İlerleyen kesimde hareketin tamamını gözlemlemeden tanınmaya dönük çalışmalar yapılmıştır. Özellikle gerçek zamanlı oyun uygulamalarında düşük gecikmeli hareket tanıma önemli olduğu için, literatürde yapılan çalışmalar bu alana kaymaya başlamıştır. Bu nedenle, tez kapsamında son çalışma alanı olarak düşük gecikmeli hareket tanıma konusu üzerine gidilmiştir.

7. Düşük Gecikme Süreli Hareket Tanıma

Bu kesimde düşük gecikme süresi (*latency*) değerine sahip bir tanıma yöntemi geliştirilmeye çalışılmıştır. Düşük gecikme süreli hareket tanıma güncel ve gittikçe hız kazanan bir araştırma alanıdır [88-90]. Literatürdeki yöntemlerde genelde sınıflama için tüm hareket dizisine ihtiyaç duyulmaktadır. Bu tür çalışmalarda hareket dizisinin bir kesimi kullanılarak sınıflama yapıldığında sınıflama başarımı düşmektedir. Sadece sınırlı sayıda çalışmada hareketin tümünü görmeden düşük gecikme süreli hareket tanıma ele alınmaktadır.

Düşük gecikme süreli çalışmalardan en başarılısı Zanfir [90] ve diğerleri tarafından gerçekleştirilmiştir. Bu çalışmada yine Shotton tarafından önerilen iskelet modelinden elde edilen öznitelikler kullanılmaktadır. Öznitelik olarak eklem noktalarının konum bilgileri ve bu konum bilgilerinin 1. ve 2. dereceden türevleri kullanılmıştır. Eklem noktalarının konum bilgilerinin türevleri hareket dizisi üzerinden kayan bir pencere gezdirilerek hesaplanmıştır. Öznitelik çıkarma her çerçeve (*frame*) için ayrı ayrı yapılmıştır. Pencere merkezinde yer alan çerçeve için yapılmıştır. Öznitelikler çıkartılmadan önce konum bilgileri üzerinde bir normalizasyon işlemi yapılmıştır. Buradaki amaç farklı boyutlardaki şahıslardan kaynaklanacak farklılığın azaltılmasıdır. Eklem koordinatları üzerinde gerçekleştirilen bu normalizasyona ait algoritma aşağıda verilmiştir. Algoritmada $p_{start}^{(1)}$ kök eklem konumunu, r_i uzuvların ortalama uzunluklarını ve d_i iki eklem arasındaki uzaklığı belirtmektedir. Bu algoritmada genişlik önce arama yönetimine göre eklem noktaları dolaşmakta ve $p_{end}^{(i)}$ is bu arama yöntemiyle ulaşılan son eklem noktasını belirtmektedir.

```
 $p_{start}^{(1)} \leftarrow p_{start}^{(1)}$   
for all  $p_{start}^{(i)} \leftarrow p_{end}^{(i)}$  do  
     $d_i \leftarrow (p_{start}^{(i)}, p_{end}^{(i)})$   
     $d_i' = r_i \frac{d_i}{\|d_i\|}$   
     $p_{end}^{(i)} \leftarrow p_{start}^{(i)} + d_i'$   
end for  
return  $P' = [p_1', \dots, p_n']$ 
```

Eklem konumları kalça eklem noktasına göre görelî hesaplanmaktadır. Normalizasyon ve özniteliklerin çıkarılmasından sonra hareket açısından belirleyici çerçevelerin seçilmesine geçilmiştir. Buradaki amaç daha tanımlayıcı bir eğitim kümesi elde etmektir. Bu soruna çözüm için sınıflama işlemi eğitim sırasında yapılmıştır. Bir çerçeve seçilirken eğitim kümesinde yer alan komşuları göz önünde bulundurulmuş ve seçilecek çerçevenin sınıfına ait komşu sayısının toplam komşu sayısına oranı hesaplanmıştır. Hesaplanan bu değer aynı zamanda posterior olasılığıdır. Belli eşik değerinden yüksek olan çerçeveler seçilmiştir.

Tanımlayıcı çerçeveler seçildikten sonra sınıflandırma işlemine geçilmiştir. Sınıflandırmada kullanılan yöntem aşağıda belirtilmiştir. Belirtilen algoritmada c sınıf etiketlerini, X_{train} eğitim kümesini, $v(X_{train})$ ise sınıflayıcının (KNN) güven değerini (confidence) belirtmektedir.

p_{start}^1 Kök Eklem konumu

$p_{start}^{(1)} \leftarrow p_{start}^{(1)}$

for all time $t \leq T$ **do**

X_t çerçevesi için öznitelikleri hesapla

X_t nin K en yakın komşusunu bul

for all $X_{train} \in kNN(X_t)$ **do**

$s(c) \leftarrow s(c) + v(X_{train})$

end for

if $t > N_{min}$ and $\frac{s(c)}{\sum_b s(b)} > \theta$

then return $argmax(s(c))$

end for

return class = $argmax(s(c))$

Başka bir çalışmada Ellis ve diğerleri [88], yine hareket gerçekleşirken en az gecikme (*low latency*) ile hareketleri tanımaya çalışmışlardır. Bu yöntem MSRC-12 ve MSR-Action 3D veri kümeleri üzerinde 88.7% ve 65.7% başarı oranı sağlamıştır. Fakat yöntem test edilirken Cross-Subject Test yerine çapraz geçerlik testi (*cross validation*) uygulanmıştır. Bu yöntemde öznitelik olarak iskelet eklemelerine ait konum bilgilerinin farkları alınmıştır. Bir çerçeveye ait öznitelikler çıkarılırken o çerçevedeki eklem konum bilgileri ile 5 ve 30 çerçeve öncesinde yer alan eklem konum bilgileri arasındaki farklar kullanılmıştır. Fark işlemi eş eklemler arasındaki euclidian uzaklığı ile hesaplanmıştır. Diğer bir öznitelik kümesi ise o anki çerçeve ve 10 çerçeve öncesindeki tüm eklem noktaları arasındaki farktır. Özniteliklerin çıkarılmasından sonra farklı modeller ile hareketler tanınmaya çalışılmıştır. İlk olarak Bag of Words (BOW) yöntemi uygulanmıştır. Çerçeveler 1000 kümeye ayrılmıştır. Öznitelikleri çıkarılan çerçeve bir kümeye atanmış ve seride yer alan çerçevelerin küme etiketleri histogramı çıkarılmıştır. Bu histogramlar kullanılarak bir SVM modeli eğitilmiştir. Tanıma işlemi harekete ilişkin küme histogramının çıkarılması ve SVM ile sınıflandırılmasıyla yapılmıştır. Diğer modelde ise hareket serileri zaman serisi olarak ele alınmış bir Conditional Random Field (CRF) modeli oluşturulmuştur. Son modelde ise yine CRF kullanılmış fakat bu sefer tüm pozlar yerine sadece tanımlayıcı pozlar ele alınmıştır.

Tanıma işleminde sadece %30 oranında çerçeve ile yapılmıştır. %30 oranında çerçeve ile %90 düzeylerinde doğru sınıflama yapılmış olmasına karşın Cross-Subject Test yapılmadığı için bu başarımların değerleri çok fazla karşılaştırılabilir değildir.

Hoai ve Torre 'de [89] video görüntüleri üzerinde düşük gecikme süreli hareket tanıma üzerine bir yöntem önermişlerdir. Önerilen bu yöntemde hareketler online olarak tanınmaya çalışılmıştır. Hareketler bölütlere ayrılarak tanınmaya çalışılmıştır. Bir bölüt içinde sınıflandırıcı tarafından en yüksek değere ulaşılan hareket sınıfı atanmıştır. Daha sonra bir sonraki bölüte geçilmektedir. Tanıma işleminde eğer bir bölütte önceki bölüte göre daha yüksek skorlu bir hareketle karşılaşırsa, ilgili hareket yüksek skorlu olanla etiketlenmektedir. Bu yöntemde sınıflayıcı olarak SMM, SVM ve Structured Output SVM (SOSVM) yöntemleri sınıflayıcı olarak kullanılmaktadır. Bu yöntemle hareketin %30 civarı gözlemlendiğinde %65 civarında bir doğru sınıflama oranına erişilmektedir.

7.1. Düşük Gecikme Süreli Hareket Tanıma için Geliştirilen Model

7.1.1. Çerçeve Tabanlı Tanıma

Hareket tanıma süresinin kısaltılması kapsamında ilk olarak, harekete ait çerçeveler üzerinde sınıflama yapılması yöntemi denenmiştir. Bu amaçla ilk olarak eğitim kümesinde yer alan hareketlere ait çerçevelerden (*frame*) öznitelikler çıkartılmıştır. Bu öznitelikler o anki çerçeveye ait iskelet bilgisinden elde edilen eklem açıları, eklem noktalarının kalça eklem noktasına uzaklığı (F_k), eklem noktalarının konum bilgisinin önceki çerçevedeki konum bilgilerinden farkıdır (F_d) . Kalça eklemine uzaklık ve önceki çerçeve ile olan uzaklık öznitelikleri Eşitlik 7.2 ve Eşitlik 7.3'te belirtilmiştir.

$$P_i = \{X_1, \dots, X_n\} \mid X_i = \{J_1, J_2 \dots, J_{20}\} \mid J_i = \{x, y, z\}, P_k \text{ Kalça eklem noktası çerçeve} \quad (7.1)$$

$$F_k = \|P_i - P_k\| \quad (7.2)$$

$$F_d = \|P_{i+1} - P_i\| \quad (7.3)$$

Bu öznitelikler kullanılarak bir SVM modeli eğitilmiştir. Daha sonra test kümesinde yer alan bir harekete ait tüm çerçeveler bu sınıflayıcı ile sınıflandırılmıştır. Çerçeve sınıflama sonuçlarında çerçeveler üzerinde en çok sayıda etikete sahip sınıf, harekete ait sınıf olarak kabul edilmiştir. Ayrıca test sırasında belirleyici olmayan çerçevelerin elenmesi için sınıflama işlemi sonucunda düşük skor değerine sahip çerçeveler dikkate alınmamıştır. Yine eğitim sırasında belirleyici olan çerçevelerin seçilebilmesi için K en yakın komşusunun sınıf değerlerine bakılmış ve kendi sınıf değerine sahip örneklerin oranı diğer sınıf değerlerine sahip örneklere göre belirli bir eşik değerinin altındaysa seçilmemiştir. Bu yöntemle yapılan testlerde hareket ilk %30 ve %40'ının gözlemlenerek yapılan sınıflama sonuçlarında başarı oranı %30 için 0.34 ve %40 için ise 0.42 oranında kalmıştır. Sınıflayıcı olarak RF ve KNN yöntemleri de kullanılmıştır fakat sonuç üzerinde olumlu bir etki yaratmamıştır.

Daha sonra belirleyici çerçevelerin seçimi için çerçeveler arasındaki geçişteki değişim dikkate alınarak bir seçim yoluna gidilmiştir. Peş peşe gelen çerçeveler arasındaki enerji değişimi hesaplanmıştır. Enerji değerleri eklem peş peşe gelen çerçevelerdeki eklem noktalarının her üç ekseninde de birbirlerine olan uzaklıklarının

toplamları ile bulunmaktadır. Enerji değerinin hesaplanması Eşitlik 7.4'te belirtilmiştir. Kısacası aynı noktalara ait yer değiştirme bilgisi toplanarak hesaplanmıştır. Daha sonra en çok, en az ve ortalama enerji değerlerine sahip çerçeveler eğitim ve test sırasında yukarıda belirtilen yöntemde olduğu gibi kullanılmıştır. Maalesef bu yöntem sonucunda başarılı sınıflama oranındaki artış çok düşük olmuştur, doğru sınıflama oranı %1-2 oranında artmıştır.

$$E = \sum_{j=1}^{20} \|X_{i+1,j} - X_{i,j}\| , X_{i,j} = \{J_1, J_2 \dots , J_{20}\} \quad (7.4)$$

Çerçevelere ait öznitelikler arasında görelî sıra bilgisi de eklenmiştir. Çerçevenin indisinin, hareket dizisine oranı öznitelik vektörüne eklenerek sınıflamada kullanılmıştır. Sıra bilgisinin eklenmesi sınıflama üzerinde olumsuz etki yapmış ve sınıflama başarımını düşürmüştür. Çerçeve bazlı yöntemlerde farklı sınıflara ait hareketlerin benzer çerçeveler içermesi ve tüm hareketlerde başlangıç ve bitiş durumlarındaki benzerlik bu yöntemlerde başarılı sonuçlar elde edilmesine mani olmuştur.

Çerçeve üzerinden gidilen yöntemlerden sonra, kesim 5'te yer alan hareket dizilerinden öznitelik çıkarmaya dayalı yöntemler üzerinde durulmuştur. Kesim 5' te yer alan özniteliklere ek olarak eklem açıları'nın türevi, eklem konum bilgilerinin 2. türevleri ve eklem noktalarının kalça eklemine olan uzaklığı bilgileri de eklenmiştir. Eklem açıları'nın türevi Eşitlik 7.5 ile konum bilgilerinin 2. türevi ise Eşitlik 7.6 ile hesaplanmaktadır. A_k eklemlere ait açı serisini, F_k ise konum bilgileri serisini belirtmektedir. Bu yöntem ile %30 gözlem oranı ile Cross-Subject Test'te 0.56 düzeyinde bir doğru sınıflama oranına erişilmiştir. Sonraki aşamalarda zaman serisinden öznitelik çıkaran bu yaklaşım üzerinden gidilmiştir.

$$F_{Ak} = \frac{d}{dx} A_k \quad k = 1,2..8 \quad (7.5)$$

$$F_{k^2} = \frac{d^2}{dx^2} F_k + \frac{d^2}{dy^2} F_k + \frac{d^2}{dz^2} F_k \quad (7.6)$$

7.1.2. Hareket Dizisi Tabanlı Tanıma

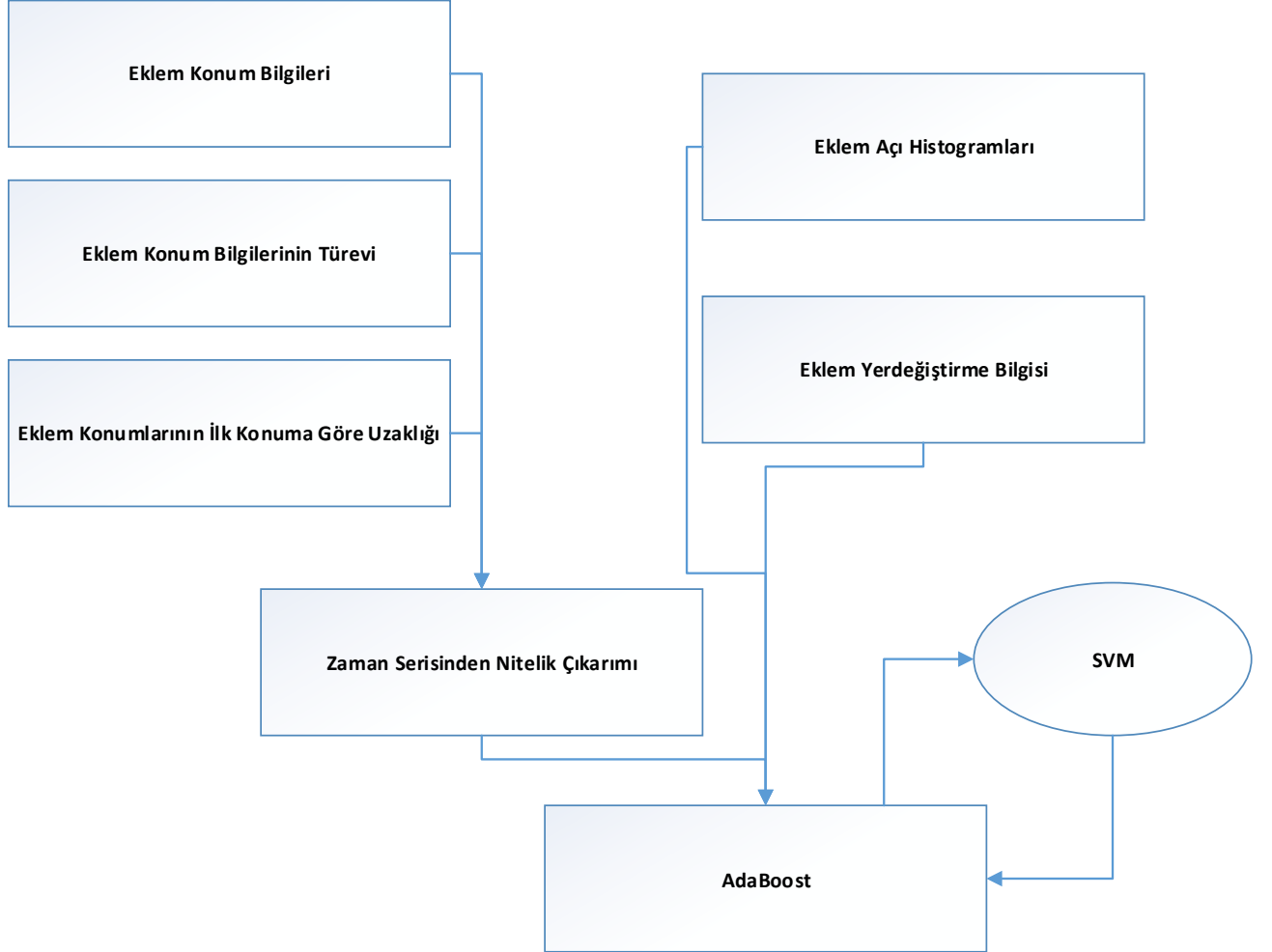
Hareketlerin düşük gecikme süresi ile tanınması kapsamında bir kesim öznitelikler zaman serisi olarak ele alınırken, diğer bir grup öznitelik ise hareketin gözlemlenen kesimden çıkartılarak kullanılmıştır. Zaman serisi olarak ele alınan öznitelikler üzerinden öznitelik çıkarmada [68]'de belirtilen yöntem kullanılmıştır. Bu yöntemde ilk olarak zaman serileri üzerinde 4. Kesimdeki öznitelik çıkarma yönteminde olduğu gibi WPT (Wavelet Packet Transform) kullanılmıştır. Fakat bu kez önceki yöntemde olduğu gibi PCA ile boyut indirgeme ve Fuzzy tabanlı bir nitelik seçimi yapılmamıştır. Yani hesaplanan bütün öznitelikler kullanılmıştır. Hesaplama açısından maliyet artmıştır. Kesim 5'te bahsedilen yöntemlerde olduğu gibi seri yine iki yeni alt sinyale ayrıştırılmıştır. Bu iki yeni dikey temelde yer alan ϕ_j fonksiyonları ölçek ve wavelet fonksiyonlarıdır. Bu işlem bir filtre bankası kullanılarak gerçekleştirilmekte ve sinyal 2 ayrı alt banda ayrılmaktadır. Öznitelikler yine wavelet sabitlerine ilişkin enerji değerlerinin hesaplanmasıyla bulunmaktadır. Bu işlem durağan olmayan sinyallerin analizinde kullanımı daha dikkat çekici ve kullanışlıdır. Fakat bu yöntemin uygulanması sonucu önceden de belirtildiği üzere öznitelik uzayında büyük ölçekli bir boyutluluk oluşmaktadır.

WPT dağılım ağacında yer alan her alt uzay bir öznitelik uzayı olarak ele alınmaktadır. Her alt uzay için WPT sabitlerinin katsayılarının karelerin toplamı alınıp sabit sayısına bölünmüştür. Daha sonra normalizasyon için elde edilen toplamın logaritması alınmıştır. Bu işlem Eşitlik 7.7'de belirtilmiştir.

$$E_{\Omega_{j,k}} = \log\left(\frac{\sum_n (w_{j,k,n}^T x)^2}{N/2^j}\right) \quad (7.7)$$

$w_{j,k,n}^T$ WPT sonucu elde edilen sinyali yani WPT sabitlerini belirtmektedir. $N/2^j$ ise alt bandta yer alan sabit sayısını belirtmektedir. Bu işlem sonucunda gözlemlenen kesimlerden sabit uzunlukta öznitelikler elde edilmiştir. Her hareket için gözlemlenen uzunluk farklı olsa da çıkarılan öznitelik vektörünün uzunluğu sabit olmuştur. Elde edilecek öznitelik vektörünün boyutu girilen j değerine göre değişmektedir. Önceki kesimlerde olduğu gibi doğrudan RF ve SVM yöntemleri denenmiş fakat doğru sınıflama oranları Adaboost yöntemine göre daha düşük çıkmıştır. Bu sebeple sınıflama işleminde bir genel sınıflayıcı (*ensemble*) yöntemi

olan Adaboost kullanılmıştır. Adaboost yönetminde temel sınıflandırıcı olarak ise SVM kullanılmıştır. Geliştirilen yönteme ilişkin akış diagramı Şekil 7.1.'de belirtilmiştir.



Şekil 7.1. Düşük Gecikme Süreli Hareket Tanıma Yöntemine ait akış çizeneği

7.2. Öznitelikler

Öznitelik grubumuzda ilk öznitelikler eklem noktalarının görel konumları F_k . Eklem noktalarının görel konumları Eşitlik 7.8'de de belirtildiği üzere seri olarak alınmıştır.

$$F_k = \{X_1, \dots, X_n\} \mid X_i = \{J_1, J_2, \dots, J_{20}\} \mid J_i = \{x, y, z\} \quad (7.8)$$

Doğrudan görelî konum değerlerine ek olarak konum değerlerinin değışimi de öznitelik olarak çıkarılmıştır. Konum değışimlerinin hesaplanması için konum bilgilerini her üç eksen de türevi alınmıştır (Eşitlik 7.9).

$$F_{kd} = \frac{d}{dx} F_k \parallel \frac{d}{dy} F_k \parallel \frac{d}{dz} F_k \quad (7.9)$$

Son olarak da konum bilgilerinin ilk konuma göre olan farkları alınmıştır. Eşitlik 7.10'da X hareket dizisinde yer alan eklem noktalarını, X_j^i ise başlangıç çerçevesindeki, X_i^c ise c çerçevesindeki eklem konum bilgilerini belirtmektedir.

$$F_{kb} = \{X_i^c - X_j^i \mid X_i^c \in X_c ; X_j^i \in X_i\} \quad (7.10)$$

Bu öznitelikler 5. Kesim de kullanılan özniteliklerdir. Diğer özniteliklerimiz ise Kesim 3'te çalışmalarımızda kullandığımız eklem açılara ait histogramlar ve eklem noktalarının X,Y,Z eksenelerindeki yer değıştirmeleri ve toplam 3 boyuttaki yer değıştirmesidir.

7.3. Öğrenme

Öznitelikler çıkarıldıktan sonra tanıma işleminde Adaboost yöntemi kullanılmıştır. Adaboost sınıflama başarımını artırmak için boosting kavramından faydalanmaktadır. Boosting görelî zayıf ve hatalı kuralların veya sınıflayıcıların birleşiminden güçlü tahmin yeteneğine sahip kural ve sınıflayıcıların elde edilmesidir. Bu kesimde Adaboost kullanılarak, harekete ilişkin az bir bilgi ile doğrudan SVM veya RF kullanıldığında ayırt edilmesi zor olan durumlar arasındaki ayırımın daha iyi yapılabilmesi amaçlanmıştır. Hareketin tamamı gözlemlenmediği zaman hareketlerden elde edilen nitelikler benzerlikler gösterebilmektedir. Özellikle hareketin %50'den daha az kesimleri incelendiğinde, öznitelikler arasındaki farklılıklar fazla olmamaktadır. Bu durumda daha ayırt edici bir sınıflayıcıya ihtiyaç duyulmaktadır. Bu kesimde sınıflama yöntemi olarak Adaboost kullanılmasının asıl sebebi de budur. Diğer kesimlerde, Adaboost kullanılmasının sınıflama sonucu üzerinde kayda değer bir etkisi görülmemiştir. Zaten hareketin tamamı gözlemlendiğinde elde edilen sonuçta önceki kesimlerde elde edilen sonuçlara çok yakındır. Özetle hareketin tamamı gözlenmediğinde Adaboost yöntemi bir fark yaratmamakta, fakat kısmi bir gözlem yapıldığında daha ayırt edici olmaktadır.

Freund ve Schapire tarafından önerilen Adaboost temelinde iteleme (boosting) algoritmasına dayanmaktadır. Bu algoritma zayıf modelleri birleştirerek güçlü bir model elde etmeye dayanmaktadır. Hata olasılığı ikili bir sınıflamada $\frac{1}{2}$ den biraz az olan modele zayıf, hata olasılığını istenildiği kadar düşürülebildiği modele de güçlü model denilmektedir. Büyük bir öğrenme kümesi 3 parçaya bölündükten sonra, ilk olarak ilk parçayı alıp X_1 kümesini eğitir ve d_1 modelini oluşturulur. Ardından X_2 kümesini test kümesi gibi alır ve d_1 modeliyle sınıflandırılır. Daha sonra d_1 'in hatalı sınıfladığı ve aynı sayıda doğru sınıfladığı örnekleri birleştirilip d_2 modeli eğitilir. Ardından X_3 kümesini alınıp d_1 ve d_2 modellerine verilir ve bunlardan farklı sınıflara atananlar alınarak d_3 modeli için öğrenme kümesi oluşturulur. Test aşamasında ise d_1 ve d_2 modellerinden aynı sınıflama sonucu çıkarsa tamam denilir fakat farklı çıkarsa bu örneği d_3 'e verir onu çıktısını kabul edilir. Bu yönden veri kümesi 3 yerine daha fazla sayıya da bölünebilir bu durumda daha fazla model geliştirilir ve her model kendinden öncekilerin yanlış sınıfladıklarıyla eğitilmektedir.

Bu çalışma kapsamında sınıflama aşamasında kullandığımız Adaboost.M1 yöntemidir. Bu yöntemdeki ana fikir eğitim kümesi oluşturulurken örneklerin seçilme olasılıklarının hataya bağlı olarak değiştirilmesidir. Örneğin bir (x^t, r^t) çiftinin j taban modelinde eğitilme olasılığı p_j^t olsun. Birinci model eğitilirken tüm olasılıklar eşit ve $p_1^t = 1/N$ dir. Sonraki modeller $j = 1$ den başlayarak eklenir. Adaboost modellerin zayıf olduğunu varsayar ve bu tersi bir durum sağlanırsa durur. Zayıf model hata oranı $1/2$ den az olan modellerdir. Burada hata ilk öğrenme kümesi üzerinde değil, j . adımda oluşturulmuş öğrenme kümesi üzerinden hesaplanmaktadır. B_j değeri Eşitlik 7.11 ile hesaplanmakta ve bir sonraki adımda yer alan ağırlıkların güncellenmesinde kullanılmaktadır. ε_j modelin hatasını göstermektedir.

$$B_j = \frac{\varepsilon_j}{1-\varepsilon_j} < 1 \quad (7.11)$$

j modeli bir örneği doğru sınıflandırırorsa bir sonraki modelde eğitim kümesine girme olasılığı Eşitlik 7.12 deki gibi olur ve azalır. Tersisi durumda ise eşit kabul edilir ve değişmez.

$$p_{j+1}^t = B_j p_j^t \quad (7.12)$$

Yani bir sonraki model önceki model tarafından yanlış sınıflandırılmış örneklere odaklanmaktadır. Adaboost algoritması aşağıda belirtilmiştir [80].

Öğrenme :

Tüm $\{x^t, r^t\}_{t=1}^N \in X$ için $p_1^t = 1/N$

Tüm modeller $j = 1, \dots, L$

X_j yi X den p_j^t olasılıklarıyla örnekleyerek oluştur

d_j yi X_j ile eğit

Her (x^t, r^t) için $y_j^t \leftarrow d_j(x^t)$ hesapla

Hatayı hesapla: $\varepsilon_j \leftarrow \sum_t p_j^t \cdot 1(y_j^t \neq r^t)$

Eğer $\varepsilon_j < 1/2$ ise $L \leftarrow j - 1$ dir

$$B_j = \frac{\varepsilon_j}{1 - \varepsilon_j}$$

Tüm (x^t, r^t) için çıktı doğruysa olasılıkları azalt

Eğer $y_j^t == r^t$ ise $p_{j+1}^t = B_j p_j^t$ değilse $p_{j+1}^t \leftarrow p_j^t$

Olasılık toplamları 1 olacak şekilde normalleştir.

$$Z_j \leftarrow \sum_t p_{j+1}^t ; p_{j+1}^t \leftarrow p_{j+1}^t / Z_j$$

Test :

X için model çıktıları hesapla $d_{ij}(x), j = 1, \dots, L, i = 1 \dots K$

Sınıf çıktıları hesapla $y_i = \sum_{j=1}^L \log(1/B_j) d_{ij}(x)$

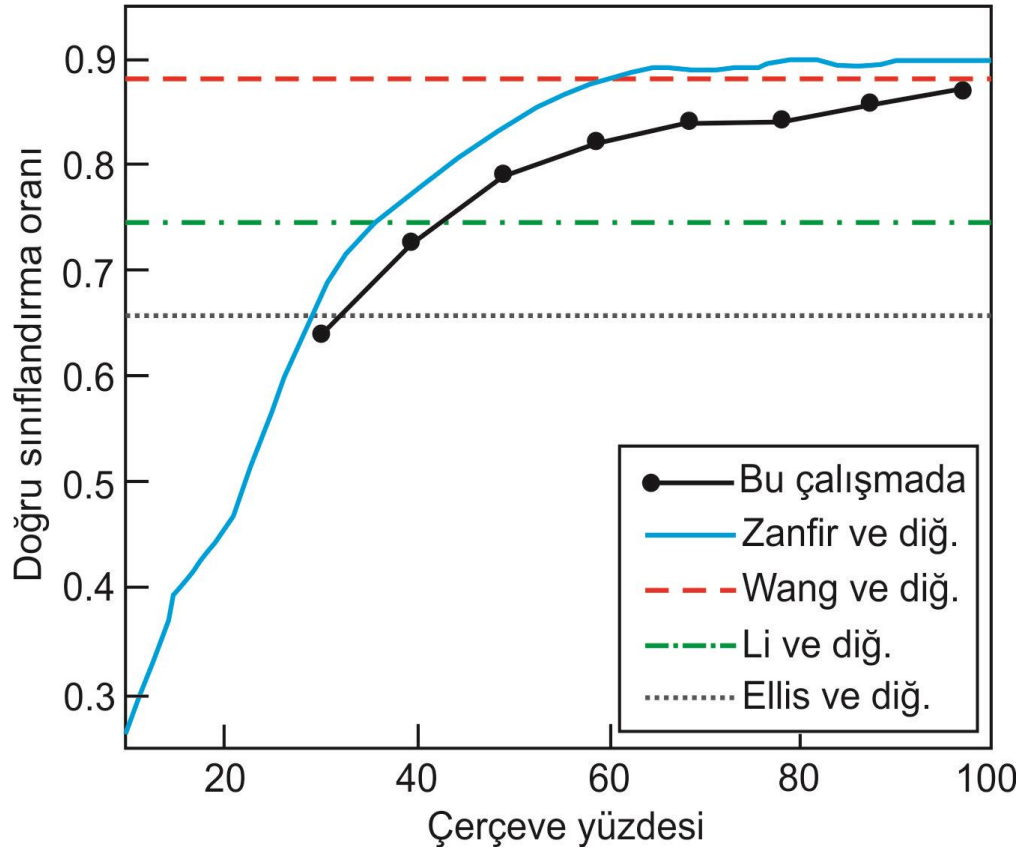
Adaboost algoritmasının test aşaması oylama yöntemiyle koşut olarak gerçekleştirilmektedir. Bir gözlem için tüm modellerin çıktıları hesaplanır ve sonuçlar ağırlıklı oylama ile birleştirilir. Ağırlıklar modellerin başarıları ile orantılıdır. Farklı modellerin öğrenme kümeleri arasındaki fark, modellerin hatalarına bağlıdır ve bir uygulamada Adaboost algoritmasının başarılı olup olmayacağı veriye ve kullanılan model algoritmasına bağlıdır. Bu çalışmada model algoritması olarak SVM kullanılmıştır. SVM çekirdeği olarak Radial Basis Function (RBF) kullanılmıştır. RBF çekirdeği kullanılmasının amacı sınıflayıcıyı zayıflatmaktır. Burada zayıflatmak ile kastedilen sınıflayıcının görelî sınıflama başarımını azaltmaktır. Adaboost'un temel mantığı zayıf kurallardan tahmin yeteneği daha güçlü kurallar elde etmek olması sebebidir. SVM lineer çekirdek ile daha yüksek doğru sınıflama oranlarına erişmektedir. Fakat Adaboost yöntemi zayıf sınıflayıcılardan güçlü sınıflayıcı oluşturmaya dayanmaktadır ve zayıflatılmış bir SVM ile Adaboost sınıflandırıcısının

başarımı artmaktadır [102]. Adaboost yöntemi kullanılırken SVM sınıflayıcısının zayıflatılması başarımı daha yüksek kılmıştır.

7.4. Sonuçlar

Bu kesimde hareket serilerinin belirli bir kesimi gözlenerek sınıflama yapılmıştır. Sınıflayıcı olarak Adaboost algoritması üzerinde SVM RBF çekirdeği model algoritma olarak alınıp sınıflama yapılmıştır. Geliştirilen yöntemin MSR-Action 3D veri kümesi üzerindeki sınıflandırma sonuçları, Şekil 7.2.'de literatürde yer alan diğer yöntemlerle [3,70,88,90] karşılaştırmalı olarak verilmiştir. Bu tür yöntemlerin kıyaslamasında sadece Cross-Subject Test kullanılmasından dolayı Cross-Subject Test sonuçlarına yer verilmiştir. Sadece Zanfir ve diğerleri [90] tarafından geliştirilmiş olan yöntem düşük gecikme süreli hareket tanıma üzerinedir.

Testler sırasında hareket serisinin en az %30'u gözlemlenmiştir. %30 altındaki gözlemlerde, az sayıda çerçeve bulunduğu için serilerden öznitelik çıkarma sırasında hatalar ile karşılaşmıştır. Ayrıca, az sayıda çerçeveden elde edilen öznitelikler özellikle kısa hareketler için ayırt edici olmamaktadır. Bazı hareketlerde hareketin %30'u alındığında gözlemlenen çerçeve sayısı 7-8 çerçeve gibi oldukça az olabilmektedir. Bu nedenle çok az sayıdaki çerçeveden de yeterince tanımlayıcı öznitelikler elde edilememektedir. Özellikle zaman serisinden elde edilen özniteliklerde, bu durum daha belirgin olmaktadır. Yine referans alınan diğer yöntemde [90] bakıldığında, %30 altında elde edilen sonuçlar anlamlı değildir. Önerilen çalışma bu yöntemde göre daha başarısız olsa da, sonuçlar yine de bu yöntemle karşılaştırılabilir düzeydedir. Farklı gözlem oranları için Cross-Subject Test'te elde edilen sonuçlar Çizelge 7.1.'de verilmiştir. Çizelge 7.1'de görüldüğü gibi, geliştirilen model, hareketin %30'nu görse bile %66,34 sınıflama başarımını yakalamaktadır. Bu başarımlar literatürdeki çoğu çalışmadan iyi düzeydedir.



Şekil 7.2. Literatürde yer alan diğer yöntemlerle geliştirilen yöntemin karşılaştırması, Li ve diğ. [3], Wang ve diğ.[70] Ellis ve diğ.[88] Zanfir ve diğ [90]

Çizelge 7.1. Cross-Subject Test için harekete ilişkin çerçeve görünme oranlarına göre başarımları

| 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|-------|-------|-------|-------|-------|-------|-------|-------|
| 66,34 | 74,17 | 80,15 | 83,11 | 84,71 | 84,93 | 86,51 | 87,97 |

8. Sonuç

Bu tez kapsamında derinlik bilgisini kullanarak insan hareketlerini tanıyan çeşitli yöntemler çalışılmıştır. Geliştirilen yöntemler genelde Kinect algılayıcısından elde edilen eklem iskeletine ait özniteliklere dayanmaktadır. Geliştirilen ilk yöntemde 20 eklemli iskelet modeli kullanılarak insan hareketleri tanınmıştır. Bu yaklaşımda önemli eklem noktaları arasındaki açı değerlerine ait histogramlar ve eklem noktalarının 3 boyutlu koordinat uzayındaki ve x,y,z eksenlerindeki yer değiştirmeleri öznitelik olarak kullanılmıştır. Hareketlerin sınıflandırılmasında SVM ve RF algoritmaları kullanılmıştır. Bu modeller kendi veri setimiz olan HUN-3D ve bu tür çalışmalarda sıkça kullanılan MSRC-12 ve MSR-Action 3D veri kümeleri üzerinde test edilmiştir. Önerilen yöntem bu veri kümeleri ile başarılı sonuçlar üretmiştir. Test-2 de RF algoritmasıyla sırasıyla HUN-3D, MSRC-12 ve Action-3D veri kümeleri üzerinde 93.5%, 98.7% ve 97.7% doğru sınıflama oranlarına ulaşılmıştır. Ek olarak RF modeli MSRC-12 üzerinde Test-1 ve Test-2 sonucunda 95% doğru sınıflama oranı yakalamıştır. Yine MSR-Action 3D üzerinde RF modeli 90% oranında doğru sınıflama yapmıştır. Cross-Subject Test'te ise doğru sınıflama oranı 83% ve 88.2% oranlarına gerilemiştir. Bu durumun sebebi aynı hareketlerin farklı kişiler tarafından gerçekleştirilmesi sonucunda oluşan farklılıklardır. Yine de literatürde yer alan diğer yöntemler ile karşılaştırıldığında sonuçlar başarılıdır. Bu yöntem hesaplaması kolay öznitelikler ile farklı veri kümeleri üzerinde başarılı sonuçlar üretebilmektedir. Bu durumun başlıca nedeni yöntemin önceden eğitilmiş iskelet çıkarma modelinden [7] elde edilen öznitelikler kullanılmasıdır. Hareketlerin daha iyi ayırt edilmesine olanak sağlamaktadır. İskelet çıkarma yöntemi eklem noktalarının tespit ve takibini son derece başarılı bir şekilde yaptığı için hareketlerin daha iyi ayırt edilmesine olanak sağlamaktadır. Buna ek olarak RF yönteminin SVM'ye göre daha bir sınıflama performansı göstermesi de sonuçların başarılı çıkmasındaki bir diğer etkidir.

Geliştirilen 2. yöntemde ise SMM kullanılarak hareketler tanınmaya çalışılmıştır. SMM özellikle çevrimiçi (online) hareket tanımada sıkça kullanılan yöntemlerden biridir. Bu yöntemde hareket serileri üzerinde bir pencere gezdirilerek öznitelikler çıkarılmış ve daha sonra bu öznitelikler kümelenecek bir sözlük oluşturulmuştur. Çıkarılan öznitelikler ilk kesimdekilerle aynı özniteliklerdir. Oluşturulan sözlük ile tüm hareketler birer sözcük dizisi haline getirilmiştir ve bu sözcük dizileri SMM ile tanınmaya çalışılmıştır. Bu yöntem önceki yönteme göre daha başarısız sonuçlar

üretmiştir. Sonuçlar Test-1 ve Test-2 de kabul edilebilir düzeyde olmasına karşın yöntemin kişilerden bağımsızlığını gösteren Cross-Subject Test'te başarısızdır. Bu durumun başlıca nedeni öznitelik çıkartılırken kayan pencere kullanılması ve kullanılan niteliklerin sadece pencere üzerinden çıkarılmasının tanımlayıcı olmaya yetmemesidir. Çünkü farklı kişiler aynı hareketleri gerçekleştirirken farklı hızlarda davranabilmekte ve buda pencere tabanlı öznitelik çıkarımı açısından sorun teşkil etmektedir.

Geliştirilen 3. yöntemde ise hareketler zaman serileri olarak ele alınmış ve zaman serilerinden öznitelik çıkarma amaçlı Khusbaba ve diğerleri [82] tarafından geliştirilen yöntem kullanılmıştır. Öznitelik olarak, eklem noktalarının konumları, konum bilgisinin türevi ve eklemlerin başlangıç konumuna göre uzaklıkları çıkarılmıştır. Bu öznitelikler tüm çerçeveler için çıkarılmış ve hareketler bir öznitelik dizisi haline getirilip bu farklı uzunluktaki öznitelik dizileri üzerinden sabit uzunlukta öznitelikler çıkartılmıştır. Bu yöntemle Cross-Subject Test'te 82.4% düzeyinde bir başarı sağlanmıştır. Başarı düzeyi ilk yöntem kadar yüksek olmasa da hareketi düşük gecikme süresi ile tanımda daha başarılı sonuçlar üreten bir yaklaşımdır. Bu nedenle son kesimde geliştirilen yöntemde bu yöntem baz alınmış ve sınıflama başarımını arttırmaya dönük geliştirmeler yapılmıştır.

Bir sonraki yöntemde ham derinlik bilgisi kullanılarak sınıflama başarımı arttırılmaya çalışılmıştır. Bu kesimde ham derinlik bilgisi kullanan 2 yöntem önerilmiştir. Bunlardan ilki olan Gradyan görüntüsü (GG) yönteminde tüm çerçevelere ait derinlik bilgisi bileştirilerek bir hacim oluşturulmuş ve bu hacme ait 3 eksendeki gradyan bilgisi hesaplanmıştır. Her çerçeve için hesaplanan gradyan değerleri kümülatif olarak toplanmış ve elde edilen matris üzerinden HOG öznitelikleri çıkarılmıştır. Diğer yöntem olan Derinlik Geçmiş Haritası (DGH) ise derinlik haritalarının indisi numaralarıyla çarpılıp toplanmasına dayanmaktadır. DHI üzerinden HOG öznitelikleri çıkarılmaktadır. Her iki yöntemde de hesaplanan HOG öznitelikleri açılı histogramları ve eklem yer değiştirme öznitelikleri ile birleştirilip sınıflama başarımı yükseltilmiştir. Tez kapsamındaki çalışmalarda en yüksek doğru sınıflama oranlarına bu yöntemler ile erişilmiştir. Literatürdeki yöntemler ile karşılaştırıldığında sonuçlar oldukça başarılıdır.

Son yöntem de ise hareketlerin düşük gecikme süresi ile tanınması yani tüm hareket dizisini görmeden hareketin sınıflandırılması amaçlanmıştır. Geliştirdiğimiz diğer yöntemler hareket dizisinin ilk %30, %40 lık kesimi ile yapılan testlerde başarısız sonuçlar üretmişlerdir. Yalnızca zaman serileri kullanan yöntem %50 üzerinde bir doğru sınıflama oranı sağlamıştır. Bu sebeple bu yöntem üzerinden devam edilmiştir. Çerçeveler üzerinden öznitelik çıkarıp çoğunluk oylaması ile çalışan yöntemler üzerinde de çalışılmasına rağmen arzu edilen başarı düzeyine çıkılamamıştır. Zaman serileri ile elde edilen öznitelikler üzerine ilk kesimde kullanılan öznitelikler de sadece hareketin gözlemlenen kesiminden elde edilerek eklenmiş ve başarımlar üzerinde bir artış sağlanmıştır. Bu yöntemde sınıflandırıcı olarak SVM kullanan Adaboost sınıflandırıcısından faydalanılmıştır. Adaboost genelde zayıf sınıflayıcılarla kullanılmasına rağmen SVM gibi güçlü bir sınıflandırıcı ile kullanılarak düşük gecikme süresi ile tanıma da kayda değer başarı oranlarına erişilmiştir. Elde edilen sonuçlar her ne kadar literatürdeki mevcut en başarılı çalışmayı geçemese de sonuçlar karşılaştırılabilir seviyededir. Mevcut durumun üzerinde geliştirmeler ile daha yüksek seviyeye çıkarılması hedeflenmektedir. Tez çalışmaları kapsamında geliştirilen yöntemler Çizelge 8.1’de karşılaştırılmalı olarak verilmiştir. Belirtilen sonuçlar MSR-action 3D üzerinde elde edilen sonuçlardır. Son yöntemde gösterilen sonuçlar hareketin %100’ü görüldükten sonra elde edilen sonuçlardır.

Çizelge 8.1 Geliştirilen yöntemlerin karşılaştırmalı başarımları

| | Test-1 | Test-2 | Cross-Subject Test |
|--------------------------|--------|--------|--------------------|
| Açı+Yer değiştirme | 95.3 | 97.7 | 88.2±7.39 |
| SMM | 87.82 | 93.75 | 68.03±5.44 |
| Zaman Serileri | 88.3 | 91.2 | 82.42±7.81 |
| Açı + Yer değiştirme +GG | 93.1 | 97.7 | 89.0±7.06 |
| Açı+Yer değiştirme +DGH | 95.4 | 98.0 | 89.4±6.86 |
| Düşük Gecikme Süreli | 86.8 | 90.1 | 87,97 |

İlerideki çalışmalarda hareketlerin çevrim içi olarak tanınması amaçlanmaktadır. İnteraktif kullanıcı uygulamalarında hareketlerin devam ederken tanınması gerekmektedir. Bu amaçla hem düşük gecikme süreli tanıma başarımının artırılması hem de gerçek zamanlı başarılı bir hareket bölütleme yönteminin geliştirilmesi gerekmektedir. Düşük gecikmeli hareket tanımada kullanılan modelin geliştirilmesi amacıyla zaman serileri arasındaki uzaklıkların çeşitli metrikler kullanılarak ölçülmesine dayalı öznitelikler ile DTW ve FFT tabanlı yöntemler test edilecektir. İleriye dönük bir başka çalışma alanı ise karmaşık hareketlerin tanınmasıdır. Birden fazla katmandan oluşan hiyerarşik bir sınıflayıcı kullanılarak ilk katmanlarda önce atomik hareketler tanınıp daha sonra üst katmanda ise bu atomik hareketlerin kombinasyonları olan karmaşık hareketler tanınabilir. Ayrıca birden fazla kişinin etkileşimde bulunduğu, dans etme, kavga etme gibi hareketlerin tanınması da, diğer bir gelecek çalışma alanı olarak düşünülebilir.

Kaynaklar

- [1] Sung, J., Ponce, C., Selman, B., Saxena, A., Human Activity Detection from RGBD Images, *AAAI workshop on Pattern, Activity and Intent Recognition*, **2011**
- [2] Xia, L., Chen, C., Aggarwal, J. K., Human Detection Using Depth Information by Kinect, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2011**
- [3] Li, W., Zhang, Z., Liu, Z., Action Recognition Based on A Bag of 3D points, *IEEE Computer Vision and Pattern Recognition Workshop*, **2010**.
- [4] Oreifej, O., Z. Liu, Z., and Redmond, W. A., HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences, *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2013**.
- [5] Doliotis, P., Stefan, A., McMurrough, C., Eckhard, C., Athisos, V., Comparing Gesture Recognition Using Color and Depth Information, *Conference on Pervasive Technologies Related to Assistive Environments (PETRA)*, **2011**.
- [6] Popa, M., Koç, A., K., Rothkrantz L., J., M., Shan, C., Wiggers, P., Kinect Sensing of Shopping related Actions, *Interactive Human Behavior Analysis in Open or Public Spaces Workshop*, **2011**.
- [7] Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A., Real-Time Human Pose Recognition in Parts from Single Depth Images, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2011**.
- [8] Luber, M., Spinello, L., Arras, K., O., Learning to Detect and Track People in RGBD Data, *RGBD-2011 Workshop*, **2011**.
- [9] Comaniciu, D., Meer, P. , Mean shift: A robust approach toward feature space analysis, *IEEE Trans. PAMI*, 24(5), **2002**.
- [10] Aggarwal, J.K. and Ryoo, M.S., Human activity analysis: A review. *ACM Comput. Surv.* vol. 43, 3, Article 16, **2011**.
- [11] Bobick, M., Davis, J., The recognition of human movement using temporal templates, *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 23-3 (Mar), 257-267, **2011**.
- [12] Shechtman, E., Irani, M., Space-time behavior based correlation, *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. 405-412, **2005**.
- [13] Ke, Y., Sukthankar, R., Hebert, Spatio-temporal shape and flow correlation for action recognition, *Visual Surveillance Workshop*, **2007**.

- [14] Sheikh, Y., Sheikh, M., Shah, M., Exploring the space of a human action, *In IEEE International Conference on Computer Vision (ICCV)*. Vol. 1. 144-149, **2005**.
- [15] Laptev, I., Lindeberg, T., Space-time interest points, *In IEEE International Conference on Computer Vision (ICCV)*. 432, **2003**.
- [16] Harris, C., Stephens, M., A combined corner and edge detector, *In Alvey Vision Conference*, 147-152, **1988**.
- [17] Lublinerman, R., Ozay, N., Zarpalas, D., Camps, O., Activity recognition from silhouettes using linear systems and model (in) validation techniques, *IEEE International Conference on Pattern Recognition (ICPR)*. 347-350, **2006**.
- [18] Shao, L., Gao, R., A Wavelet Based Local Descriptor for Human Action Recognition, *Proceedings of the British Machine Vision Conference*, 72.1-72.10, **2010**.
- [19] Ikizler, N., Duygulu, P., Histogram of Oriented Rectangles: A New Pose Descriptor for Human Action Recognition, *Image and Vision Computing*, Volume 27, Issue 10, pp. 1515-1526, **2009**.
- [20] Forsyth, D., Arikan, O., Ikemoto, L., O'Brien, J., Ramanan, D., Computational studies of human motion I: tracking and animation, *Foundations and Trends in Computer Graphics and Vision* 1 (2/3) , **2006**.
- [21] Rajpoot, N., Masood, K., Human Gait Recognition with 3D Wavelets and Kernel based Subspace Projections, *International Workshop on HAREM*, **2005**
- [22] Shawe-Taylor, J., Chritianni, N., *Kernel Methods for Pattern Analysis*, Cambridge University Press, **2004**.
- [23] Belhumeur, P., Hesponha, J., Kriegman, D., Eigenfaces vs Fischerfaces: Recognition using class specific projection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, **1996**.
- [24] Min, J., Kasturi, R., Activity Recognition on Multiple Motion Trajectories, *IEEE International Conference on Pattern Recognition*, **2004**.
- [25] Gupta, A., Davis, L., S., Objects in Action: An Approach for Combining Action Understanding and Object Perception, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2007**.
- [26] Freund, Y., Robert, E., S., A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting, *In Computational learning theory*, pp. 23-37, **1995**.
- [27] Yuan, X., & Yang, X., A robust human action recognition system using single camera, *In International Conference on Computational Intelligence and Software Engineering*, pp. 1-4, **2009**.

- [28] Boyer, R. S., & Moore, J. S., A fast string searching algorithm, *Communications of the ACM*, 20(10), 762-772, **1977**.
- [29] Chen, H. S., Chen, H. T., Chen, Y. W., & Lee, S. Y., Human action recognition using star skeleton, *In Proceedings of the 4th ACM international workshop on Video surveillance and sensor networks*, pp. 171-178, **2006**.
- [30] Nguyen, N. T., Phung, D. Q., Venkatesh, S., and Bui, H., Learning and detecting activities from movement trajectories using the hierarchical hidden Markov models, *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2. 955-960., **2005**.
- [31] Zhang, D., Gatica-Perez, D., Bengio, S., And McCowan, I., Modeling individual and group actions in meetings with layered HMMs, *IEEE Transactions on Multimedia* 8,3,509-420, **2006**.
- [32] Shi, Y., Huang, Y., Minnen, D., Bobick, A. F., Essa, I., A., Propagation networks for recognition of partially ordered sequential action, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2. 862-869, **2004**.
- [33] Ivanov, Y. A., Bobick, A. F., Recognition of visual activities and interactions by stochastic parsing, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22,8,852-872,**2000**.
- [34] Minnen, D., Essa, I. A., Starner, T., Expectation grammars: Leveraging high-level expectations for activity recognition, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 2. 626-632, **2003**.
- [35] Noguchi, A., & Yanai, K., A surf-based spatio-temporal feature for feature-fusion-based action recognition. *In Trends and Topics in Computer Vision* , pp. 153-167, Springer Berlin Heidelberg, **2012**.
- [36] Bay, H., Tuytelaars, T., & Van Gool, L., Surf: Speeded up robust features, *In Computer Vision–ECCV*, pp. 404-417, **2006**.
- [37] Lucas, B. D., & Kanade, T., An iterative image registration technique with an application to stereo vision, *In IJCAI* ,Vol. 81, pp. 674-679. **1981**.
- [38] Chen, M. Y., Mummert, L., Pillai, P., Hauptmann, A., & Sukthankar, R., Controlling your TV with gestures, *In Proceedings of the international conference on Multimedia information retrieval* ,pp. 405-408 ACM, **2010**.
- [39] Chen, M. Y., & Hauptmann, A. (2009). Mosift: Recognizing human actions in surveillance videos, **2009**.
- [40] Weinland, D., Ronfard, R., Boyer, E., Motion History Volumes for Free Viewpoint Action Recognition, *Computer Vision and Image Understanding* ,1th Edition, **2006**.

- [41] Pehlivan, S., Duygulu, P., A new pose-based representation for recognizing actions from multiple cameras, *Computer Vision and Image Understanding*, Vol 115/2 140-151, **2011**.
- [42] Roh, M., Shin, H., Lee, S., View-Independent human action recognition with Volume Motion Template on single stereo camera, *Pattern Recognition Letters*, vol. 31, 639-647, **2010**.
- [43] Huang, K. S., Triverdi, M., 3d shape context based gesture analysis with tracking using omni video array, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2005**.
- [44] Gu, J., Ding, X., Wang, S., Wu, Y., Action and Gait Recognition From Recovered 3-D Human Joints, *IEEE Transactions on Systems and Cybernetics* Vol. 40, No,4 , **2010**.
- [45] XMAS veri kümesi: <http://pascal.inrialpes.fr/data/human/>, (Şubat, **2012**)
- [46] INRIA veri kümesi: <http://4drepository.inrialpes.fr/public/viewgroup/6>, (Şubat, **2012**).
- [47] Stark, H., & Woods, J. W., *Probability, random processes, and estimation theory for engineers*, Englewood Cliffs: Prentice Hall, **1986**.
- [48] Zhou, S., K., Chelleppa, R., Moghaddam, B., Visual Tracking and recognition using appearance-adaptive models in particle filters, *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1491–1506 , **2004**.
- [49] Pehlivan, S., Duygulu, P., 3D Human Pose Search using Oriented Cylinders, in *IEEE Workshop on Search in 3D and Video (S3DV) in conjunction with the ICCV*, **2009**.
- [50] Canton-Ferrer, C., Casas, J.R., Pardasi, M., Human Model and motion based 3d action recognition in multiple view scenarios, *European Signal Processing Conference*, **2006**.
- [51] Hu, M. K., Visual pattern recognition by moment invariants, **Information Theory, IRE Transactions on**, 8(2), 179-187, **1962**.
- [52] Kohen, I., Li, H. Inference of human postures by classification of 3d human body shape, *IEEE International Workshop on Analysis and Modeling of Faces and Gestures*, **2003**.
- [53] Canny, J., A Computational Approach To Edge Detection, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, **1986**.
- [54] Barrow, H., Tenenbaum, J., Bolles, R. And Wolf, H., Parametric correspondence and chamfer matching: Two new techniques for image matching, *In Int'l Joint Conf. of Artif. Intel.*, pages 659–663., **1977**.

- [55] Dalal, N., Triggs, B., Histograms of oriented gradients for human detection, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2005**.
- [56] Streit, R. L., & Luginbuhl, T. E., *Probabilistic multi-hypothesis tracking* (No. Nuwc-Npt-Tr-10428), Naval Underwater Systems Center Newport, **1995**.
- [57] Myers, C. S., A comparative study of several dynamic time warping algorithms for speech recognition, Diss. Massachusetts Institute of Technology, **1980**.
- [58] Rabiner, L., R., Juang, B., Fundamentals of speech recognition, Prentice-Hall, Inc., **1993**.
- [59] Primesense API: www.primesense.org. (Şubat, **2012**).
- [60] Prismall, S., P., Object reconstruction by moments extended to moving sequences, Phd thesis, Department Electronic and Computer Science, University of Southampton, **2005**.
- [61] Raptis, M., Kirovski, D., & Hoppe, H., Real-time classification of dance gestures from skeleton animation, *In Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 147-156, **2011**.
- [62] Cheng, Z., Qin, L., Ye, Y., Huang, Q., & Tian, Q., Human daily action analysis with multi-view and color-depth data, *In Computer Vision–ECCV 2012 Workshops and Demonstrations*, pp. 52-61, Springer Berlin Heidelberg, **2012**.
- [63] Hsieh, C. H., Huang, P. S., & Tang, M. D., Human action recognition using silhouette histogram, *In Proceedings of the Thirty-Fourth Australasian Computer Science Conference-Volume 113*, pp. 11-16, Australian Computer Society, **2011**.
- [64] Yang, X., Zhang, C., & Tian, Y., Recognizing actions using depth motion maps-based histograms of oriented gradients, *In Proceedings of the 20th ACM international conference on Multimedia* ,pp. 1057-1060, **2012**.
- [65] Xia, L., & Aggarwal, J. K., Spatio-Temporal Depth Cuboid Similarity Feature for Activity Recognition Using Depth Camera, *In Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2834-2841, **2013**.
- [66] Oreifej, O., Z. Liu, Z., and Redmond, W. A., HON4D: Histogram of oriented 4D normals for activity recognition from depth sequences, *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **2013**.
- [67] Zhao, Y., Liu, Z., Yang, L., & Cheng, H., Combing rgb and depth map features for human activity recognition, *In Signal & Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Asia-Pacific (pp. 1-4), **2012**.

- [68] Yang, X., Tian, Y., EigenJoints-based Action Recognition Using Naive-Bayes-Nearest-Neighbor, *IEEE Workshop on CVPR for Human Activity Understanding from 3D Data*, **2012**.
- [69] Boiman, O., Shechtman, E., & Irani, M. (2008, June). In defense of nearest-neighbor based image classification. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-8, **2008**.
- [70] Wang, J., Liu, Z., Wu, Y., & Yuan, J., Mining actionlet ensemble for action recognition with depth cameras. *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1290-1297, **2012**.
- [71] Press, MS. "*PrimeSense Supplies 3-D-Sensing Technology to Project Natal for Xbox 360*". MsPress: p. MsPress. Retrieved March 31, **2010**.
- [72] Fothergill, S., Mentis, H., Kohli, P. and Nowozin, S., Instructing people for training gestural interactive systems, *In Proceeding of ACM Annu. Conf. Hum. Factors Comput*, pp 1737–1746, **2012**.
- [73] Li, W., Zhang Z. and Liu, Z., Action recognition based on a bag of 3D points, *In Proceedings of IEEE Comput. Vis. Pattern Recognit. Work. (CVPRW)*, **2010**.
- [74] Artin, M., *Algebra*, Prentice Hall. **1991**.
- [75] Juhl, L. J. and Bateman A., The rise and fall of supervised machine learning techniques, *Bioinformatic* Vol. 27 24, pages 2331-3332, **2011**.
- [76] Liaw, A., and Wiener, M., Classification and regression by randomForest, *R News*, 2, 18-22, **2002**.
- [77] Cawley, G. C., Talbot, N. L., & Girolami, M., Sparse multinomial logistic regression via bayesian l1 regularisation, *Advances in neural information processing systems*, 19, 209, **2007**.
- [78] Xia, L., Chen, C., Aggarwal, J. K., View Invariant Human Action Recognition with Depth Cameras, *IEEE CVPR Workshop on Human Activity Understanding from 3D Data*, **2012**.
- [79] Rabiner L. R., A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, *Proceedings of The IEEE*, Vol. 77, No: 2, February, pp 257-286, **1989**.
- [80] Alpaydm, E., *Introduction to Machine Learning*, MIT Press, ISBN 0-262-01211-1, **2011**.
- [81] MacQueen, J. B., Some Methods for Classification and Analysis of Multivariate Observations, *Proc. Symp. Math. Statist. and Probability (5th)*, 281–297, **1967**.
- [82] Khushaba, R. N., Al-Jumaily, A., & Al-Ani, A., Novel feature extraction method based on fuzzy entropy and wavelet packet transform for myoelectric

control, *In International Symposium on Communications and Information Technologies ISCIT'07*. pp. 352-357, **2007**.

- [83] Jolliffe, I., *Principal component analysis*. John Wiley & Sons, Ltd., **2005**.
- [84] Lee, H. M., Chen, C. M., Chen, J. M., & Jou, Y. L., An efficient fuzzy classifier with feature selection based on fuzzy entropy. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 31(3), 426-432, **2001**.
- [85] Pal, N. R., & Bezdek, J. C., Measuring fuzzy uncertainty, *IEEE Transactions on Fuzzy Systems*, 2(2), 107-118, **1994**.
- [86] Bezdek, J. C., *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers, **1981**.
- [87] Shashua A., Gdalyahu Y., and Hayon G., Pedestrian detection for driving assistance systems: Single-frame classification and system level performance, *In Proceedings of IEEE Intelligent Vehicles Symposium*, **2004**.
- [88] Ellis, C., Masood, S. Z., Tappen, M. F., LaViola Jr, J. J., & Sukthankar, R., Exploring the trade-off between accuracy and observational latency in action recognition, *International Journal of Computer Vision*, 101 (3), 420-436, **2013**.
- [89] Hoai, M., & De la Torre, F., Max-margin early event detectors, *International Journal of Computer Vision*, 107(2), 191-202, **2014**.
- [90] Zanfir, M., Leordeanu, M., & Sminchisescu, C., The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection. *In IEEE International Conference on Computer Vision (ICCV)*, pp. 2752-2759, **2013**.
- [91] Marr, D., & Vaina, L., Representation and recognition of the movements of shapes, *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 214(1197), 501-524, **1982**.
- [92] Rubin, J. M., and Richards, W. A., *Boundaries of visual motion*, Tech. rep., Massachusetts Institute of Technology, **1985**.
- [93] Rui, Y., & Anandan, P., Segmenting visual actions based on spatio-temporal motion patterns. *In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 111-118, **2000**.
- [94] Ogale, A., Karapurkar, A., Guerra-Filho, G., & Aloimonos, Y., View-invariant identification of pose sequences for action recognition, *In Video Analysis and Content Extraction Workshop (VACE)*, **2004**.
- [95] Feng, Z., & Cham, T. J., Video-based human action classification with ambiguous correspondences, *In IEEE Conference on Computer Vision and Pattern Recognition-Workshops, CVPR Workshops*, pp. 82-82, **2005**.

- [96] Ke, Y., Sukthankar, R., & Hebert, M., Efficient visual event detection using volumetric features, *In Tenth IEEE International Conference on Computer Vision*, Vol. 1, pp. 166-17, **2005**.
- [97] Ke, Y., Sukthankar, R., & Hebert, M., Event detection in crowded videos, *In IEEE 11th International Conference on Computer Vision ICCV*, pp. 1-8, **2007**.
- [98] Morency, L. P., Quattoni, A., & Darrell, T., Latent-dynamic discriminative models for continuous gesture recognition, *In IEEE Conference on Computer Vision and Pattern Recognition, CVPR'07.*, pp. 1-8, **2007**.
- [99] Sminchisescu, C., Kanaujia, A., & Metaxas, D., Conditional models for contextual human motion recognition, *Computer Vision and Image Understanding*, 104(2), 210-220, **2006**.
- [100] Shao, L., Ji, L., Liu, Y., Zhang, J., Human action segmentation and recognition via motion and shape analysis, *Pattern Recogn. Lett.* 33, 438-445, **2012**.
- [101] Saguna, Zaslavsky, A., Chakraborty, D., Recognizing Concurrent and Interleaved Activities in Social Interactions, *IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing (DASC)*, pp.230-237, 12-14, **2011**.
- [102] Li, X., Wang, L., & Sung, E., AdaBoost with SVM-based component classifiers, *Engineering Applications of Artificial Intelligence*, 21(5), 785-795, **2008**.

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı : Ali Seydi Keçeli
Doğum Yeri : İslahiye/Gaziantep
Medeni Hali : Bekar
E-posta : aliseydikeceli@gmail.com
Adresi : Keklikpınarı Mah. 906 sk. 39/7 Çankaya Ankara

Eğitim

Lise : İçel Anadolu Lisesi
Lisans : Hacettepe Üniversitesi Bilgisayar Mühendisliği
Yüksek Lisans : Hacettepe Üniversitesi Bilgisayar Mühendisliği
Doktora : Hacettepe Üniversitesi Bilgisayar Mühendisliği

Yabancı Dil ve Düzeyi

İngilizce- İyi (2011,KPDS 85)

İş Deneyimi

Entegre Enformasyon Sistemleri : Yazılım Mühendisi
Havelsan : Yazılım Mühendisi
Hacettepe Üniversitesi : Araştırma Görevlisi (Halen)

Deneyim Alanları

Görüntü işleme, veri madenciliği, makine öğrenmesi, OOP, JEE, Matlab

Tezden Üretilmiş Projeler ve Bütçeleri

Tezden Üretilmiş Yayınlar

- Ali Seydi Keçeli, Ahmet Burak Can, Recognition of Basic Human Actions Using Depth Information, International Journal of Pattern Recognition and Artificial Intelligence, **2014**
- Ali Seydi Keçeli, Ahmet Burak Can, A Multimodal Approach for Recognizing Human Actions Using Depth Information, 22nd International Conference on Pattern Recognition (ICPR), **2014**

- Ali Seydi Keçeli, Ahmet Burak Can, Action Recognition Based on Feature Extraction From Time Series, IEEE 22. Sinyal İşleme ve İletişim Uygulamaları (SIU) Kurultayı, **2014**
- Ali Seydi Keçeli, Ahmet Burak Can, Recognition of Human Actions By using Depth Information, IEEE 21. Sinyal İşleme ve İletişim Uygulamaları (SIU) Kurultayı, **2013**

Tezden Üretilmiş Tebliğ ve/veya Poster sunumu ile Katıldığı Toplantılar

IEEE 21. Sinyal İşleme ve İletişim Uygulamaları Kurultayı,2013

IEEE 22. Sinyal İşleme ve İletişim Uygulamaları Kurultayı,2014

22nd International Conference on Pattern Recognition (ICPR), 2014