ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

MAKİNE ÖĞRENMESİ YÖNTEMLERİYLE AKAN GÖRÜNTÜLERDEN OTOMATİK AKTİVİTE SINIFLANDIRMA

Elit Cenk ALP

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2018

Her hakkı saklıdır

TEZ ONAYI

Elit Cenk ALP tarafından hazırlanan "**Makine Öğrenmesi Yöntemleriyle Akan Görüntülerden Otomatik Aktivite Sınıflandırma**" adlı tez çalışması 11/04/2018 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Dr. Öğr. Üyesi Hacer YALIM KELEŞ
Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Jüri Üyeleri:

Başkan :Doç Dr. Hacer KARACAN
Gazi Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Üye :Dr. Öğr. Üyesi Hacer YALIM KELEŞ
Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Üye :Dr. Öğr. Üyesi Mehmet Serdar GÜZEL
Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Yukarıdaki sonucu onaylarım.

Prof. Dr. Atila YETİŞEMİYEN
Enstitü Müdürü

ETİK

Ankara Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez içindeki bütün bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davrandığımı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim.

11/04/2018



Elit Cenk ALP

ÖZET

Yüksek Lisans Tezi

MAKİNE ÖĞRENMESİ YÖNTEMLERİYLE AKAN GÖRÜNTÜLERDEN OTOMATİK AKTİVİTE SINIFLANDIRMA

Elit Cenk ALP

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Hacer YALIM KELEŞ

Videodan insan aktivitelerinin otomatik tespit edilmesi ve sınıflandırılması son yıllarda oldukça fazla miktarda araştırmacının üzerinde çalıştığı aktif bir araştırma alanıdır. Bu tekniklerin eğlence uygulamaları, güvenlik gözlemleme sistemleri, sağlık sistemleri gibi alanlarda önemli kullanım alanları vardır.

Tez çalışması kapsamında, Weizmann ve KTH veri kümelerine ait olan videolar ile çeşitli makine öğrenmesi yöntemleri kullanılarak aktivite sınıflandırma çalışması yapılmıştır. Bu çalışmalar kapsamında hem özniteliklerin çıkarımında, hem de akış sınıflandırmada iki farklı yaklaşım ele alınmıştır. Öznitelik çıkarımında, el ile özniteliklerin çıkarılması ve derin ağlar kullanılarak özniteliklerin çıkarılması-yaklaşımları kullanılmıştır. Gizli markov modeller (HMM) ve tekrarlayan sinir ağları'ndan uzun kısa-süreli hafıza (RNN-LSTM) tabanlı yaklaşımlar, akış sınıflandırmada kullanılmıştır.

İmgeden klasik özniteliklerin çıkarılmasında Hu'nun değişmez 7 momentleri, projeksiyon, hareket geçmişi görüntüsü gibi yöntemler kullanılmıştır. Hareket geçmişi görüntüsünün farklı bir türevi olarak değiştirilmiş hareket geçmişi görüntüsü (MMHI) yöntemi geliştirilmiştir. El ile kodlanmış öznitelik çıkarma yaklaşımlarının yanı sıra, imgeyi ifade eden öznitelikleri kendi çıkaran ve öğrenen evşimsel sinir ağı yöntemi de kullanılmıştır.

Weizmann veri kümesinde, öznitelik olarak MMHI ve Hu momentlerinin kullanıldığı ve HMM ile video sınıflandırılmanın yapıldığı deneyde % 99.35 başarı oranı yakalanmıştır. KTH veri kümesinde ise CNN ile öznitelikler bulunup, HMM ve LSTM ile sınıflandırılma yapılmıştır. CNN ve HMM kullanılarak yapılan deneylerde, KTH veri kümesinde yapılmış deneyler sonucunda literatürde kayda geçmiş en iyi sonuç olarak % 99.30 başarı oranı yakalanmıştır.

Nisan 2018, 67 sayfa

Anahtar Kelimeler: Makine Öğrenmesi, Hareket Sınıflandırma, Derin Öğrenme, Gizli Markov Model, İnsan Hareket Tespiti, Hareket Sınıflandırılması

ABSTRACT

Master Thesis

AUTOMATIC ACTIVITY CLASSIFICATION FROM TIME-SEQUENCED IMAGES USING MACHINE LEARNING METHODS

Elit Cenk ALP

Ankara University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Asst. Prof. Dr. Hacer YALIM KELEŞ

Human action classification from video streams is an active research topic that is worked on by many researchers. This domain has a wide range of application areas such as entertainment applications, surveillance systems, health care systems.

In the context of this study, human action classification is performed using the Weizmann and KTH video datasets using different machine learning techniques. In this scope, two different approaches are followed both for the feature extraction and for the classification. For feature extraction, we studied hand-coded feature extraction methods and deep learning based feature extraction methods.

We utilized 7 Hu Moments, projection and motion history image (MHI) approaches as for the classical features. As an alternative to MHIs, we proposed a Modified Motion History Image (MMHI) approach. In addition to the hand-crafted features, we utilized Convolutional Neural Networks (CNNs) for learning representations from images.

In Weizmann dataset experiments, we obtained % 99.35 accuracy by using MMHI and Hu moments as features and Hidden Markov Models (HMMs) in sequence modeling. In KTH dataset experiments, we extracted features using CNNs and utilized them with HMMs ad LSTMs in sequence modelling. We obtained the best reported result in the literature, i.e. % 99.30 , using the CNN and HMMs in KTH experiments. For sequence classification, HMMs and Long Short-Term Memory based Recurrent Neural Networks (RNN-LSTM) are utilized.

April 2018, 67 pages

Key Words: Machine Learning, Action Classification, Deep Learning, Hidden Markov Model, Human Motion Classification, Motion Classification

TEŞEKKÜR

Çalışmalarımı yönlendiren, araştırmalarımın her aşamasında bilgi, öneri ve yardımlarını esirgemeyerek akademik ortamda olduğu kadar beşeri ilişkilerde de engin fikirleriyle yetişme ve gelişmeme katkıda bulunan danışman hocam Dr. Öğr. Üyesi Hacer YALIM KELEŞ'e (Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı), çalışmalarım süresince birçok fedakârlıklar göstererek beni destekleyen eşim Duygu Can ALP'e, annem Ümmihan ALP'e ve babam Cengiz ALP'e en derin duygularla teşekkür ederim.

Elit Cenk ALP
Ankara, Nisan 2018

İÇİNDEKİLER

TEZ ONAYI SAYFASI
ETİK ..... **i**
ÖZET ..... **ii**
ABSTRACT ..... **iii**
SİMGELER DİZİNİ ..... **vii**
ŞEKİLLER DİZİNİ ..... **viii**
ÇİZELGELER DİZİNİ ..... **ix**
1. GİRİŞ **1**
2. LİTERATÜR ÖZETİ **4**
3. ÖZNİTELİKLERİN ÇIKARILMASI **8**
3.1 Hu'nun Değişmez 7 Momentleri **8**
3.2 Hareket Geçmişi Görüntüsü **10**
3.3 Değiştirilmiş Hareket Geçmişi Görüntüsü **12**
3.4 Evrişimsel Sinir Ağları **14**
3.4.1 Yerel çabuk kavrayan alanlar **15**
3.4.2 Paylaşılan ağırlıklar ve bias **18**
3.4.3 Havuz katmanı **22**
4. HAREKET SINIFLANDIRILMASININ MODELLEMESİ **24**
4.1 Gizli Markov Modelleri **29**
4.1.1 Tanım **29**
4.1.2 Tanıma **31**
4.1.3 Eğitim **32**
4.2 Tekrarlayan Yapay Sinir Ağları **32**
4.2.1 Tekrarlayan yapay sinir ağlarının problemleri **34**
4.2.2 Uzun kısa süreli bellek ağları **35**
5. YAPILAN DENEYLER **38**
5.1 Weizmann Veri Kümesiyle Yapılan Deneyler **38**
5.1.1 Siluet ve projeksiyon ile HMM **40**
5.1.2 Siluet ve Hu momentleri ile HMM **44**
5.1.3 Değiştirilmiş MHI ve Hu momentleri ile HMM **46**

5.1.4 CNN ile HMM 49
5.2 KTH Veri Kümesi ile Yapılan Deneyler 52
5.2.1 Değiştirilmiş MHI ve Hu momentleri ile HMM 53
5.2.2 CNN ile HMM 54
5.2.3 CNN ile LSTM 57
6. SONUÇLAR 61
KAYNAKLAR 63
ÖZGEÇMİŞ 67

SİMGELER DİZİNİ

3-D	Üç boyutlu
2-D	İki boyutlu

Kısaltmalar

CNN	Convolution Neural Network
ESA	Evrişimsel Sinir Ağları
GCP	Google Cloud Platform
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
KTH	Kungliga Tekniska Högskolan (Kraliyet Teknoloji Enstitüsü)
LSTM	Long Short Term Memory
MEI	Motion Energy Image
MHI	Motion History Image
MoG	Mixture of Gaussian
PCA	Principal Component Analysis
RNN	Recurrent Neural Network
SOM	Self- Organizing Feature Map
TSA	Tekrarlayan Yapay Sinir Ağları

ŞEKİLLER DİZİNİ

Şekil 3.1 Hareket enerjisi örneği (Bobick ve Davis, 2001) 11
Şekil 3.2 Değiştirilmiş hareket geçmişi örneği (Alp ve Keles 2017 13
Şekil 3.3 Çok katmanlı yapay sinir ağı örneği 14
Şekil 3.4 CNN'deki, 24x20 boyutundaki girdi vektörü 15
Şekil 3.5 CNN'deki, yerel çabuk kavrayan alana karşılık gelen gizli hücre 17
Şekil 3.6 İlk çabuk kavrayan alanın, gizli katmandaki ilk hücreye bağlanması 17
Şekil 3.7 İkinci çabuk kavrayan alanın, gizli katmandaki ikinci hücreye bağlanması ... 18
Şekil 3.8 Öznitelik haritası yöntemi ile çıkarılmış 3x20 x16 boyutundaki ilk gizli katman 20
Şekil 3.9 Yirmi farklı öznitelik haritası örneği (Nielsen 2015) 21
Şekil 3.10 Maksimum havuzlama işlemi ile oluşmuş 10x8 boyutundaki katman 22
Şekil 3.11 Evrişimsel ve havuzlama katmanından oluşmuş bir CNN (Nielsen 2015) 23
Şekil 4.1 MHI-Hu momentleri öznitelikleri ile HMM aktivite sınıflandırma sistemi (Alp ve Keles 2017) 24
Şekil 4.2 CNN öznitelikleri ile HMM veya LSTM ile sınıflandırma sistemi 26
Şekil 4.3 Basit bir Gizli Markov Modeli (HMM) 30
Şekil 4.4 Çıktısı olmayan bir RNN (Goodfellow vd. 2016) 33
Şekil 4.5 LSTM'deki tek bir hücrenin modeli (Goodfellow vd, 2016) 36
Şekil 5.1 Weizmann veri kümesinden birkaç örnek çerçeve 40
Şekil 5.2 MMHI – Hu özniteliği ile 5 durumlu HMM deneyinin karmaşıklık matrisi 47
Şekil 5.3 KTH veri kümesinden birkaç örnek çerçeve 53
Şekil 5.4 Video klip uzunluğu 30 ve 17 durum sayılı deneyin karmaşıklık matrisi 57
Şekil 5.5 KTH'da, 128-64 LSTM mimarisinde eğitim ve onaylama başarı oranları 59

ÇİZELGELER DİZİNİ

Çizelge 5.1 Weizmann'da, video kliplerin, küçültme ve izdüşüm öznitelikleri, HMM deneyleri sonuçları 41
Çizelge 5.2 Weizmann'da, video kliplerin, izdüşüm öznitelikleri, HMM deneyleri sonuçları 42
Çizelge 5.3 Weizmann'da, küçültme ve izdüşüm öznitelikleri, HMM deneyleri sonuçları 43
Çizelge 5.4 Weizmann'da, tek bir karenin, küçültme ve izdüşüm öznitelikleri, HMM deneyleri sonuçları 3
Çizelge 5.5 Weizmann'da video kliplerin, siluet ve Hu moment öznitelikleri, HMM deneyler sonuçları 44
Çizelge 5.6 Weizmann'da, siluet ve Hu öznitelikleri HMM deneyleri sonuçları 45
Çizelge 5.7 Weizmann'da tek bir karenin, siluet ve Hu öznitelikleri, HMM deneyleri sonuçları 45
Çizelge 5.8 Weizmann'da, video kliplerin, MMHI ve Hu öznitelikleri, HMM deneyleri sonuçları 46
Çizelge 5.9 Weizmann'da, MMHI ve Hu öznitelikleri, HMM deneyleri sonuçları 48
Çizelge 5.10 Weizmann'da, tek bir karenin, MMHI ve Hu öznitelikleri, HMM deneyleri sonuçları 48
Çizelge 5.11 Weizmann'da klip boyutuna göre, siluet ve MMHI öznitelikleri, HMM deneylerinin karşılaştırılması 49
Çizelge 5.12 Weizmann'da video kliplerin, CNN öznitelikleri, HMM deneyleri sonuçları 50
Çizelge 5.13 Weizmann'da video kliplerin, augmentation yöntemiyle çoğaltılarak, CNN özniteliğinin HMM deneyleri sonuçları 51
Çizelge 5.14 KTH'da, video kliplerin, MMHI öznitelikleri, HMM deneyleri sonuçları 55
Çizelge 5.15 KTH'da, video kliplerin, CNN öznitelikleri, HMM deneyleri sonuçları 56
Çizelge 5.16 KTH'da, video kliplerin, CNN öznitelikleri, LSTM deneyleri sonuçları 58

1. GİRİŞ

Video akışlarından; insan aktivitelerinin tanınması, otomatik video gözetleme, içerik tabanlı görüntü erişimi ve insan-bilgisayar etkileşiminin modellenmesi gibi birçok uygulamalar bilgisayarda görme için etkin araştırma alanlarıdır. Aktivite sınıflandırma sistemlerinde, videodan analizler yapılarak, videoda bulunan aktiviteler belirlenmektedir. Ancak bu işlem hem uzun sürmekte, hem de istenilen başarı oranları elde edilememektedir. Bu problem gün geçtikçe daha doğru ve daha hızlı çözülmesi gereken bir problem haline gelmektedir. Çünkü bu sistemler, son zamanlarda önem kazanmaya başlayan, sağlık sistemlerinde, eğlence yazılımlarında ve güvenlik sistemlerinde aktif olarak kullanılmaktadır.

Sağlık sistemlerindeki hasta, yaşlı insanların aktivitelerini takip eden uygulamalarda kişinin durumunun tespiti çok önemlidir. Yaşlı kişi, düştüğünde, uygulama düşme aktivitesini algılayıp, ambulansı arayarak insan hayatını kurtarabilir. Eğlence uygulamalarında, insan aktivitelerini tanıyabilen kameralar ile oyunlar oynanabilmekte ve spor uygulamalarında ise aktivite sınıflandırma sistemleri kullanılarak, sporcuların, aktivitelerinin takip edilmesiyle performans ölçülebilmektedir. Örneğin, spor dallarından tenis için özelleşmiş uygulamada yapılan maçta; oyuncu kaç kere servis karşılamış, kaç kere servis kullanmış gibi aktiviteleri bulunarak performansı ölçülebilmektedir.

Yaygın kullanılmakta olan bir başka sınıflandırma uygulama alanı ise güvenlik için kullanılan sistemlerdir. Geleneksel güvenlik sistemlerinde, güvenlik görevlilerini bütün gün boyunca kameraları takip ederek, hırsızlık gibi olayları kontrol etmektedir. Aktivite sınıflandırma sistemleri ile birlikte, kameralardan hırsızlık gibi olaylar otomatik olarak tespit edilip gerekli yerlere haberler verilebilmektedir. Böylelikle insan gözünden kaçabilecek aktiviteler de tespit edilebilmektedir.

Videodan aktivite sınıflandırırken, sadece resimleri kullanmak bazı durumlarda yeterli olmamaktadır. Başarı yüzdesini artırmak için resimlerin yanı sıra geçiş sıraları (akışları) da sınıflandırılmada kullanılmaktadır. Bu tez çalışmasında, zaman bazında aktiviteyi

sınıflandırmak için gizli markov model (Hidden Markov Model (HMM)) ve tekrarlayan sinir ağlarından (RNN) uzun kısa süreli hafıza (Long Short-Term Memory (LSTM)) kullanılmıştır. Resimlerin akışını zamanda modellerken kullanılan öznitelikleri bulmak için; hareket geçmişi görüntüsü (Motion History Image (MHI)), Hu'nun değişmez 7 momentleri (Ming-Kuei 1962) ve evrişimsel sinir ağı (Convolution Neural Network (CNN)) yöntemleri kullanılmıştır.

Öznitelik çıkarılmasında MHI (Bobick ve Davis 2001) yöntemi kullanıldığında, videoda bulunan karelerdeki, hareketli bölümleri birleştirerek, tek bir resim olarak göstermektedir. Bu yaklaşımda, videodaki son karedeki yapılan aktiveler daha beyaz, eski karelerdeki aktiviteler ise gittikçe koyulaşmakta ve bir süre sonra karelerdeki beyazlık yok olmaktadır. Bu sayede, hareket geçmişi görüntüsüne baktığımız herhangi bir zamanda, aktivitenin ne olduğu, nereden nereye gittiği algılanabilmektedir. Bu öznitelik ile tüm aktivite, zamansal olarak gri tonlarından oluşmuş bir resim üzerinde özetlenmektedir (Ahad vd. 2009).

Bu çalışma kapsamında, geleneksel MHI hesaplama yöntemi geliştirilerek, aktiviteyi sınıflandırmada daha etkin bir formülasyon uygulanmış, değiştirilmiş hareket geçmişi görüntüsü yöntem önerisi (MMHI) getirilmiştir (Alp ve Keles 2017). Bu değiştirilmiş hareket geçmişi görüntüsü yöntemi, aktivitede değişimini, geleneksel yöntemdeki gibi doğrusal olarak değil, parabolik olarak azaltarak hesaplamaktadır. MMHI yönteminin başarısı yapılan deneylerle gösterilmiştir.

HMM'de kullanılacak girdilerin boyutu büyük olduğunda, video akışı sınıflandırma performansı düşmektedir. Bu nedenle, resmi oluşturan ham piksel setini HMM'e girdi olarak sunmak iyi sonuç vermemektedir. Bütün pikselleri girdi olarak kullanmak yerine, daha az boyutta olan Hu'nun 7 momentleri girdi olarak kullanılmıştır.

Weizmann veri kümesinde, Hu'nun 7 momentlerinin girdi olarak kullanıldığı deneylerin kısa özeti şu şekildedir. Değiştirilmiş hareket geçmişi görüntüsü (MMHI) bulunduktan sonra, Hu'nun değişmez 7 momentleri hesaplanıp, HMM'in eğitilmesinde girdi olarak kullanılmaktadır. Her kategorideki öznitelikler, ayrı bir HMM model oluşturularak

eğitilmektedir. MMHI yöntemi ile bulunmuş olan zamansal özniteliklerin, HMM'deki, gizli katmanlarda modellenerek, aktivitenin deseni oluşturulmaktadır.

Veri kümesi olarak KTH kullanıldığında, MMHI ile çıkarılan öznitelikler, videoların aktivitelerini sınıflandırmasında kullanılmıştır. KTH veri setinde, MMHI yöntemiyle bulunmuş olan özniteliklerin, HMM ile sınıflandırılmasıyla elde edilen sonuç, Weizmann veri kümesi ile yapılan deneyden daha başarısız kalmıştır. MMHI ile çıkarılan öznitelikler (el ile seçilmiş) yerine, özniteliklerin otomatik olarak bulunduğu bir yöntem olan derin ağlar kullanılmıştır. Derin ağların kullanılmasıyla çıkarılan öznitelikler, MHI ile bulunan özniteliklerden daha iyi sonuç vermiştir. Derin ağlar kullanılarak bulunan öznitelikler HMM ile denendikten sonra aynı öznitelikler RNN-LSTM'de kullanılmıştır. HMM'in, RNN-LSTM'den daha iyi sonuç verdiği gözlemlenmiştir.

Tezin geri kalanı sırasıyla; Literatür Özeti, Öznitelikleri Çıkarılması, Hareket Sınıflandırmanın Modellenmesi, Yapılan Deneyler ve Sonuç bölümlerinden oluşmaktadır.

2. LİTERATÜR ÖZETİ

Videolardan, aktivitelerin sınıflandırılması ile ilgili literatürde bir çok çalışma vardır. Weinland vd. (2011), aktivite sınıflandırılması ile ilgili bir inceleme yayınlamışlardır. Çalışmalarında; aktivitelerin, mekansal ve zamansal yapısına göre, sürekli video akışındaki eylemleri nasıl parçalayıp ayırt ettiklerine göre ve kamera bakış açılarındaki varyasyonları nasıl ele aldıklarına göre sınıflandırma yapmışlardır. Bobick ve Davis (2001) çalışmalarında zamansal şablon (temporal template) gösterimini geliştirmişlerdir. Bu zamansal şablonda, insan aktivitesi, tek bir resim üzerinde gösterilmektedir. Bobick ve Davis, moment tabanlı öznitelikleri kullanarak, hareket geçmişi görüntüsü ve hareket enerjisi görüntüsü yöntemlerini tanımlamışlardır. MHI hesapladıktan sonra, sınıflandırma için Hu'nun değişmez 7 momentlerini öznitelik olarak kullanmışlardır. Bulunan öznitelikler, SVM ile eğitilmiştir. Bu çalışmalarda kullanılan özniteliklerden farklı olarak tezde, MHI ve MEI görüntüsü birleştirilerek MMHI bulunmuştur.

Yamato vd. (1992) çalışmalarında, aktiviteleri tanımlamak için ardışık olarak akmakta olan tenis resimlerini kullanmışlardır. Resimlerdeki aktiviteleri tanımlamak için ayrık (discrete) HMM kullanan bir çalışma yapmışlardır. Aktivite anındaki her resmin öznitelik vektörü, vektör niceleme (quantization) vasıtasıyla bir sembole tayin etmişlerdir. Daha sonra tayin edilen sembolleri, HMM eğitiminde kullanmışlardır. Ancak, niceleme sürecinde bazı bilgiler kaybolduğu için iyi sonuçlar elde edememişlerdir. Buna rağmen HMM ile ardışık resimlerin sınıflandırılmasında umut verici sonuçlar elde etmişlerdir. Sundaresan vd. (2003), yürüyüşlerden insanları tanımak için kesintisiz (continuous) HMM kullanmışlardır. Öznitelik olarak da siluetlerin uzunluklarını kullanmışlardır. Tezde, videodaki aktivitelerin sınıflandırılmasında Sunderasan vd. gibi kesintisiz (continuous) HMM kullanılmıştır.

Mendoza ve Blanca (2007), sürekli (continuous) HMM ile videoda aktivite sınıflandırma çalışmaları yapmışlardır. HMM'i eğitmek için şekil-bağlam (contour) özniteliği vektörü kullanmışlardır. KTH veri setinde, % 94 başarı oranı elde etmişlerdir. Martinez-Contreas vd. (2009), aktivite sınıflandırmasında, siluet tabanlı bir yaklaşım

kullanmışlardır. Aktivite modellenirken, MHI kullanmışlardır; zamandaki aktiviteyi, 2D modelleyerek çalışmışlardır. MHI ile öznitelikleri bulduktan sonra, Kohonen self-organizing feature map (SOM) ile niceleme (quantization) yapmışlardır. Bulunan öznitelikleri, ayrık HMM ile sınıflandırmışlardır. Tezde, ayrık HMM ve MHI kullanmak yerine, sürekli HMM ve MMHI kullanılıp deneyler yapılmıştır. Böylelikle Mendoza ve Blanca'nın çalışmasından KTH veri setinde daha iyi bir sonuç olan % 99.41 başarı oranı elde edilmiştir.

Vezzani vd. (2018), emisyon olasılık fonksiyonunu, mixture of gaussian (MoG) olarak modelleyerek kullanmışlardır. Siluetlerin histogramını, görüntüdeki siluet matrisi bulunduktan sonra, satırlardaki ve sütunlardaki piksel değerlerinin toplamı ile hesaplanmaktadır. Bu histogramları öznitelik olarak kullanmışlardır. Weizmann veri kümesinde % 96 başarı oranı elde etmişlerdir. Hsieh vd. (2011), insan figürünün üç parçasını karakterize eden, üç farklı kutupsal koordinat sistemi ile siluetleri çıkarmışlardır. Her kutupsal koordinat sisteminde, bazı hücreleri farklı açılarla bölümlendirerek niceleme işlemi (quantize) yapmışlardır. Principal Component Analysis (PCA) ile öznitelik sayısını azaltarak, Weizmann veri setinde % 98 başarı oranı elde etmişlerdir. Kuehne vd. (2014) insan aktivitelerini, zamansal olarak modelleyen bir sistem kullanmışlardır. Konuşma tanıma sistemlerinde olduğu gibi, HMM kullanılarak, aktivite birimlerini modellemişlerdir. Tezdeki deneylerde ise öznitelik olarak Vezzani gibi siluetlerin histogramı kullanılmıştır. Öznitelik ön işleme yöntemlerinden biri olarak PCA kullanılmıştır.

Ahad vd. (2009), MHI yöntemini kullanarak yönlü hareket geçmişi görüntüsünü (Directional Motion History Image (D-MHI)) geliştirmişlerdir. Bu yöntem, MHI ve MEI, optik akış açısına göre hesaplamaktadır. MHI, MEI'den elde edilen öznitelikleri ve Hu'nun 7 momentlerinin, özniteliklerini kullanmışlardır. Bu yöntemle % 93 başarı oranı elde etmişlerdir. Afsar vd. (2015), insanların aktivitelerini, videolardan otomatik olarak tespit edebilen ve aktivitelerini tanımlayabilen bir sistem geliştirmişlerdir. İnsan siluetlerindeki sınırları, temel bir öznitelik olarak kullanarak, niceleme (vector quantization) yöntemi ile sembollere çevirerek, ayrık HMM'de kullanmışlardır.

Le vd. (2011), independent subspace analysis algoritmasını geliştirerek, denetimsiz (unsupervised) bir öğrenme tekniği kullanmışlardır. Bu yöntemi, çeşitli derin ağ öğrenme yöntemleri ile harmanlayarak birçok veri kümesi için başarılı sonuçlar elde etmişlerdir. KTH veri setinde % 93.9 başarı oranı elde etmişlerdir. Ji vd. (2013), insana aktivitesini sınıflandırmak için zamansal öznitelikleri modellemede, 3D CNN'i kullanmışlardır. KTH veri setinde % 90.2 başarı oranı elde etmişlerdir. Simonyan ve Zisserman (2014), insan aktivitesini sınıflandırmada, iki akışlı (stream) bir CNN kullanmışlardır. Birinci akış, mekânsal öznitelikleri modellerken, ikinci akış çok çerçeveli optik akış verisini kullanarak, zamansal öznitelikleri modellemektedir. İki akıştan çıkan sonuçlar, SVM'i kullanarak veya ortalama hesaplayarak, sınıflandırmışlardır. SVM kullanarak, UCF-101 veri kümesinde (Soomro vd. 2012) % 88.0, HMDB-51 veri kümesinde (Kuhne vd. 2011), % 59.4 başarı oranı elde etmişlerdir. Karpathy vd. (2014), aktivite sınıflandırması için iki farklı CNN modeli kullanan bir sistem kullanmışlardır. Context akışı adı verilen CNN, düşük çözünürlüklü görüntüler ile eğitilirken, fovea akışı adı verilen CNN ise görüntünün orta bölümünden alınan, yüksek çözünürlüklü kısım ile eğitmişlerdir. Karpathy vd. erken füzyon, geç füzyon ve yavaş füzyon olmak üzere üç farklı füzyon modelini kullanmışlardır. En iyi sonucu, yavaş füzyon modeli ile elde edilirken, farklı füzyon tekniklerindeki performansların da birbirine oldukça yakın olduğunu bildirmişlerdir. Tezde, CNN öznitelik seçimi için kullanılmıştır.

Donahue vd. (2017), videoda aktivite sınıflandırılması için LSTM kullanmışlardır. Her video karesi ile beslenen CNN katmanlarının ardından, LSTM ağı ile sistem eğitilip sınıflandırma yapmaktadır. Buna benzer olarak Ng vd. (2015), aktivite sınıflandırması için CNN-LSTM kullanmışlardır. Bu çalışmada, CNN ve LSTM bir ağda birleştirilerek eğitim yapıldığında, UCF-101 veri setinde % 88.6 başarı oranı elde etmişlerdir. Tezde CNN ve LSTM direk birleştirilmek yerine, CNN ile bulunan öznitelikler, HMM ve LSTM'e ayrı ayrı verilip karşılaştırılması yapılmıştır.

Lei vd. (2016), videolardan, aktiviteyi sınıflandıran, CNN ve HMM'i birleştiren, hiyerarşik bir mimari kullanmışlardır. HMM'in emisyon dağılımında, Gaussian fonksiyonu yerine, CNN kullanmışlardır. Weizmann veri kümesinde % 89.2, KTH veri

kümesinde % 94.43 başarı oranı elde etmişlerdir. Lei vd. (2016), HMM ve CNN yöntemlerini birleştiren, bir aktivite-sınıflandırma sistemi kullanmışlardır. Çalışmalarında, emisyon olasılıklarının modellenmesinde kullanılan, Gaussian Mixture Modelini, CNN ile değiştirmişlerdir. KTH veri setinde % 93.97 başarı oranı elde etmişlerdir. Tezde, Lei'nin çalışması gibi HMM mimarisinin içerisinde CNN kullanılmamıştır. Bunun yerine CNN ile bulunmuş olan öznitelikler HMM'e girdi olarak verilip deneyler yapılıp KTH veri setinde % 99.41 başarı oranı yakalanmıştır.

Literatür çalışmaları incelendiğinde, HMM kullanılarak yapılan aktivite sınıflandırma çalışma sayısı son zamanlarda oldukça azaldığı görülmektedir. Öznitelik çıkarımında, elle bulunan yöntemler, son dönemlerde yerini otomatik olarak öznitelik çıkarabilen yöntemlere bırakmıştır. Çünkü, spesifik bir problemde kullanılan öznitelikler, el ile bulunan yöntemler başka bir problemde iyi sonuç vermeyebilirken, derin ağlar gibi otomatik olarak öznitelikleri bulan yöntemler ise başka problemlerde de iyi sonuç verebilmektedir.

3. ÖZNİTELİKLERİN ÇIKARILMASI

İnsan aktivitesi sınıflandırılırken, öznitelik çıkarımı çok önemli bir rol oynamaktadır. Bulunan öznitelikler ne kadar nitelikli ise, bu öznitelikler kullanılarak yapılan sınıflandırma da o kadar iyi olmaktadır.

Öznitelik bulunması, el ile bulunan öznitelikler ve öğrenilebilen öznitelikler olmak üzere ikiye ayrılmaktadır. El ile bulunan öznitelikler, verilerin kısıtlı olduğu, durumlarda daha iyi sonuçlar verirken, bir veri kümesinde iyi sonuç veren yöntem, başka bir veri kümesinde başarılı sonuçlar vermeyebilmektedir. Öğrenilebilen öznitelikler ise, verilerin çok olduğu durumlarda iyi sonuç verirken, verilerin az olduğu durumlardaysa, iyi sonuçlar verememesi en büyük dezavantajı olarak görülmektedir.

Sonraki bölümlerde KTH ve Weizmann veri kümelerinde yapılan deneylerde kullanılmış olan öznitelik seçim yöntemleri anlatılmıştır.

3.1 Hu'nun Değişmez 7 Momentleri

Tez çalışmasında yapılan deneylerin bir bölümünde, Hu'nun değişmez 7 momentleri (Ming-Kuei 1962) kullanılmıştır. Hu'nun 7 momentleriyle çıkarılan öznitelikler nitelikli olmakta ve bu öznitelikler kullanılarak başarılı sonuçlar alınabilmektedir.

Hu'nun değişmez 7 momenti ile verilerin değişimleri, eğrilikleri, ortalamaları, basıklıkları tanımlanmaktadır. Görüntülerden öznitelik çıkarılmasında, yaygın olarak kullanılmaktadır. Hu'nun 7 değişmez momentleri, matematiksel moment teorisinden, Hu tarafından esinlenilerek 1960 yılında geliştirmiştir. Bu yöntem ile nesne tanımlamada çok iyi başarılar elde edilmiştir. İki boyutlu görüntülerin tanımlama problemlerinde, çokça kullanılmaktadır. Sınıflandırma ile ilgili çeşitli uygulamalarda denenmiş, hızlı ve güvenilir özniteliklerin elde edildiği görülmüştür (Türkoğlu 2013).

İki boyutlu (a + b) dereceli ve f(x, y) dağılımlı bir görüntü için (klasik mekanikte

olduğu gibi) moment Eşitlik (3.1)'deki gibi tanımlanmaktadır.

$$m_{ab} = x^a y^b f(x,y).\partial x.\partial y \tag{3.1}$$

Eşitlik (3.1)'de a, b = 0, 1, 2, … değerlerini alabilir. Bu momentlerin değerleri, herhangi bozulma etkisine karşı, genelde değişmezler. Bu değişmeme etkisi, sonucunda merkez momentleri tanımlanmıştır ve Eşitlik (3.2)'deki gibi ifade edilmektedir.

$$\mu_{ab} = \iint (x - \bar{x})^a (y - \bar{y})^b f(x,y).\partial(x - \bar{x}).\partial(y - \bar{y}) \tag{3.2}$$

$\bar{x}$ ve $\bar{y}$ değerleri görüntünün x ve y koordinatlarının ortalama değerleridir. Bu değerler Eşitlik (3.3)'de tanımlanmıştır.

$$\bar{x} = \frac{m_{10}}{m_{00}} \; ve \; \bar{y} = \frac{m_{01}}{m_{00}} \tag{3.3}$$

Merkez momentleri, dönüşüm altında değişmezler. Genellikle, boyutlarına göre momentleri normalize etmek için merkezi momentler kullanılmaktadır. Bu işlem, μ_{00} moment değeri ile gerçekleştirilmektedir. Normalize merkez momenti, Eşitlik (3.4)'de gösterilmiştir.

$$\eta_{ab} = \frac{\mu_{ab}}{\mu_{00}^r} \tag{3.4}$$

Eşitlik (3.4)'de $r = 1 + \frac{(a+b)}{2}$'dir. Hu (Ming-Kuei 1962), yedi adet değişmez moment, normalize merkez momentlerini kullanarak geliştirmiştir. Hu'nun momentleri Eşitlik (3.5)'de hesaplanmaktadır.

$$\varphi_1 = \eta_{20} + \eta_{02}$$

$$\varphi_2 = (\eta_{20} + \eta_{02})^2 + 4\eta_{11}^2$$

$$\varphi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\varphi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{03} + \eta_{21})^2$$

$$\varphi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})]$$

$$\varphi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{13}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\varphi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{03} + \eta_{12})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \quad (3.5)$$

Hu'nun değişmez moment değerleri; döndürme, yer değiştirme, ölçeklendirme ve parlaklık işlemleri yapıldığında da değişmezler.

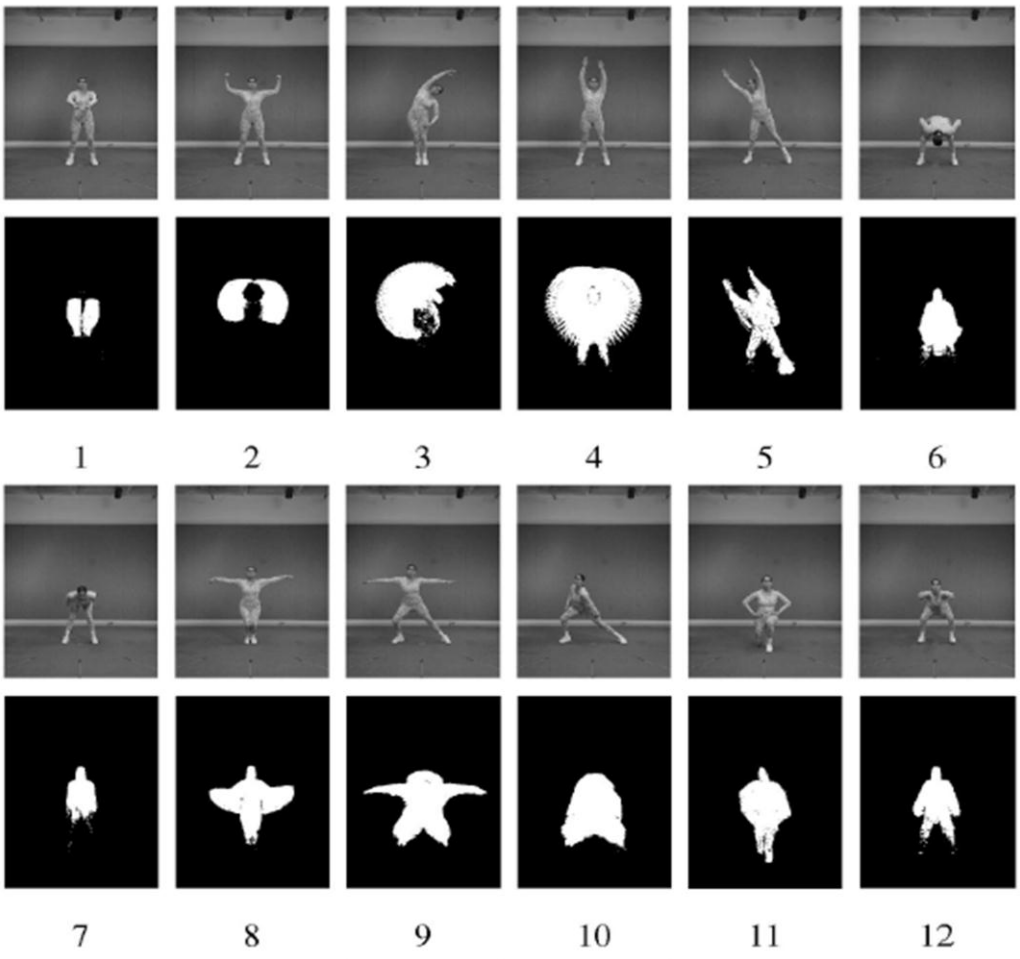
3.2 Hareket Geçmişi Görüntüsü

Hareket geçmişi görüntüsü (Weinland vd. 2011), her pikselin zamansal yoğunluğunu kullanarak aktivite akışını ve sırasını göstermektedir. Hareket geçmişi görüntüsü, video çerçevelerinde son zamanlara yaklaştıkça parlaklaşmakta, geçmiş zamanlara gittikçe de solgunlaşmaktadır. Daha yüksek yoğunluk değerine sahip olan pikseller, en son aktiviteye, daha düşük yoğunluk değerlere sahip olan pikseller ise daha önceden yapılan aktivitelere aittir. Eğer görüntü tamamen karanlıksa (0 değerine sahipse) o noktada yakın zaman içerisinde aktivite olmadığını göstermektedir.

Hareket geçmişi görüntüsüne bakarak aktivitenin olup olmadığı ve aktivitenin sınıfı tahmin edilebilmektedir. MHI'ın avantajlarından biri, birkaç görüntüde olan aktiviteyi tek bir resim ile gösterebilmesidir. MHI'ı ifade eden Hτ (x, y, t) fonksiyonu Eşitlik (3.6)'da gösterildiği gibi hesaplanmaktadır.

$$H_\tau(x,y,t) = \begin{cases} \tau & if\ \Psi(x,y,t) = 1 \\ \max(0, H_\tau(x,y,t-1)\ - \delta) & aksi\ halde. \end{cases} \quad (3.6)$$

Eşitlik (3.6)'da, (x, y) ve t, sırasıyla uzamsal konumu ve zamanı göstermektedir. τ parametresi MHI'ın, zamansal süresini belirtmektedir. δ, azalma parametresidir. Ψ (x, y, t), mevcut video görüntüsünde bir aktivitenin varlığını işaret etmektedir. Bu işaret çeşitli yöntemlerle bulunmaktadır. Videoda, sırası gelen her yeni video karesi için güncelleme fonksiyonu çağrılmaktadır. Literatürde, bu güncelleme fonksiyonunu tanımlamak için bir dizi farklı görüntü işleme tekniği, örneğin arka arkaya çıkarma, optik akış, görüntü farkları vb. kullanılmaktadır. MHI, çerçeve çıkarma işleminden elde edilen bir eşik ξ eşliğinde bir görüntüden oluşturulmaktadır (Eşitlik 3.7).



Şekil 3.1 Hareket enerjisi örneği (Bobick ve Davis 2001)

$$\Psi(x,y,t) = \begin{cases} 1 & if\ \ \mathrm{D}(x,y,t) \geq \xi \\ 0 & aksi\ halde. \end{cases} \tag{3.7}$$

Eşitlik (3.7)'de, D(x, y, t) iki görüntü arasındaki fark olarak alınmaktadır. Bu tez çalışmasında, D(x, y, t) olarak görüntü farkı fonksiyonu kullanılmaktadır.

Aktivite olan yerlerin, biriktirilerek tek bir resim haline getirilmesine de hareket enerjisi görüntüsü (MEI) denilmektedir. $E_T(x,y,t)$, hareket enerjisi resmini gösterirse;

$$E_T(x,y,t) = \bigcup_{i=0}^{\tau-1} D(x,y,t-i) \tag{3.8}$$

τ, hareketin sürdüğü video süresini ifade etmektedir (Bobick ve Davis 2001).

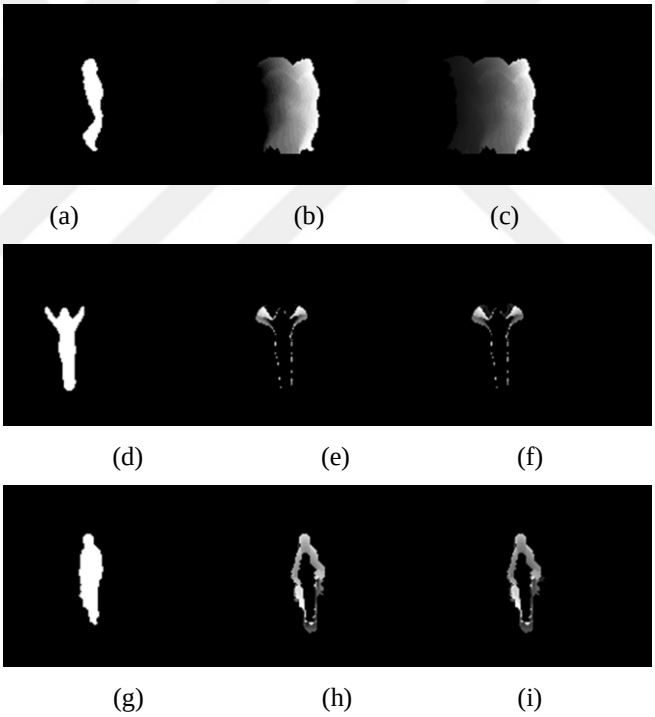
3.3 Değiştirilmiş Hareket Geçmişi Görüntüsü

Geleneksel hareket geçmişi görüntüsünde, bulunan aktiviteler zaman ilerledikçe, sabit bir değerle azalmaktadır. Değiştirilmiş MHI'da, sabit bir değer ile azalması yerine sabit bir değer ile oransal olarak azalma yapılmıştır. Bunun için Hτ fonksiyonunda, azalma değişkeni, zamanla oransal olarak azalacak şekilde değiştirilmiştir. Bu yaklaşım, sabit bir lineer azalma değişkeni yerine, üstel bir azalma değişkeni kullanılarak, son hareketin daha etkili vurgulaması açısından avantaj kazandırmaktadır. Tez çalışmasında bulunan bu yönteme, Modified MHI (değiştirilmiş hareket geçmişi görüntüsü) kısaca MMHI ismi verilmiştir.

$$H_\tau(x,y,t) = \begin{cases} \tau & if\ \ \Psi(x,y,t) = 1 \\ H_\tau(x,y,t-1) * \beta & aksi\ halde. \end{cases} \tag{3.9}$$

Eşitlik (3.9)'da, β, 0 ile 1 arasında bir ondalık değerdir; bu değişken zamanla, resimdeki değerleri, üstel olarak düşürmektedir. Ψ (x, y, t), ardışık gelen görüntü kareleri arasındaki farklar, gelen görüntünün, önceki görüntüden çıkarılması ile hesaplanmaktadır.

Tez kapsamında gerçekleştirilen deneylerin tamamında β, 0.9 olarak kullanılmıştır. Çünkü 0.9 değerinden daha düşük değerler seçildiğinde MMHI görüntüsü çok hızlı bir şekilde geçmekte, bu yüzden de geçmiş görüntüleri görülememektedir. Seçilen 0.9 değer ile zaman ilerledikçe aktivite olan yerlerin piksel değerlerinde % 10'luk bir azaltma oluşturulmaktadır. Ancak, piksel değeri, yuvarlama nedeniyle asla 4 değerinin altına düşememektedir. Bu değerin altına düşmemesi sayesinde hareket geçmişini de MEI gibi tutabilme kapasitesine sahip olmaktadır. Dolayısıyla, hareket geçmişi görüntüsü her iki amaç için de kullanılabildiğinden iki görüntünün de saklanmasına gerek kalmamaktadır. MEI, MMHI görüntüsündeki 0 değerinden büyük değerlerin hepsine 255 verilerek elde edilebilmektedir. MMHI'ın Weizmann veri kümesindeki bazı videolara uygulanmış hali şekil 3.2'de gösterilmektedir.



Şekil 3.2 Değiştirilmiş hareket geçmişi örneği (Alp ve Keles 2017)

MMHI oluşturma aşamaları şu şekilde özetlenebilir.

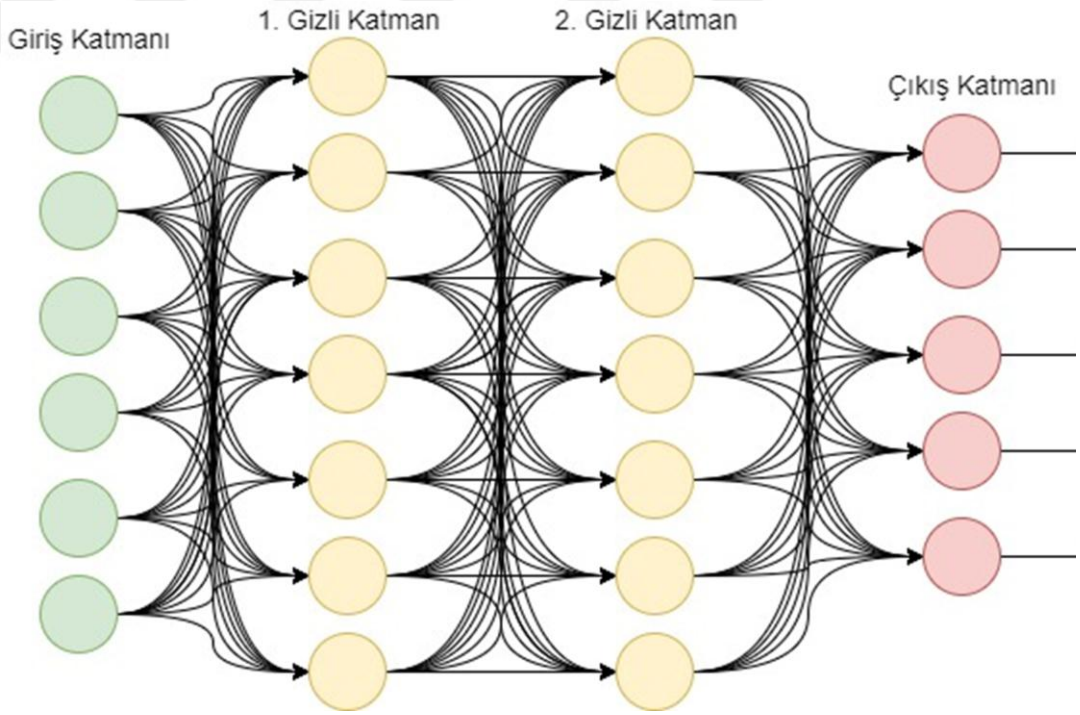
1. Oluşturulmuş olan MHI, 1 ile 0 arasında belirlenmiş olan katsayı, β, (Tez

çalışmasında 0.9 seçilmiştir.) ile çarpılmaktadır. Geçmiş zamanda olmuş olan aktivitelere ait olan pikseller, zaman ilerledikçe, her bir kare için % 10 oranında azalmaktadır.

2. İlerlemekte olan videonun, o andaki görüntüsü bir önceki görüntüsünden çıkarılmaktadır. Videodaki ışık değişimlerindeki bozulmalardan dolayı, oluşabilecek olan yanlış aktivite gösterimi, eşik değeri ayarlanarak aşılabilmektedir.
3. Bulunulan fark, o video için tutulmakta olan hareket geçmişi görüntüsüne eklenmektedir.

3.4 Evrişimsel Sinir Ağları

Çok katmanlı sinir ağlarında, sıralı olan katmanlar birbirine tamamen bağlı olarak bulunmaktadır. (Bir hücre, bir sonraki katmandaki tüm hücrelere bağlanmaktadır.)



Şekil 3.3 Çok katmanlı yapay sinir ağı örneği

(Nielsen 2015). Şekil 3.3'de, çok katmanlı bir mimari gösterilmektedir.

Şekil 3.3'deki giriş katmanında, 6 hücre bulunmaktadır. Ağda, 2 tane gizli katman bulunmaktadır. Bu gizli katmanlarda 7'şer adet hücre bulunmaktadır. Çıkış katmanı olarak 5 hücre bulunmaktadır. Çok katmanlı yapay sinir ağlarında, katmanlar arasında hücreler birbirlerine, belirli oranlarda bağlı bulunmaktadır. Bu oranlara, ağırlık denilmektedir ve eğitim sırasında bu ağırlıkların, değerleri hesaplanmaktadır.24 satır ve 20 sütundan oluşan bir resmi sınıflandırılmaya çalışılıyor ise, 24x20 sayıdaki hücreye girdi olarak, piksel değerleri verilmektedir. Bu durumda sinir ağında 480 (= 24 x 20) adet girdi hücresi olmaktadır. Bu ağın eğitiminde, 480 adet hücre, bir sonraki katmanda bulunan hücre sayısının çarpımı kadar ağırlık belirlenmelidir.

Görüntüleri tanımlamak için tüm piksellerde, tümüyle bağlı (fully connected) katmanlı yapı kullanmak uygun olmamaktadır. Çünkü tümden bağlı katmanlar, piksellerin mekânsal bilgisini dikkate almamaktadır. Örneğin birbirine çok yakın pikseller ile birbirine çok uzak pikseller, tümden bağlı katmanlarda aynı işlemi görmektedir. Bu yüzden çok katmanlı sinir ağları, görüntü tanımlamada, başarılı sonuçlar vermemektedir. Çok katmanlı yapay sinir ağı kullanmak yerine, görüntüleri tanımada daha başarılı sonuçlar veren CNN tez çalışmasında kullanılmıştır.

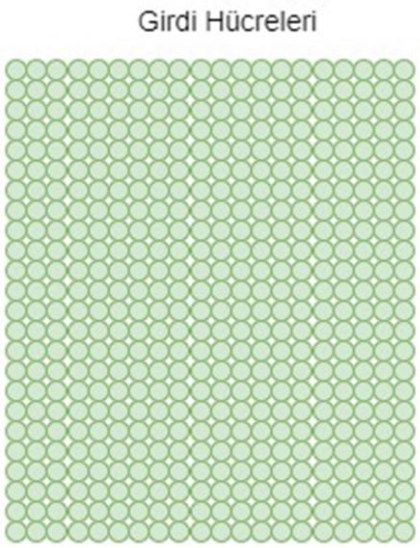
CNN mimarisinde, parametre sayısı az olduğu için, çok katmanlı sinir ağları mimarisinden daha hızlı eğitim yapılmaktadır. Bu sayede katman sayısı daha da büyültülerek görüntülerde, sınıflandırma yapmak mümkün hale gelmektedir.

CNN, üç temel ana fikir üzerinde inşa edilmiştir; yerel çabuk kavrayan alanlar (local receptive fields), bias ve paylaşılan ağırlıklar (shared weights) ve havuzlama (pooling) katmanı. Bu kavramlar, sonraki bölümlerde açıklanmaktadır.

3.4.1 Yerel çabuk kavrayan alanlar

Tümden bağlı (fully-connected) katmanlarda, girdi hücreleri dikey olarak gösterilirken,

evrişimsel sinir ağlarında, girdi hücrelerini dikdörtgen olarak gösterilmektedir. Bu sayede, tanımlanacak resmin yüzeyi gösterilerek daha iyi anlaşılır olması sağlanmıştır.



Şekil 3.4 CNN'deki, 24x20 boyutundaki girdi vektörü

Şekil 3.4'de 24 satır ve 20 sütundan oluşan bir resmin girdi hücresi olarak ele alınması gösterilmiştir.

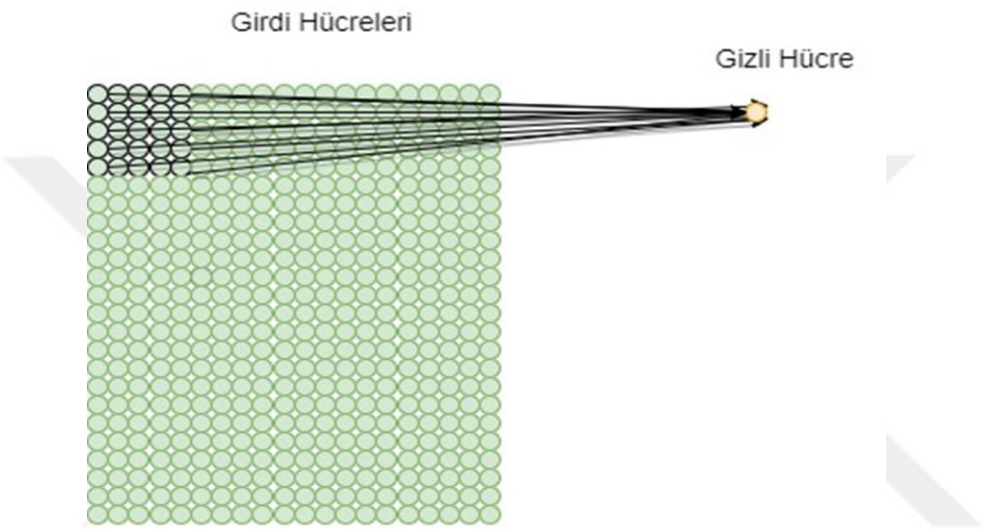
Çok katmanlı sinir ağlarında, bir piksele ait olan girdi hücresi, birinci gizli katmandaki tüm hücrelere bağlı bulunmaktayken, CNN'de, her girdi hücresi, birinci gizli katmandaki bütün hücrelere bağlı bulunmamaktadır. Bunun yerine, bir sonraki katmandaki, yerel bir alandaki gizli hücrelere bağlı bulunmaktadır.

İlk gizli katmandaki her bir gizli hücre, girdi hücrelerdeki belli bir bölgeden beslenmektedir. Eğer yerel alan olarak 5x5'lik bir bölge seçilmiş ise; 25 girdi hücresi, bir gizli hücreye bağlanmaktadır. Şekil 3.5'de gizli hücreye, girdi hücrelerinin tamamı değil; sadece 5x5'lik alandaki hücrelere bağlanmış bulunmaktadır.

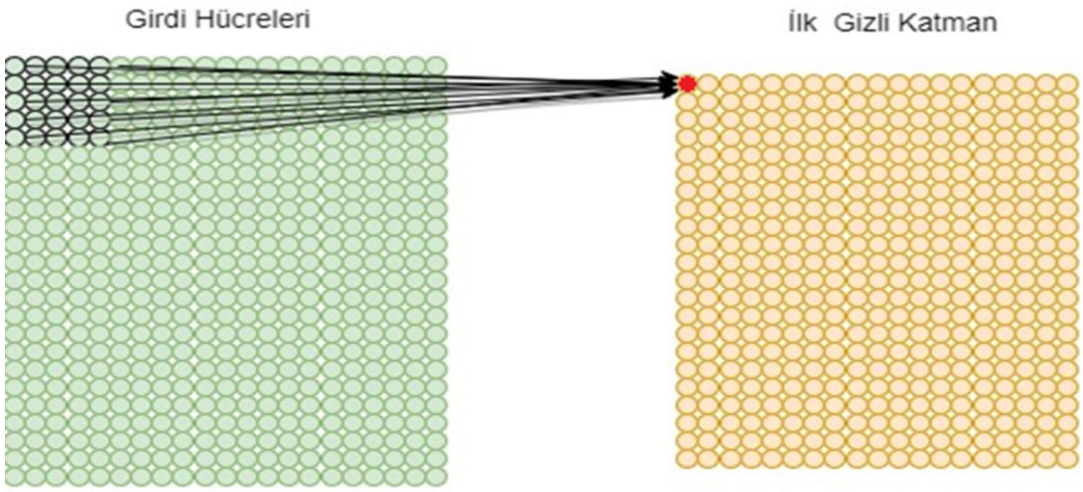
Bu gizli katmanda bulunan hücrenin bağlı olduğu bölgeye, yerel çabuk kavrayan alanlar (local receptive fields) denir. Her gösterilen ok işareti, hücreler arasındaki bağı göstermektedir. Bu bağın güçlülüğü veya zayıflığı, ağırlık denilen katsayı ile

belirlenmektedir. Eğitim sırasında, bu ağırlıklar güncellenmektedir.

Yerel çabuk kavrayan alanlar kaydırılarak, tüm görüntü üzerinde gezdirilmektedir. Her bir alan için, bir sonraki gizli katmanda bir hücre bulunmaktadır. Şekil 3.6'da girdi hücrelerinin, sol üst köşesinden başlayarak 25 (=5 x 5) tane hücre için ilk gizli katmana yapılan bağlar gösterilmektedir.



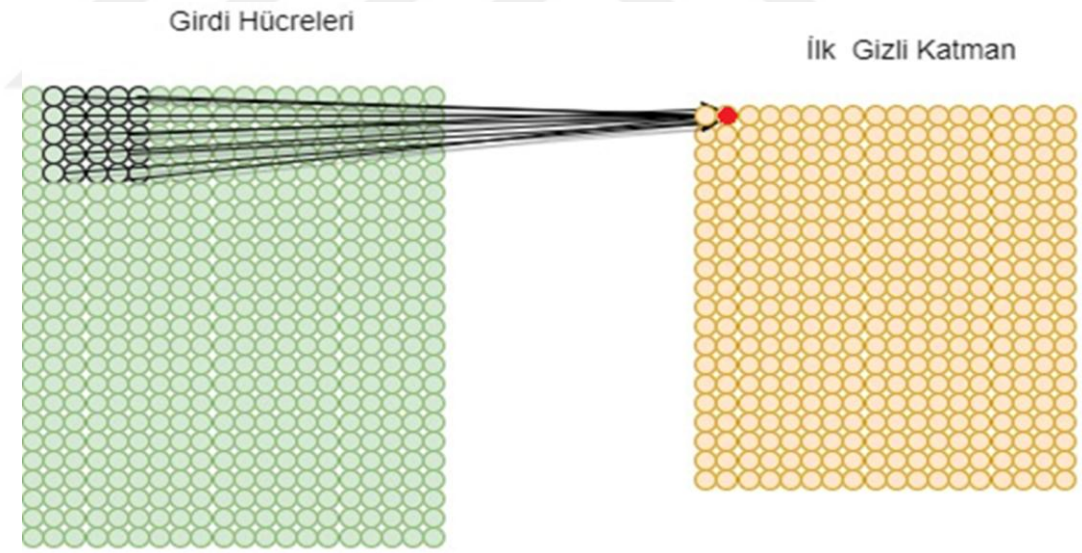
Şekil 3.5 CNN'deki, yerel çabuk kavrayan alana karşılık gelen gizli hücre



Şekil 3.6 İlk çabuk kavrayan alanın, gizli katmandaki hücreye bağlanması

Tüm görüntünün taranması için yerel çabuk kavrayan alanı resim üzerinde bir piksel ilerleyerek gezdirilmektedir. Şekil 3.6'da bulunan 25 tane yerel çabuk kavrayan alan, bir sağa kayarak ikinci hücreyi beslemektedirler. Şekil 3.7'de ilk gizli katman ikinci gizli hücre için girdi hücreleri gösterilmektedir.

Yerel çabuk kavrayan alan, sağa ve aşağı kaydırılarak, ilk gizli katman oluşturulmaktadır. Eğer 24x20 satır ve sütundan oluşan bir görüntü ve yerel çabuk kavrayan alan 5x5'lik bir alan ise gizli katman 20x16 hücreden oluşturulacaktır. Çünkü sadece 20 hücre sağa ve 16 hücre aşağı kayarak ilerlemektedir. Genellikle yerel çabuk kavrayan alanlar, bir piksel atlayarak ilerlemektedir. Bazen farklı adım mesafeleri de seçilebilmektedir. Örneğin, resim boyutu 800x600 gibi bir boyut olduğu durumlarda, adım mesafesini daha büyük seçerek, gizli katmandaki hücre sayısı azaltılabilmektedir. Buna bağlı olarak, eğitim sırasında daha az ağırlık hesaplanacağı için daha hızlı eğitim yapılabilmektedir.



Şekil 3.7 İkinci çabuk kavrayan alanın, gizli katmandaki ikinci hücreye bağlanması

3.4.2 Paylaşılan ağırlıklar ve bias

Gizli katmandaki hücrelerin, eğitime katılmasında kullanılan parametreye bias denilmektedir. Her gizli hücrenin parametre sayısı, bir önceki girdi katmanındaki 5x5

alandaki hücrelerle bağlantılı olan ağırlıklar ile bias parametre sayısının toplamı kadar olmaktadır. Gizli hücrenin çıktısı Eşitlik (3.10)'da gösterildiği gibi hesaplanmaktadır.

$$\sigma\left(b + \sum_{l=0}^{4} \sum_{m=0}^{4} w_{l,m} a_{j+l,k+m}\right) \tag{3.10}$$

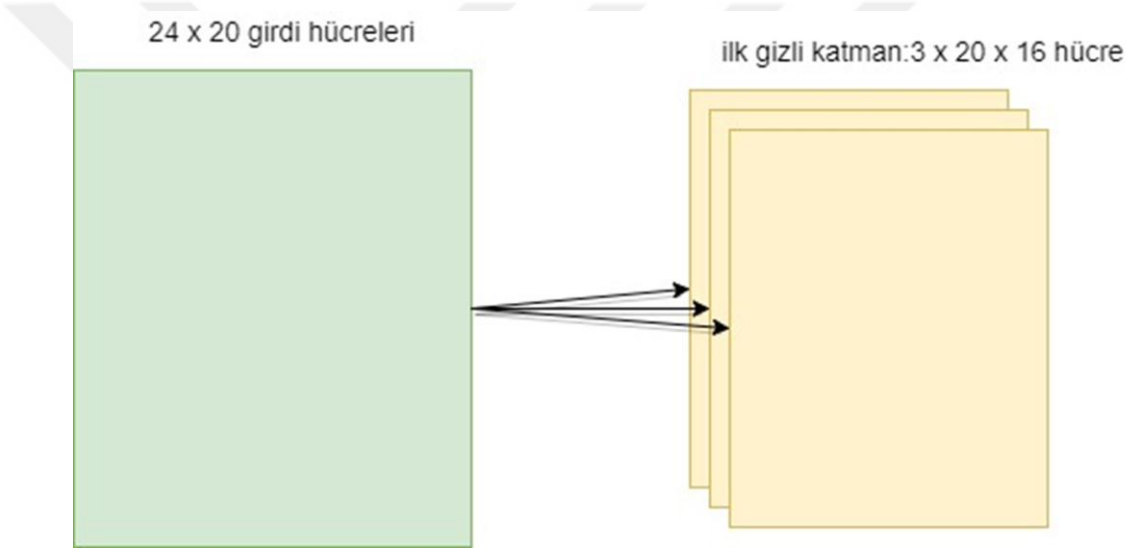
Eşitlik (3.10)'da, σ parametresi, hücre aktivasyon fonksiyonudur. Genellikle evrişimsel sinir ağlarında, aktivasyon fonksiyonu olarak rectified linear unit kullanılmaktadır. Kısaca ReLU olarak adlandırılmaktadır. Çok katmanlı sinir ağlarında aktivasyon fonksiyonu, genellikle sigmoid seçilmektedir. Bias değişkeni, b olarak gösterilmektedir ve ortak bir değer olarak Eşitlik (3.10)'da gösterilmektedir. Ortak ağırlık değişkeni $w_{l,m}$ olarak gösterilmektedir ve sayısı yerel çabuk kavrayan alan kadardır. Örneğin 5x5 boyutunda bir yerel çabuk kavrayan alan bulunuyorsa, l ve m değerleri de 1'den 5'e kadar olmaktadır. Değişken $a_{x,y}$ resimdeki x satırındaki ve y sütunundaki, piksel değerini, girdi olarak göstermektedir.

İlk gizli katmanın tüm hücreleri, aynı ağırlık değerine sahip olduğu için aynı öznitelikleri çıkarmaktadır. Bir gizli hücre ile diğer hücreler arasındaki tek fark görüntünün farklı yerlerinden çıkarılan öznitelikler olmalarıdır. Bulunan ortak ağırlık ve bias, bir kenar belirliyorsa, resmin herhangi bir yerinde de kenar bulma işlemini yapacaktır. Bu nedenle resimdeki aynı öznitelik belirleyicinin kullanılması faydalı olacaktır.

Örneğin, köpek resmini tanımlayan bir ağırlık ve bias değeri eğitim ile bulunmuş olsun. Köpek, resmin neresinde olursa olsun, köpek için belirlenen ağırlık ve bias değerleri kullanılacağından, köpek algılanacaktır. Bu nedenlerden dolayı, girdi hücrelerinden gizli katmana geçiren haritaya (map), öznitelik haritası (feature map) denilmektedir. Öznitelik haritasındaki ağırlıklara, ortak ağırlık ve kullanılan biase de ortak bias denilmektedir. Ortak ağırlık ve bias ikisi beraber kernel veya filtre olarak isimlendirilmektedir. Örneğin filtre olarak kenar bulma seçilmişse, kenar bulma filtresi, tüm resimlerde de kenarları bulmaktadır. Eğer öznitelik haritası olarak köpek bulan bir harita seçilmişse, o öznitelik haritasının tek işi, köpekleri bulmaktır.

Ağ mimarisinde, sadece bir tane öznitelik haritası ile resmi tanımlamak mümkün değildir. Resmi tanımlamak için birden çok öznitelik haritasına ihtiyaç duyulmaktadır. Bu yüzden, evrişimsel sinir ağlarının katmanlarında, birden çok öznitelik haritası bulunmaktadır.

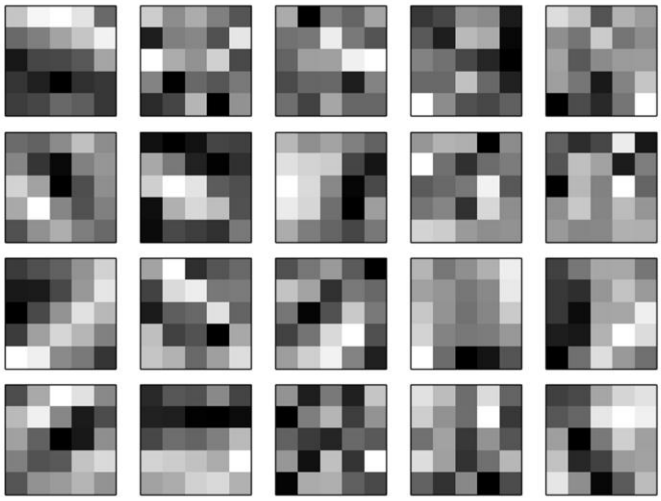
Şekil 3.8'de üç tane öznitelik haritası kullanılmıştır. Her öznitelik haritası 5x5 ortak ağırlık ve bir ortak bias ile tanımlanmıştır. Bu nedenle, bu şekilde gösterilen ağ sadece üç farklı öznitelik tanıyabilmektedir. Bu öznitelikler, tüm görüntülerde tespit edilebilmektedir.



Şekil 3.8 Öznitelik haritası yöntemi ile çıkarılmış 3x20x16 boyutundaki ilk gizli katman

Şekil 3.8'de, anlaşılır olması için üç tane öznitelik haritası gösterilmiştir. Çalışmalarda kullanılan evrişimsel sinir ağlarında, çok daha fazla öznitelik haritası kullanılmaktadır. İlk olarak kullanılan evrişimsel sinir ağlarından olan LeNet-5'de (Lecun vd. 1998) 6 tane 5x5 boyutunda yerel çabuk kavrayan alanlı öznitelik haritası kullanılmıştır. Örnek olarak şekil 3.9'da öğrenilmiş olan öznitelik haritaları gösterilmiştir.

Şekil 3.9'da 20 farklı resim, 20 farklı öznitelik haritasını (filtre veya kernel) simgelemektedir. Her resim, 5x5 pikselden oluşmaktadır. Açık olan pikseller küçük ağırlıkları, koyu olan pikseller ise büyük ağırlıkları göstermektedir.



Şekil 3.9 Yirmi farklı öznitelik haritası örneği (Nielsen 2015)

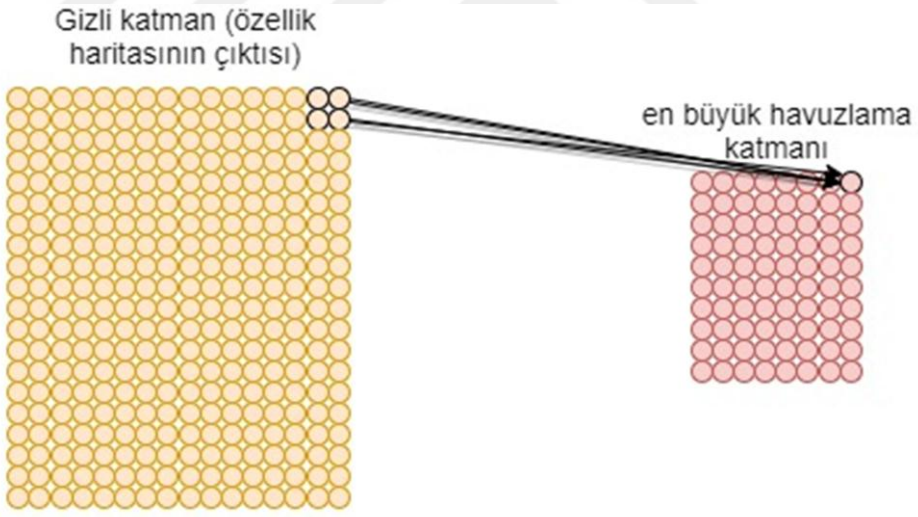
CNN'lerin en büyük avantajı, ortak ağırlıkların ve biasın sağlamakta olduğu parametre sayısını azaltmasıdır. Her öznitelik haritası için 25 = 5 x 5 ortak ağırlık ve 1 tane bias bulunmaktadır. Bu yüzden toplam 26 parametre bulunmaktadır. Eğer 20 tane öznitelik haritası var ise bir evrişimsel katmanında toplam 20 x 26 = 520 adet parametre bulunmaktadır. Buna karşılık tümden bağlı (fully connected) katmanda ise ilk katmanda 784 = 28 x 28 adet girdi hücresi bulunmaktadır. Eğer ikinci katmanda, 30 tane gizli hücre var ise 784 x 30 tane ağırlık ve 30 tane bias bulunmaktadır. Bu parametrelerle beraber toplam parametre sayısı 23550 tane olmaktadır. Bu da göstermektedir ki, tümden bağlı katmandaki parametre sayısı, evrişimsel katmanındaki parametre sayısından 40 kat daha fazladır. Bu sebeple çok katmanlı ağ yapılmaya çalışıldığında, parametre sayısının çok olması, eğitimde sorun yaratmaktadır. CNN'lerde ise çok katmanlı mimari yapı kullanılsa bile, eğitim daha hızlı yapılabilmektedir.

İsim olarak evrişimsel (convolutional) kelimesi, yukarıdaki eşitlikte geçen işlemin "convolutional" olarak isimlendirilmesinden kaynaklanmaktadır. Literatürde eşitlik, $a^1 = \sigma(b + w * a^0)$ olarak yazılmaktadır. Buradaki a^1, bir tane öznitelik haritasının çıktı değerlerini göstermektedir. Parametre a^0, girdi değerlerini ve * simgesi de evrişimsel (convolution) işlemi olarak gösterilmektedir.

3.4.3 Havuz katmanı

Evrişimsel sinir ağlarında, evrişimsel katmanlarına ek olarak havuzlama (pooling) katmanları da bulunmaktadır. Genellikle havuz katmanı, evrişimsel katmandan sonra kullanılmaktadır. Havuz katmanın görevi, evrişimsel katman ile çıkan bilgisinin boyutunu azaltmaktır. Bir havuz katmanı, evrişimsel katmanda oluşmuş olan bütün öznitelik haritalarının çıktısını, yoğunlaştırmaktadır.

Örneğin, havuz katmanındaki her eleman, bir önceki katmandaki 2x2'lik bir bölgedeki hücreleri özetlemektedir. Yaygın olarak kullanılmakta olan havuzlama yöntemi, maksimum havuzlamadır (max-pooling). Maksimum havuzlamada, 2x2 boyutundaki girdi hücrelerinden maksimum olanı seçilerek bir sonraki katmana geçmektedir. Şekil 3.10'da havuzlama katmanı gösterilmiştir.



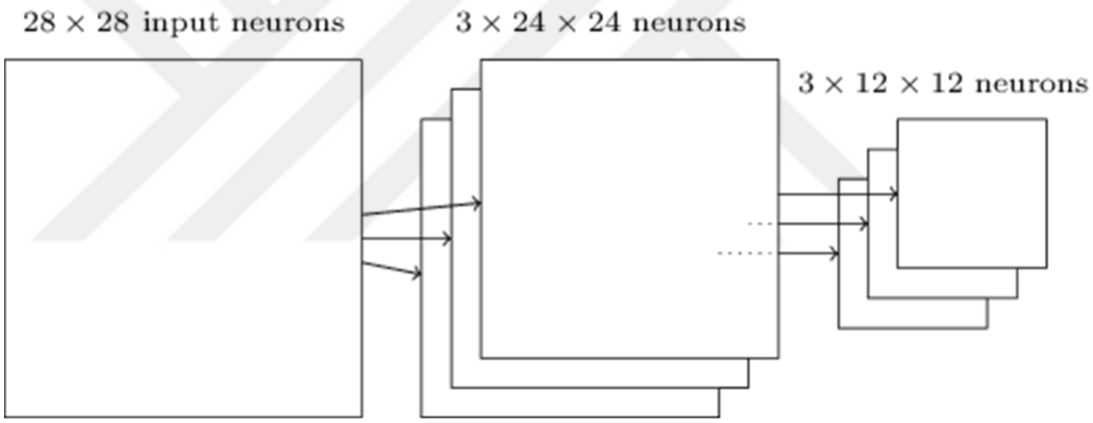
Şekil 3.10 Maksimum havuzlama işlemi ile oluşmuş 10x8 boyutundaki katman

Eğer 20x16 boyutunda bir evrişimsel katman varsa, 2x2 boyutundaki havuzlama işleminden sonra (kaydırma miktarı iki olarak seçildiğinde) 10x8 boyutunda bir katman oluşacaktır.

Evrişimsel katmanda, bir öznitelik haritası yerine, birden fazla öznitelik haritası

kullanılmaktadır. Havuzlama işlemi, her bir öznitelik haritasına ayrı ayrı uygulanmaktadır. Eğer üç adet öznitelik haritası var ise, evrişimsel katmanı ve maksimum havuz katmanı şekil 3.11'deki gibi gösterilmektedir.

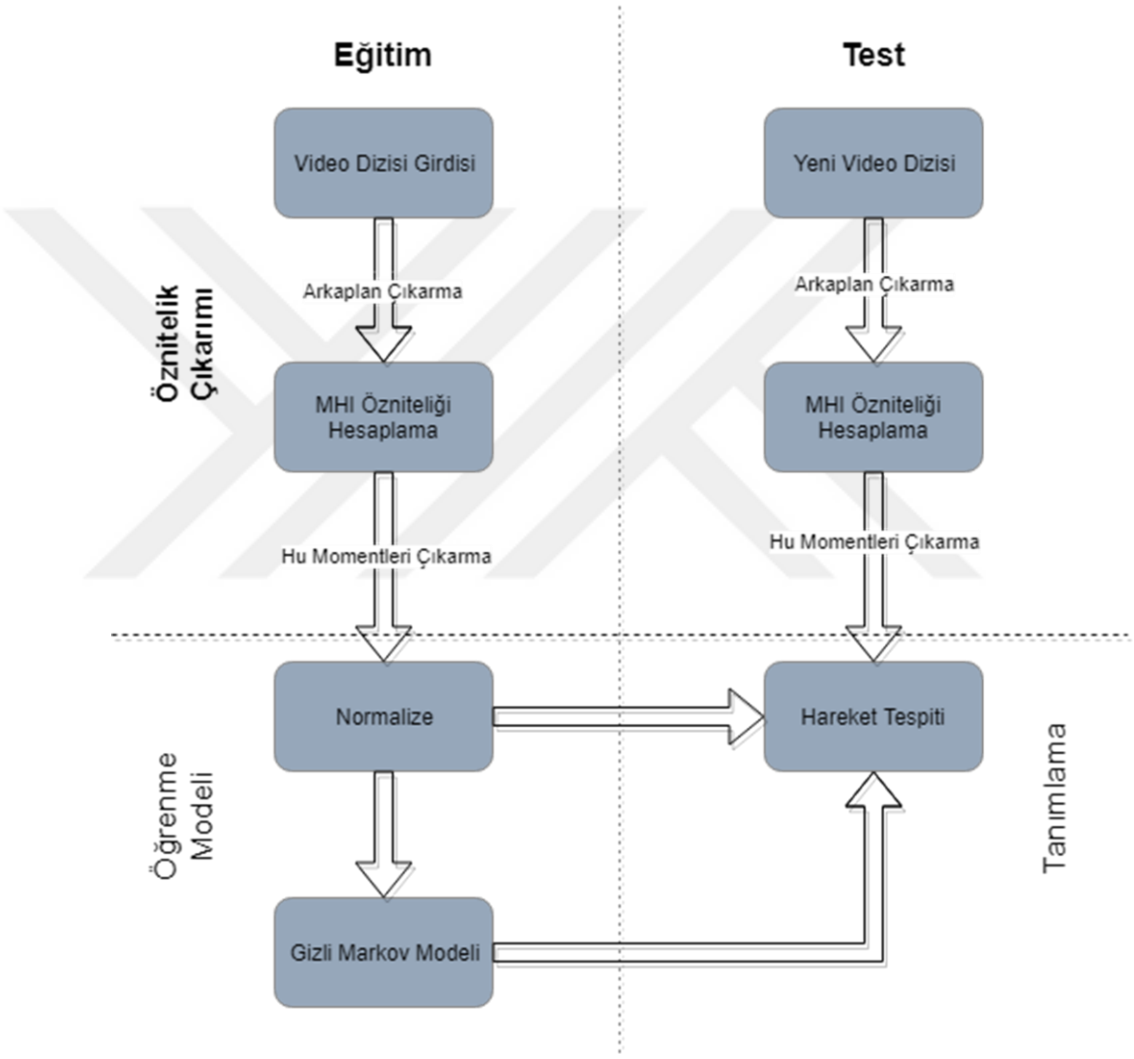
Havuzlama tekniklerinden, maksimum havuzlamadan başka kullanılan teknik L2 havuzlamasıdır. 2x2 boyutundaki havuzlama alanından, piksellerin kareleri alınıp, topladıktan sonra, karekökü alınarak havuzlama işlemi gerçekleştirilmektedir. Bu işleme, karekök yöntemi denilmektedir. Her iki havuzlama yönteminde de detayları farklı olsa da amaçları aynıdır. Maksimum havuzlamada olduğu gibi L2 havuzlama da, evrişimsel katmandaki bilgiyi yoğunlaştırmaktadır. Havuzlama tekniği aynı zamanda öznitelikteki uzamsal kaymalara karşı gürbüz algılama sağlamaktadır.



Şekil 3.11 Evrişimsel ve havuzlama katmanından oluşmuş bir CNN (Nielsen 2015)

4. HAREKET SINIFLANDIRILMASININ MODELLEMESİ

Videolardaki aktivitelerin sınıflandırmasının yapılabilmesi için bu çalışmada çeşitli yöntemler kullanılmıştır. Kullanılan yöntemlerden biri HMM olup aktivitenin modellenmesi şekil 4.1'de gösterilmektedir.



Şekil 4.1 MHI-Hu moment öznitelikleri ile HMM aktivite sınıflandırma sistemi

Modelin eğitim aşamasının sözde kodu (pseudocode) şu şekildedir:

```
procedure trainHMM(trainVideos, categories, scalingMethod)
    hmmModels = {}
```

```
    scalingModels = {}
    for each category in categories do
        frames = []
        hmmModels[category] = new HMM()
        scalingModels[category] = scalingMethod
        for each video in trainVideos[category] do
            for each frame in video do
                frame = calculateBackground(frame)
                mhi = calculateMMHI(frame)
                hu = calculateHu(mhi)
                frames.append(hu)
            end for
            scalingModels[category].fit(frames)
            scaledImages = scalingModels[category].transform(frames)
            hmmModels[category].fit(scaledImages)
        end for
    end for
    return hmmModels, scalingModels
end procedure
```

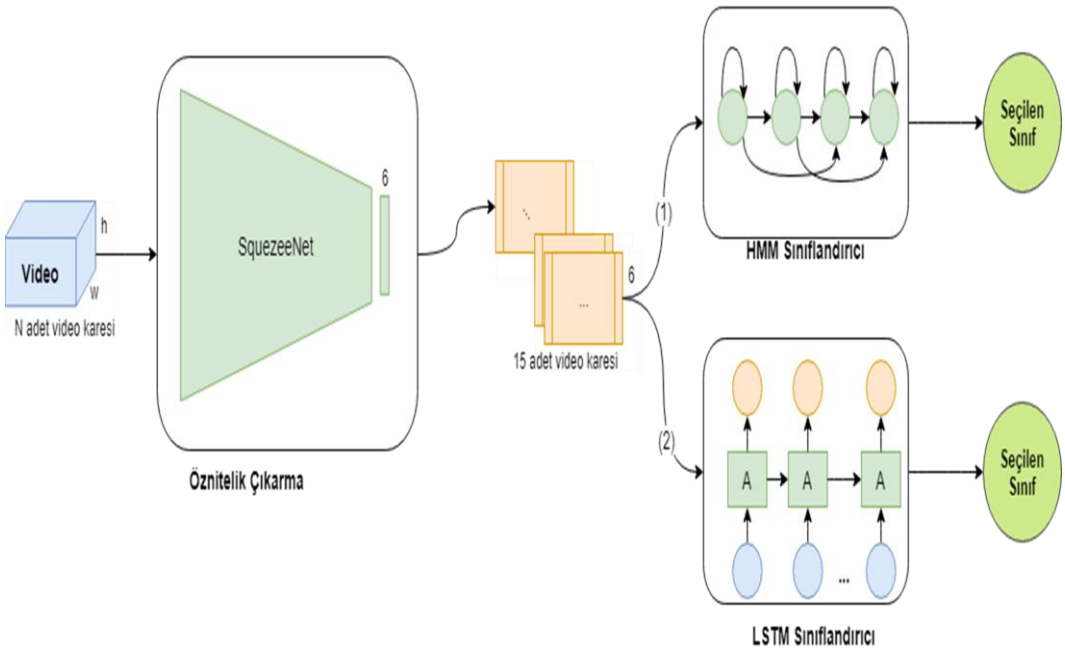
Video sınıflandırılmasında kullanılan yöntemin test aşamasının sözde kodu şu şekildedir:

```
procedure testHMM(testVideos, categories, hmmModels, scalingModels, windowSize)
    expectedClasses = []
    predictedClasses = []
    for each category in categories do
        for each video in testVideos[category] do
            frames = []
            for each frame in video do
                frame = calculateBackground(frame)
                mhi = calculateMMHI(frame)
                hu = calculateHu(mhi)
                frames.append(hu)
            end for
            for each index in range(frames.length – windowSize) do
                videoClip = frames[index : index + windowSize]
                max = -∞
                for each predictCategory in categories do
                    videoClip =scalingModel[predictCategory].transform(videoClip)
                    score = hmmModels[predictCategory].score(videoClip)
                    if score > max then
                        max = score
                        predictedClass = predictCategory
```

```
            end if
        end for
        predictedClasses.append(predictedClass)
        expectedClasses.append(category)
    end for
  end for
end for
calculateMetrics(predictedClasses, expectedClasses )
end procedure
```

Derin ağlar kullanılarak yapılan deneylerin, özeti şekil 4.2'de gösterilmektedir. Derin ağlardan biri olan CNN seçilmiştir. Sıfırdan bir model eğitmek yerine, daha önceden eğitilmiş olan modeller kullanılmıştır. Eğitilmiş model olarak, VGG-16 (Simonyan ve Zisserman 2014) seçilmiş olup bazı deneyler bu model ile yapılmıştır. Fakat VGG-16 mimarisinde, eğitim yapmak çok uzun zaman almaktadır. VGG-16 modelini seçmek yerine, daha küçük ve hızlı olduğundan daha iyi sonuçlar veren SquezeeNet (Iandola vd. 2016) modeli deneylerde kullanılmıştır. Bu modeli seçerek, eğitimde ve testte geçen süre 6 kat daha hızlı olması sağlanmıştır.

CNN ile öznitelikler çıkarıldıktan sonra, LSTM veya HMM seçilerek aktivite sınıflandırılması yapılmıştır. Sistemin özeti şekil 4.2'de gösterilmektedir



Şekil 4.2 CNN öznitelikleri ile HMM veya LSTM ile sınıflandırma sistemi

CNN eğitim aşamasındaki sözde kod şu şekildedir:

```
procedure trainCNN(trainVideos, categories)
    cnnModel = getNotIncludeTopNotTrainableSquuezeNet()
    cnnModel.addLayer(dropOut(0.5))
    cnnModel.addLayer(convolution2D(categories.length))
    cnnModel.addLayer(activation('relu'))
    cnnModel.addLayer(globalAvaragePooling2D())
    cnnModel.addLayer(activation('softmax'))
    x = []
    y = []
    for each category in categories do
        for each video in trainVideos[category] do
            for each frame in video do
                x.append(frame)
                y.append(category)
            end for
        end for
    end for
    cnnModel.fit(x, y)
    return cnnModel,
end procedure
```

CNN'den çıkarılan özelliklerin HMM eğitiminin sözde kodları şu şekildedir:

```
procedure trainHMM(trainVideos, categories, scalingMethod, cnnModel)
    hmmModels = {}
    scalingModels = {}
    for each category in categories do
        frames = []
        hmmModels[category] = new HMM()
        scalingModels[category] = scalingMethod
        for each video in trainVideos[category] do
            for each frame in video do
                frames.append(frame)
            end for
            features = cnnModel.predict(frames)
            scalingModels[category].fit(features)
            scaledImages = scalingModels[category].transform(features)
            hmmModels[category].fit(scaledImages)
        end for
    end for
    return hmmModels, scalingModels
```

```
end procedure
```

CNN özelliklerinin, HMM'de test edilmesinde kullanılan sözde kod şu şekildedir:

```
procedure testHMM(testVideos, categories, hmmModels, scalingModels, windowSize)
    expectedClasses = []
    predictedClasses = []
    for each category in categories do
        for each video in testVideos[category] do
            for each frame in video do
                features = cnnModel.predict(frames)
                frames.append(hu)
            end for
            for each index in range(frames.length – windowSize) do
                videoClip = frames[index : index + windowSize]
                max = -∞
                for each predictCategory in categories do
                    videoClip =scalingModel[predictCategory].transform(videoClip)
                    score = hmmModels[predictCategory].score(videoClip)
                    if score > max then
                        max = score
                        predictedClass = predictCategory
                    end if
                end for
                predictedClasses.append(predictedClass)
                expectedClasses.append(category)
            end for
        end for
    end for
    calculateMetrics(predictedClasses, expectedClasses )
end procedure
```

CNN'den çıkarılan özelliklerin LSTM eğitimi ve testinde kullanılan sözde kodları şu şekildedir:

```
procedure trainTestLSTM(trainVideos, testVideos, categories, windowSize)
    train_x = []
    train_y = []
    test_x = []
    test_x = []
    lstmModel = createLSTMModel()
```

```
    for each category in categories do
        for each video in trainVideos[category] do
            features = cnnModel.predict(frames)
            for each index in range(video.frames.length – windowSize) do
                train_x.append(cnnModel.predict (video.frames[index : index + windowSize]))
                train_y.append(category)
            end for
        end for
        for each video in testVideos[category] do
            features = cnnModel.predict(frames)
            for each index in range(video.frames.length – windowSize) do
                test_x.append(cnnModel.predict (video.frames[index : index + windowSize]))
                test_y.append(category)
            end for
        end for
    end for
lstmModel.fit(train_x, train_y, test_x, test_y)
end procedure
```

4.1 Gizli Markov Modelleri

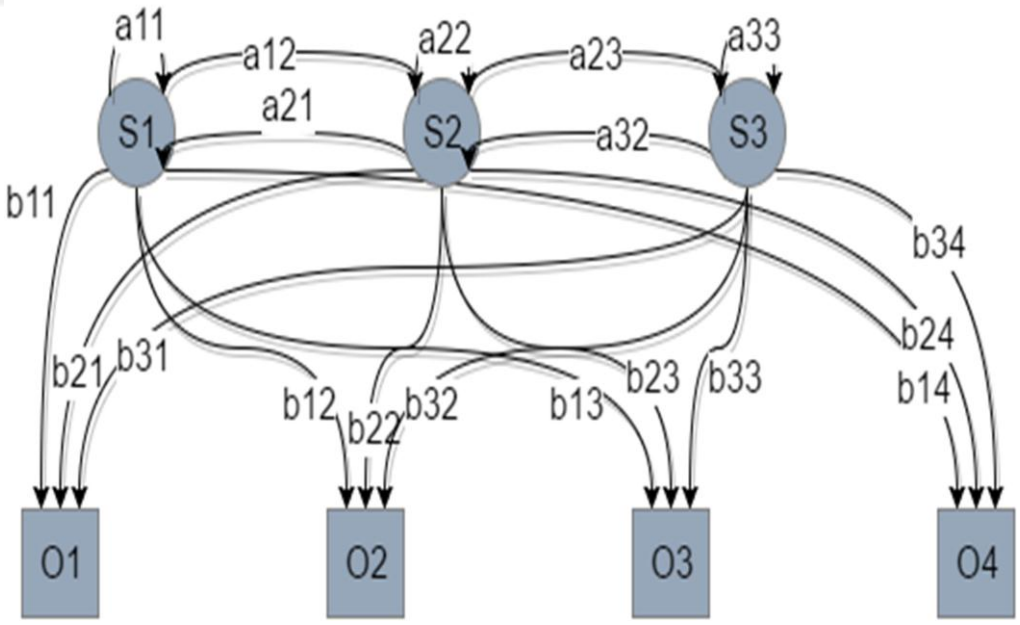
Gizli markov modeller, zamandaki akışı modellemeye uygundur. Ses işlemede yaygın olarak kullanılmakta ve iyi sonuçlar alınmaktadır. Videodan, insan aktivitelerinin sınıflandırılması için uygun olduğu pek çok çalışmayla gösterilmiştir; bu çalışmalardan ilki Yamato vd. (1992)'nin çalışmasıdır.

HMM ile sınıflandırmada kullanılmakta olan algoritma şu şekildedir: Videodaki her aktivite sınıfı, ayrı ayrı Baum-Welch algoritması (Baum 1972) kullanılarak eğitilerek modellenmektedir. Test aşamasında, videolar, Viterbi algoritması (Viterbi 1967) ile aktiviteye olan yakınlığı (log-likelihood) ölçülmektedir. En yüksek sonuç alınan aktivite, HMM sınıflandırılmasından seçilen aktivite olmaktadır.

4.1.1 Tanım

Klasik HMM, şekil 4.3'de gösterilmektedir. Gizli markov modelleri aşağıdaki parametrelerle tanımlanmaktadır.

- N durum sayısıdır. $S = \{S_1, S_2, S_3, \dots, S_N\}$ ve t anındaki gizli durum da, $q_t \in S = \{ s_1, s_2, s_3, \dots., s_N\}$ şeklinde gösterilmektedir. (Rabiner 1990)
- π, başlangıç durumu olasılıkları olarak $\pi = \{\pi_1, \pi_2, \pi_3, \dots, \pi_N\}$ şeklinde gösterilmektedir. Burada π_i, s_i durumunun başlangıç olasılığını simgelemektedir.
- A simgesi, durumlar arasındaki geçiş olasılıkları olup $A = (a_{ij})_{N*N}$ ile gösterilmektedir. Burada $a_{ij} = p\big(q_{t+1} = s_j | q_t = s_i\big)(1 \le i, j \le N)$, a_{ij}, t zamanından a_i durumundan, t+1 zamanına a_j durumuna geçişin olasılığını göstermektedir.
- B, emisyon olasılık dağılımını göstermektedir. $B = \{b_i(o_t)\}$, olarak da simgelenmektedir. $b_i(o_t) = p(o_t|s_t)(1 \le i \le N), b_i(o_t)$, t zamanındaki, s_i durumundaki o_t gözlem sembolünün olasılığıdır.
- T, dizideki gözlem sembolü sayısıdır. Bu dizide 1'den T'ye kadar olan gözlem sembolleri $O = \{o_1, o_2, o_3, \dots., o_T\}$ olarak gösterilmektedir.



Şekil 4.3 Basit bir gizli markov modeli (HMM)

Bir gizli markov modeli, λ = {A, B, π} olarak gösterilmektedir. Burada, π gömülü skolastik süreci, A markov zincirini, B ise gözlem sembolleri ve durumlar arasındaki ilişkiyi tanımlamaktadır.

Her durumdaki, bir gözlem simgesi üretme olasılığı, Gaussian olasılık yoğunluk fonksiyonu ile hesaplanmaktadır. Eşitlik (4.1)'de bu olasılığın hesaplanması gösterilmektedir.

$$b_i(o_t) = \frac{1}{\sqrt{(2\Pi)^d|\Sigma_i|}} \exp(-\frac{1}{2}(o_t - \mu_i)^T \Sigma_i^{-1}(o_t - \mu_i)) \quad (4.1)$$

Eşitlikte μ_i, Σ_i sırasıyla, gözlemlerin ortalama vektörü ve kovaryans matrisleridir; T operatörü bir matrisin transpozunu almaktadır; Σ_i^{-1}, Σ_i'nin tersidir; d, gözlem sembolü o_t'nin boyutudur.

4.1.2 Tanıma

$\lambda^1, \lambda^2, \lambda^3, \ldots., \lambda^N$sırasıyla act1, act2, act3, ..., actN'nin gizli markov modelleri olduğunu varsayarsak, $O = \{o_1, o_2, o_3, \ldots., o_N\}$ gözlem dizisi olmaktadır. Her verilen eğitilmiş aktivite modeli için ileriye ve geriye (forward and backward) algoritması kullanılarak, test dizisinin olasılığı hesaplanmaktadır. (Yamato vd. 1992)

$$p(O|\lambda^1), p(O|\lambda^2), p(O|\lambda^3), \ldots, p(O|\lambda^N) \quad (4.2)$$

Bir aktiviteye karşılık gelen maksimum (likelihood) olasılık değeri en iyi aktivite olarak seçilmektedir. Eşitlik (4.3)'de en iyi aktivitenin nasıl seçildiği gösterilmektedir.

$$\text{test_numarası} = \underset{1 \le n \le N}{\text{argmax}}(p(O|\lambda^N) \quad (4.3)$$

4.1.3 Eğitim

Daha önceki bölümlerde belirtildiği gibi, HMM'i eğitmek için Baum ve Welch algoritması (Baum 1972) kullanılmaktadır. Gizli markov modellerin tanımında, açıklanan parametrelerin başlangıç değerleri, rastgele veya eşit olarak tanımlanmaktadır. Deneylerde kullanılmakta olan HMM'lerde, modelin ilk değerlerini bulmak için K-Means algoritması (Lloyd 1982) kullanılmaktadır. K-Means algoritmasında, parametreleri güncellemek için veri ve küme ortalaması arasındaki Öklid uzaklığını kullanılmaktadır. Öklid uzaklığı Eşitlik (4.1)'de gösterilmiştir.

$$d_i(o_t) = \sqrt{(o_t - u_i)^T (o_t - u_i)} \tag{4.4}$$

Eşitlik (4.1)'de, d_i kümedeki i gözlem (o_t) ve ortalama (u_i) arasındaki uzaklıktır. Tespit edilen küme merkezleri, HMM'in başlangıç değerleri olarak kullanılmaktadır. Her bir aktivite modelini eğitmek için Baum-Welch algoritması (Baum 1972) kullanılmaktadır. Eğitim aşamasında, kümeli üretimden kaynaklanan küçük değerler (0'a yakın değerler), probleminin çözülmesi ve yetersizlikten kaçınmak için, ölçek faktörü metodu (Yamato vd. 1992) kullanılmıştır. Bu yöntem, bu sorunların çözümüne yardımcı olmaktadır.

4.2 Tekrarlayan Yapay Sinir Ağları

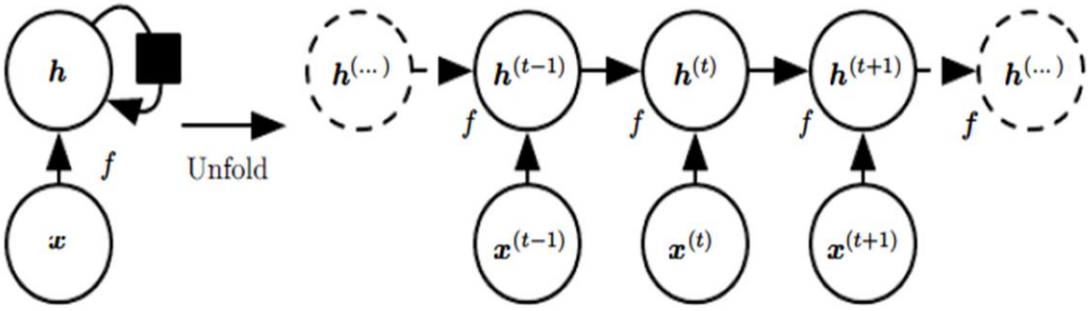
Tekrarlayan yapay sinir ağları (RNN), birimler arasındaki bağlantıların, yönlendirilmiş bir döngü oluşturduğu ağlardır. RNN ile dinamik zamansal davranış sergilemesine izin verilmektedir. İleri beslemeli sinir ağlarından farklı olarak, RNN'ler kendi giriş belleklerini, girdileri işlemek için kullanabilirler. Bu öznitelik RNN'leri, el yazısı tanıma ve konuşma tanımada, kullanılabilir bir yöntem yapmaktadır.

İnsanlar, yeni öğrendikleri bir sözcük için yeni anlamlar öğrenmezler. Daha önceden var olan benzer sözcüklerden yola çıkarak yeni öğrendikleri sözcüğe anlam yüklerler. Ancak geleneksel yapay sinir ağlarında insanlarda bulunan bu anlamlandırma özniteliği bulunmaz ve bu onların en büyük eksikliğidir. Örneğin, videodaki tüm karelere bakarak

aktiviteler sınıflandırmak istendiğinde, geleneksel sinir ağları kareler arasında insanlar gibi anlamlandırma kuramadığından, sınıflandırma yapamayacaktır.

Tekrarlayan yapay sinir ağları ise bir döngü oluşturarak, geçmiş bilgilerin kullanılmasını sağlanır ve böylelikle kareler arasında anlamlandırma yaparak sınıflandırma yapması sağlanır.

Şekil 4.4'de basit bir çıktısı olmayan tekrarlayan sinir ağı görülmektedir. X değeri h durumlu hücreye girdi oluşturmaktadır. Hücre durumu, bilgileri bir sonraki zaman diliminde ileriye doğru taşıma işlemini siyah dikdörtgen ile yapmaktadır. Hücreden çıkan bir değer yine kendisine gelerek, bir döngü oluşturmaktadır. Bu döngü ile önceki zamanın bilgileri de kullanılabildiğinden yeni bilgi, eski bilgi kullanılarak anlamlandırılabilmekte ve böylelikle sınıflandırma yapılabilmektedir.



Şekil 4.4 Çıktısı olmayan bir RNN (Goodfellow vd. 2016)

Geleneksel yapay sinir ağlarında, hücrelerden çıkan sonuçlar tekrardan kendilerine girdi olarak gelmemektedir. RNN'de ise hücreden çıkan sonuç, tekrardan kendisine girdi olarak gelmektedir. RNN açılırsa şekil 4.4'ün sağ tarafındaki gibi bir mimari ortaya çıkmaktadır. Zaman diliminde, aynı hücre kendini birden fazla tekrar etmektedir. Böylelikle de kareler arasında anlamlandırma kurulabilmektedir.

RNN'ler, bir döngü oluşturabildiklerinden, sıralı gelişen aktiviteleri birbirleriyle anlamlandırabilmektedir. Akış içerisindeki aktivitelerin anlamlandırılarak

sınıflandırabilmesinden dolayı son yıllarda yaygın olarak kullanılmaktadır. RNN'lerin, birçok kullanım alanı bulunmaktadır; konuşma tanımlama, dil modelleme, dil çevirimi, resim başlığı oluşturma vb.

4.2.1 Tekrarlayan yapay sinir ağlarının problemleri

RNN'ler, geçmiş ile bağlantı kurup anlamlandırma özniteliğinden dolayı, zaman bazlı problemlerde başarılı sonuçlar vermektedir. Ancak RNN'lerde, hangi aktiviteler hatırlanacak, ne kadar süre hatırlanacak bilinmemektedir. Bütün bilgiler, modelin içerisinde tutulmaktadır. Aktiviteler için bazı bilgiler önemliyken, bazı bilgiler gereksizdir. Bu yüzden bazı aktivitelerin sınıflandırılmasında, tüm geçmişin saklanmasına gerek yoktur.

Aktivite sınıflandırılmasında, gerekli bilgi çok önceden oluşmuş ise, bu bilgiye ulaşılamayabilir. Bazı durumlarda ise, bir önceki karedeki olay ile şimdiki karedeki olay birbiri ile bağlantılıdır. Örneğin, aktivite sınıflandırılması yapılacak olan videoda, sırasıyla bazı kişiler yemek masasına oturuyor olsun; böyle bir videoda, önceden masaya oturmuş kişilerden, bir sonraki gelen kişinin de masaya oturacağını tahmin etmek RNN için zor değildir. Çünkü bir sonraki karenin tahmin edilebilmesi için gerekli olan masaya oturma aktivitesi, çok yakın zamanda gerçekleşmiştir.

Bazı aktiviteler, yemek masasına oturma aktivitesinden çok daha karmaşıktır. Örneğin, sınıflandırma yapılacak video, yemek masasında oturulup yemeğe başlandıktan sonra gelen bir kişinin aktivitesi olsun. Yemeğe geç gelen kişinin, yemeğe oturacağının tahminin yapılması insanlar için zor değildir. Fakat RNN için yemeğe oturma aktivitesi bitip, aralara başka aktiviteler girdiği için, yeni kişinin yemeğe oturacağını tahmin etmek zor olmaktadır.

RNN'lerin çok önceden olan olayları tahmin edebilmesi için farklı bir mimari yapıya ihtiyaçları vardır. Bu tür problemlerde, daha iyi çalışan RNN'lerin, bir çeşidi olan uzun kısa süreli bellek ağları kullanılmaktadır.

Teoride, RNN'lerin uzun geçmişteki aktiviteleri, iç mimarisinde kendini tekrarlamakta olduğu için, hatırlayabilme kapasitesine sahiptir. Ancak hatırlanabilmesi için parametrelerin titizlikle seçilmesi gerekmektedir. Böyle bir parametre seçimi, pratikte mümkün olmadığından, RNN'ler uzun geçmişi öğrenemezler (hatırlayamazlar).

LSTM ağlarında böyle bir problem bulunmamaktadır. Bu yüzden LSTM'ler birçok aktivite sınıflandırılmasında çoğunlukla tercih edilmektedir. Son zamanlardaki RNN'ler ile yapılmış başarılı çalışmaların, büyük bir çoğunluğunda, LSTM'ler kullanılmıştır.

4.2.2 Uzun kısa süreli bellek ağları

Uzun kısa süreli bellek ağları, RNN'lerin uzun süreli hatırlama problemlerini çözmek amacıyla ortaya çıkmıştır. Genellikle LSTM olarak kısaltılmaktadır.

Geleneksel RNN'lerde, tekrarlayan kısımda tek bir tanjant fonksiyonu (katmanı) bulunmaktadır. LSTM'ler tek bir sinir katmanı yerine tekrarlayan zincir şeklinde dört farklı katman bulunmaktadır.

4.2.2.1 Temel

LTSM'ler, kapılar (gate) yardımıyla, hücre durumundan, bilgi silme ve hücre durumuna bilgi ekleme yeteneğine sahiptir. Bu kapılar, isteğe bağlı olarak bilginin geçmesine olanak sağlamaktadır.

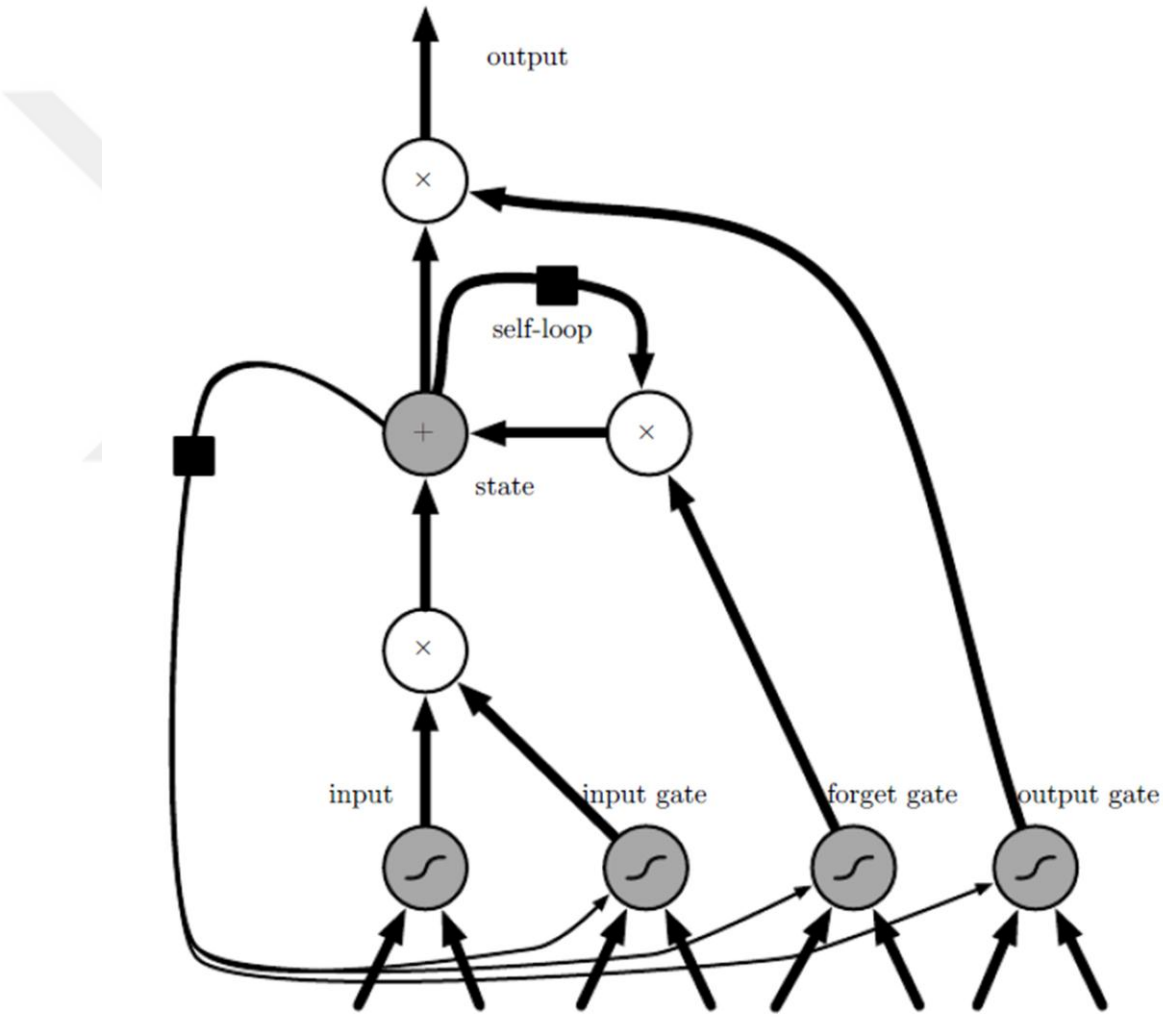
Sigmoid sinir katmanı, bilginin, ne kadar geçeceğini belirlemektedir. Sigmoid fonksiyonu, gelen veriye göre, çıktıda 0 ile 1 arasında bir sonuç çıkarmaktadır. Bir sigmoid fonksiyonundan, 0 çıkmışsa, "hiçbir şeyin geçmesine izin vermeyin", eğer 1 sonucu çıkmışsa "hepsinin geçmesine izin verin" anlamındadır.

LSTM ağları, bu kapılardan üç tanesine sahiptir. Bu kapıların görevi, korumak ve kontrol etmektir.

4.2.2.2 Aşamaları

LSTM ağlarının, sahip olduğu bazı kapılar vardır. Bu kapılar şu şekilde listelenir;

- Giriş Kapısı
- Unutma Kapısı
- Çıkış Kapısı



Şekil 4.5 LSTM'deki tek bir hücrenin modeli (Goodfellow vd. 2016)

LSTM ağının ilk aşaması, unutma kapısı (forget gate) ile yapılmakta olan, hücre durumundaki bilginin, unutulup unutulmayacağına (saklanıp saklanmayacağının) karar

verilmesi aşamasıdır. Bu aşamada unutma kapısı, bir önceki durumu girdi olarak alıp, 0 ile 1 arasında bir sayı vermektedir. Eğer 1 çıkarsa tamamen bunu tut, 0 çıkarsa bunu tutma anlamındadır.

Yemeğe geç gelen insan örneğinde, masa ile insan arasında aktivite bulunmaktaydı. Eğer, bu aktivite LSTM hücrelerinin durumunda bulunuyorsa, sonraki gelen verilerde, bu hücre durumundaki aktiviteye uygun olarak tahmin yapabileceklerdir. Eğer yeni bir aktiviteye geçilmişse, bu masa ve insan aktivitesinin hücre durumundan çıkarılması (unutulması) gerekmektedir.

LSTM'lerin bir sonraki aşaması, giriş kapısı (input gate) ile yapılan hangi bilginin hücrede tutulacağı bilgisinin belirlenmesidir. Giriş kapısı, sigmoid sinir ağı ve tanjant katmanı olmak üzere ikiye ayrılmaktadır. Sigmoid sinir katmanında, hangi değerlerin güncelleneceğine karar verilirken, tanjant katmanda ise saklanması gereken yeni vektör oluşturulmaktadır. Giriş katmanında, iki farklı değer birleşerek, yeni saklanacak olan değeri oluşturmaktadır.

Örnekte, masaya oturacak kişinin, unutulup unutulmayacağına karar verildikten sonra, yeni bir aktivite geldiğinde önceki aktivitenin ne kadarının bellekte tutulacağına, giriş kapısı katmanı ile birlikte karar verilmektedir.

Çıkış katmanı (output gate), hücre durumundan çıkan sonuca doğrudan etki etmektedir. Bu katmandan çıkan sonuç ile hücre durumu tanjant fonksiyonunda geçirilip çarpılarak hücrenin çıktı değeri oluşmaktadır.

5. YAPILAN DENEYLER

Aktivite sınıflandırma deneylerinde, kamuya açık olan KTH ve Weizmann veri kümeleri kullanılmıştır. KTH veri kümesi, Weizmann veri kümesine göre daha çok videoya sahiptir; buna rağmen farklı kişilerle farklı varyasyon verileri oluşturduğu için sınıflandırılması Weizmann veri kümesine göre daha zordur. Ayrıca, KTH verilerinde, video arka planı sabit olmaması da sınıflandırılmasını zorlaştırmaktadır.

Deneylerde, aktivitenin sınıflandırılması için LSTM ve HMM ayrı ayrı kullanılmış ve karşılaştırılmıştır. Öznitelik olarak; MHI, Değiştirilmiş MHI, Hu momentleri ve CNN yöntemleri kullanılmıştır.

Geliştirme ortamında Python dili kullanılmıştır. Python, script olması ve çok geniş kütüphanelere sahip almasından dolayı bu dil seçilmiştir. HMM eğitiminde ve testinde kütüphane olarak HMMLearn kullanılmıştır. Görüntü işleme yöntemleri için OpenCV kullanılmıştır. CNN ve LSTM ile yapılan deneylerde, kütüphane olarak Keras ve Tensorflow kullanılmıştır. CNN eğitimi çok uzun sürdüğü için Google'ın hizmet olarak sunmakta olduğu bulut bilgisayarlar kullanılmıştır. Bu bulut makinasına, ekran kartı da eklenerek CNN eğitimi, paralel olarak yapılması sağlanmıştır. Sonraki bölümlerde, veri kümesi üzerinde, farklı farklı öznitelikler kullanılarak, LSTM ve HMM ile nasıl sınıflandırma yapıldığı anlatılmaktadır.

5.1 Weizmann Veri Kümesiyle Yapılan Deneyler

Weizmann veri kümesinde, 93 adet düşük çözünürlüklü video bulunmaktadır. Videolar, 180x144 boyutundadır. Veri kümesindeki videolar, dokuz farklı insan kullanılarak oluşturulmuştur. Weizmann veri kümesinde, insan aktivitelerini sınıflandırılması yapılırken; eğilme, zıplama, tek ayakla zıplama, koşma, yan yan gitme, yürüme, tek el sallama, çift el sallama, elleri ve ayakları açıp kapama, zıplayarak ilerleme olmak üzere 10 adet sınıf bulunmaktadır.

El ile özniteliklerin belirlendiği bu deneylerde, kullanılan videolar siluet videolarıdır. Bu siluet videoları veri kümesiyle beraber gelmektedir. Siluet videoları, siyah beyaz resimlerden oluşmakta olup, insan figürünün hareket ettiği yer beyaz olarak gösterilmiştir. Bu sayede, videoda herhangi bir ön işleme gerek duyulmamıştır.

Öğrenebilen özniteliklerin kullanıldığı deneylerde ise, orijinal (renkli) video veri kümesi kullanılmıştır. Daha önceden renkli videolar ile eğitilmiş olan VGG-16 ve SquezeeNet modelleri kullanıldığı için siyah beyaz olan siluet videolarına gerek duyulmamıştır.

Weizmann veri kümesinde, az sayıda veri olduğu için leave-one-out çapraz geçerleme (cross-validation) yöntemi kullanılmıştır. Bu yöntemde, test işlemi başlamadan önce kullanılacak videolardan bir tanesi test verisi olarak, kalan videolar da eğitim verisi olarak ayrılır. Bu test işlemi tüm videolar test verisi olana kadar tekrarlanmaktadır. Tekrarlanan her test için sonuçlar tek tek kaydedilir. Kaydedilen sonuçların aritmetik ortalaması alınarak sınıflandırma sonucu bulunmaktadır. Bu sayede, veriler olabildiğince eğitim verisi olarak kullanılmış ve test verisinde kullanılması sağlanmıştır. Şekil 5.1'de Weizmann veri kümesinden örnek kareler gösterilmiştir.

Şekil 5.1'de, ilk satırdaki küçük resimler soldan sağa doğru açıklanırsa; iki el ve ayak çırpma, eğilme ve zıplayarak gitme. İkinci satırdaki resimler; koşma, iki el çırpma ve yan yan girme. Son satırdaki resimler; yürüme, tek el çırpma ve eğilme aktivitelerine aittir.

Weizmann veri kümesi, az sayıda veriye sahip olduğu için LSTM deneylerinde kullanılmamıştır. Weizmann veri kümesinde kullanılan öznitelikler şunlardır; (1) siluet bulunduktan sonra Hu momentleriyle bulunan öznitelikler, (2) değiştirilmiş hareket geçmişi bulunduktan sonra Hu momentleriyle bulunan öznitelikler, (3) siluet bulunduktan sonra izdüşümü alınmasıyla bulunan öznitelikler, (4) evrişimsel sinir ağıyla bulunan özniteliklerdir.

Bundan sonraki bölümlerde, Weizmann veri kümesi ile yapılan deneyler açıklanacaktır.



Şekil 5.1 Weizmann veri kümesinden birkaç örnek çerçeve

5.1.1 Siluet ve projeksiyon ile HMM

Bu deneylerde, veri kümesinin sağlamış olduğu siluetler kullanılmıştır. Bu kümenin videoları siyah beyaz olarak bulunmaktadır. Siluetin bulunduğu yerler beyaz arka plan siyahtır.

Videodaki kareler, ilk başta küçültme işlemi uygulanıp, boyutları 10'da 1 oranında küçültülmüştür. Küçültme işleminin ardından, izdüşüm (projeksiyon) hesaplanmıştır.

Çizelge 5.1 Weizmann'da, video kliplerin, küçültme ve iz düşüm özniteliklerinin, HMM

Durum Sayısı	Öznitelik Ön İşleme Yöntemi	HMM Tipi	Başarı Oranı %
4	Normalizasyon	Ergodik	88.44
4	ICA	Ergodik	80.31
4	PCA	Ergodik	77.19
4	Normalizasyon	Soldan Sağa	56.91
4	PCA	Soldan Sağa	58.77
6	ICA	Ergodik	84.06
6	Normalizasyon	Ergodik	79.65
6	PCA	Ergodik	46.65
6	Normalizasyon	Soldan Sağa	51.86
6	PCA	Soldan Sağa	58.75
9	PCA	Ergodik	83.43
9	ICA	Ergodik	80.68
9	Normalizasyon	Ergodik	77.42
9	Normalizasyon	Soldan Sağa	54.41
9	PCA	Soldan Sağa	60.27

Çerçeve içeriği matris olarak düşünüldüğünde, sütunlardaki ve satırlardaki değerlerin toplamı olarak hesaplanma işlemine iz düşüm denilmektedir. Bu iz düşümü, resmin özetini vermektedir. İz düşüm öznitelikleri, ICA (Independent Component Analysis), normalizasyon ve PCA (Principal Component Analysis) ile ön işleme süreçlerinden geçirilmiştir. Bu öznitelikler, soldan sağa (left to right) ve ergodik (ergodic) HMM'lerde girdi olarak kullanılmıştır. Sonuçlar tabloda gösterilmiştir. HMM'in eğitim aşamasında videoların tümü, test aşamasında ise 15 karelik video klipleri kullanılmıştır.

Ön işleme yöntemi normalizasyon, durum sayısı 4 ve HMM tipi olarak da ergodik kullanılarak % 88.44 başarı oranı elde edilerek, en iyi sonuç alınmıştır. Soldan sağa HMM tipi, ergodik HMM tipine göre daha kötü sonuç vermiştir. Ergodik HMM'de, normalizasyon ile ön işleme, diğer ön işleme yöntemlerinden daha iyiyken, soldan sağa HMM kullanıldığında, PCA ve ICA ile ön işleme, normalizasyon ile ön işlemeden daha

iyi sonuç vermiştir.

Çizelge 5.2'de, videonun boyutunda, herhangi bir değişiklik yapılmamıştır. Weizmann veri kümesindeki siluet videolarına uygulanmış olan iz düşüm ile öznitelik bulunduktan sonra HMM ile aktivite sınıflandırılması yapılmıştır. Test aşamasında, 15 kare boyutundaki video klipler kullanılmıştır.

Çizelge 5.2'deki deneylerde, durum sayısı olarak 9 kullanılmıştır. HMM tipi ergodik, ön işleme yöntemi ICA seçildiğinde, % 58.78 başarı oranı ile en iyi sonuç alınmıştır. Videodaki resimlerin boyutu küçültülmeden yapılan deneyler, küçültülerek yapılan deneylerden daha kötü sonuç vermiştir. Çünkü girdi olarak resmin pikselleri HMM'de kullanıldığında, model çok iyi eğitilememektedir. Girdideki verinin boyutu küçük olduğunda HMM daha iyi eğitilmektedir.

Çizelge 5.2 Weizmann 'da, video kliplerin, iz düşüm özniteliklerinin, HMM deneyleri sonuçları

Durum Sayısı	**Öznitelik Ön İşleme Yöntemi**	**HMM Tipi**	**Başarı Oranı %**
9	ICA	Ergodik	58.78
9	Normalizasyon	Ergodik	51.34
9	PCA	Ergodik	48.92

Çizelge 5.3'de gösterilen deneylerde, bir önceki deneylerden farklı olarak, videodaki resimlerin boyutları 10 da 1 oranında küçültülerek, iz düşüm alınmıştır. Ayrıca test aşamasında, videoların tamamı kullanılmıştır. HMM tipi ergodik, öznitelik ön işleme yöntemi olarak normalizasyon ve durum sayısı 4 seçilip, bütün video kareleri ile deney yapıldığında, en iyi başarı oranı olan % 90.32 elde edilmiştir. Bütün video kareleri ile yapılan deneyler, 15 kare boyutundaki video klipler ile yapılan deneylerden daha iyi sonuç vermiştir.

Çizelge 5.3 Weizmann'da, küçültme ve iz düşüm özniteliklerinin, HMM deneyleri sonuçları

Durum	Öznitelik Ön İşleme Yöntemi	HMM Tipi	Başarı Oranı %
4	Normalizasyon	Ergodik	90.32
4	ICA	Ergodik	88.17
4	PCA	Ergodik	86.02
6	ICA	Ergodik	84.94
6	PCA	Ergodik	83.87
6	Normalizasyon	Ergodik	80.64
9	ICA	Ergodik	89.24
9	PCA	Ergodik	84.94
9	Normalizasyon	Ergodik	77.41

Çizelge 5.4'de yapılan deneylerde, videodaki resimlerin boyutu 10 da 1 oranında küçültülüp, iz düşüm alınıp, çeşitli ön işleme yöntemlerinden geçirilerek video öznitelikleri bulunmuştur. Eğitim aşamasında, videodaki bütün karelerin öznitelikleri ve test sırasında sadece tek kare kullanılarak deney yapılmıştır.

Çizelge 5.4 Weizmann'da, tek bir karenin, küçültme ve iz düşüm özniteliklerinin HMM deneyleri sonuçları

Durum Sayısı	Öznitelik Ön İşleme Yöntemi	HMM Tipi	Başarı Oranı %
4	Normalizasyon	Ergodik	57.04
4	ICA	Ergodik	66.19
4	PCA	Ergodik	53.48
4	Normalizasyon	Soldan Sağa	56.86
4	PCA	Soldan Sağa	44.32
6	Normalizasyon	Ergodik	48.37
6	ICA	Ergodik	62.35
6	PCA	Ergodik	54.29
6	Normalizasyon	Soldan Sağa	48.85
6	PCA	Soldan Sağa	39.22
9	Normalizasyon	Ergodik	46.19
9	ICA	Ergodik	52.71
9	PCA	Ergodik	47.04
9	Normalizasyon	Soldan Sağa	46.91
9	PCA	Soldan Sağa	42.62

Çizelge 5.5 Weizmann'da video kliplerin, siluet ve Hu moment özniteliklerinin, HMM deneyler sonuçları

Durum Sayısı	Öznitelik Ön İşleme Yöntemi	HMM Tipi	Başarı Oranı %
4	Normalizasyon	Ergodik	76.53
4	PCA	Ergodik	73.92
4	ICA	Ergodik	84.43
6	Normalizasyon	Ergodik	74.27
6	PCA	Ergodik	75.69
6	ICA	Ergodik	78.32
9	Normalizasyon	Ergodik	75.93
9	PCA	Ergodik	76.09
9	ICA	Ergodik	83.59

Tek video karesi ile yapılan deneylerde, en iyi sonuç, ön işleme yöntemi normalizasyon, durum sayısı 4 ve HMM tipi olarak da ergodik seçildiğinde, % 57.04 olarak alınmıştır. Tek bir resim ile yapılan deneyler, birden fazla resim ile yapılan deneylerden daha başarılı olamamıştır.

5.1.2 Siluet ve Hu momentleri ile HMM

Bu deneylerde, veri kümesinin sağlamış olduğu siluetler kullanılmıştır. Bu kümenin videoları siyah beyaz olarak bulunmaktadır. Siluetin bulunduğu yerler beyaz arka plan siyahtır.

Videodaki karelere, Hu'nun değişmez 7 momentleri uygulanıp, ICA, PCA ve normalizasyon ön işleme süreçlerinden biri geçirilerek öznitelikler bulunmuştur. HMM tiplerinden soldan sağa HMM ve ergodik HMM'lerde, öznitelikler girdi olarak kullanılmıştır. HMM eğitiminde, tüm video kareleri kullanılmış olup, test aşamasında 15 karelik video klipler kullanılarak yapılan deneylerin sonuçları çizelge 5.5'de gösterilmiştir.

Yapılan deneylerde, ön işleme yöntemi ICA, durum sayısı 4 ve HMM tipi olarak ergodik seçildiğinde, en iyi sonuç olarak % 84.43 elde edilmiştir. Çizelge 5.5'deki

deneylerde, en iyi ön işleme yöntemi, ICA olmuştur.

Çizelge 5.6'da, eğitim aşamasında videonun bütün kareleri, test aşamasında 15 boyundaki video klipler yerine, bütün video kareleri kullanılmıştır.

Çizelge 5.6 Weizmann'da, siluet ve Hu özniteliklerinin HMM deneyleri sonuçları

Durum Sayısı	Öznitelik Ön İşleme Yöntemi	HMM Tipi	Başarı Oranı %
4	Normalizasyon	Ergodik	80.64
4	PCA	Ergodik	74.19
4	ICA	Ergodik	79.56
6	Normalizasyon	Ergodik	84.94
6	PCA	Ergodik	74.19
6	ICA	Ergodik	87.09
9	Normalizasyon	Ergodik	79.56
9	PCA	Ergodik	74.19
9	ICA	Ergodik	79.56

Durum Sayısı 6, HMM tipi ergodik ve öznitelik ön işleme yöntemlerinden ICA seçildiğinde, en iyi sonuç olarak % 87.09 başarı oranı elde edilmiştir. Tüm video kareleri kullanılarak yapılan deneyler, 15 kare uzunluğundaki klipler ile yapılan deneylerden daha iyi sonuç vermiştir.

Çizelge 5.7 Weizmann'da tek bir karenin, siluet ve Hu özniteliklerinin, HMM deneyleri sonuçları

Durum Sayısı	Öznitelik Ön İşleme Yöntemi	HMM Tipi	Başarı Oranı %
4	Normalizasyon	Ergodik	53.12
4	PCA	Ergodik	46.31
4	ICA	Ergodik	50.54
6	Normalizasyon	Ergodik	53.85
6	PCA	Ergodik	49.74
6	ICA	Ergodik	56.02
9	Normalizasyon	Ergodik	56.85
9	PCA	Ergodik	51.47
9	ICA	Ergodik	57.91

Test aşamasında birden fazla video karesi kullanıldığında, HMM video kareleri arasındaki akışı da modellemektedir. Bu yüzden, test aşamasında bir adet resmin kullanıldığı deneyler, test aşamasında video klibin boyutu birden fazla olduğu deneylerden daha kötü sonuç vermiştir. Tek bir video karesi kullanılarak yapılan deneylerde en iyi sonuç olarak % 57.91 elde edilmiştir.

5.1.3 Değiştirilmiş MHI ve Hu momentleri ile HMM

Weizmann'daki siluet veri kümesinde, MMHI ve Hu momentlerinin çeşitli ön işleme yöntemleri yöntemlerinin öznitelik olarak kullanıldığı ve HMM ile aktivite sınıflandırılması deneyleri bu bölümde yapılmıştır.

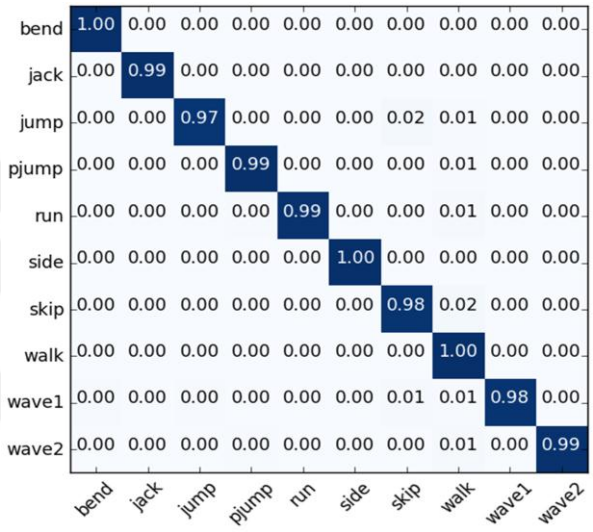
Videodaki resmin, MMHI ve Hu'nun değişmez 7 momentleri ile öznitelikleri bulunduktan sonra öznitelik sayısı 7'ye düşmektedir. ICA, PCA ve normalizasyon ön işleme süreçlerinden öznitelikler geçirilip, soldan sağa HMM ve ergodik HMM'de girdi olmaktadır. HMM eğitiminde, videonun bütün kareler, test aşamasında 15 karelik video klipler kullanılarak yapılan deneylerin sonuçları çizelge 5.8'de gösterilmiştir.

Çizelge 5.8 Weizmann'da, video kliplerin, MMHI ve Hu özniteliklerinin, HMM deneyleri sonuçları

Durum Sayısı	Öznitelik Ön İşleme Yöntemi	HMM Tipi	Başarı Oranı %
4	Normalizasyon	Ergodik	99.02
4	PCA	Ergodik	83.82
4	ICA	Ergodik	94.42
6	Normalizasyon	Ergodik	99.41
6	PCA	Ergodik	93.99
6	ICA	Ergodik	98.28
9	Normalizasyon	Ergodik	99.23
9	PCA	Ergodik	95.33
9	ICA	Ergodik	99.35

HMM tipi ergodik, durum sayısı 6 ve ön işleme yöntemi olarak da normalizasyon seçildiğinde, en iyi sonuç olarak % 99.40 başarı oranı elde edilmiştir. Weizmann veri kümesi için yakalanan en iyi sonuçlardan bir tanesidir. Çizelge 5.8'deki deneylerde, ön işleme yönteminden PCA, diğer ön işleme yöntemlerinden daha kötü sonuç vermiştir.

5 durum seçilerek, MMHI ve Hu momentlerinin kullanılarak ergodik HMM üzerinde yapılan deneydeki karmaşıklık matrisi şekil 5.2'de gösterilmiştir.



Şekil 5.2 MMHI – Hu özniteliği ile 5 durumlu HMM deneyinin karmaşıklık matrisi

Çizelge 5.8'deki deneylerden farklı olarak, test aşamasında videonun bütün karelerin kullanılarak, sınıflandırma yapıldığında alınan sonuçlar çizelge 5.9'da gösterilmiştir.

Videonun tüm kareleri kullanılarak sınıflandırma yapıldığından, en iyi sonuç olarak % 86.73 elde edilmiştir. En iyi sonuç alınan deney, ön işleme yöntemi normalizasyon ve HMM durumu 6 seçildiğinde alınmıştır. Önceki deneylerden farklı olarak, videonun bütün kareler, kullanıldığında daha kötü bir sonuç alınmıştır.

Çizelge 5.9 Weizmann'da, MMHI ve Hu özniteliklerinin, HMM deneyleri sonuçları

Durum Sayısı	Öznitelik Ön İşleme Yöntemi	HMM Tipi	Başarı Oranı %
4	Normalizasyon	Ergodik	83.52
4	PCA	Ergodik	71.91
4	ICA	Ergodik	67.18
6	Normalizasyon	Ergodik	86.73
6	PCA	Ergodik	78.38
6	ICA	Ergodik	77.54
9	Normalizasyon	Ergodik	84.49
9	PCA	Ergodik	76.31
9	ICA	Ergodik	85.18

Çizelge 5.10'da yapılan deneylerde, test aşamasında sadece bir tek resim kullanılmıştır. Herhangi bir dizi kullanılmamıştır.

Çizelge 5.10 Weizmann'da, tek bir karenin, MMHI ve Hu özniteliklerinin, HMM deneyleri sonuçları

Durum Sayısı	Öznitelik Ön İşleme Yöntemi	HMM Tipi	Başarı Oranı %
4	Normalizasyon	Ergodik	36.46
4	PCA	Ergodik	22.98
4	ICA	Ergodik	61.51
6	Normalizasyon	Ergodik	61.25
6	PCA	Ergodik	12.38
6	ICA	Ergodik	51.91
9	Normalizasyon	Ergodik	66.93
9	PCA	Ergodik	14.24
9	ICA	Ergodik	43.22

En iyi sonuç olarak, ön işleme yöntemi normalizasyon ve durum sayısı 9 seçildiğinde % 66.93 elde edilmiştir. Test aşamasında, tek video karesi kullanıldığı için HMM çok iyi sonuç vermemiştir.

Çizelge 5.11'deki deneylerde öznitelik olarak siluet ve MMHI'in farklı video klip boyutları ile deneylerin sonuçları gösterilmiştir.

Çizelge 5.11 Weizmann'da klip boyutuna göre, siluet ve MMHI özniteliklerinin, HMM deneylerinin karşılaştırılması

Test Örnekleri	Siluet (%)	MMHI (%)
5 Kare	74	98.12
10 Kare	80	98.16
15 Kare	82	99.05
Video	86	91.84

MMHI'ın öznitelik olarak kullanıldığı testler, siluetin kullanıldığı deneylerden daha iyi sonuç vermektedir. Öznitelik olarak MMHI ve video klip boyutu olarak 15 seçildiğinde en iyi sonuç olarak % 99.05 alınmıştır.

5.1.4 CNN ile HMM

Tezin bu bölümünde, derin ağlardan CNN ile çıkarılan öznitelikler kullanılmıştır. Derin ağ modellerinin, iyi eğitilebilmesi için büyük veriye ihtiyaç vardır. Weizmann ve KTH veri kümelerinde, veri sayısı az olduğundan, daha önceden eğitilmiş olan CNN modelleri kullanılmıştır. Eğitilmiş modellerden de SquezeeNet (Iandola vd. 2016) ve VGG-16 (Simonyan ve Zisserman 2014) kullanılmıştır. VGG-16 modeli ile yapılan deneyler uzun sürdüğü için bu model ile eğitim tüm deneylerde yapılmamıştır.

Çizelgelerde gösterilen deneylerde, SquezeeNet modeli kullanılmıştır. Çünkü bu model, yapılan deneylerde, AlexNet (Krizhevsky vd. 2012) modeli ile yapılan deneyler kadar başarılı ve 40 kat daha hızlıdır. Bu özniteliklerinin yanı sıra, SquezeeNet modeli, AlexNet modelinden, boyutu daha küçük, modelin açılma hızı ve cevap süresi oldukça kısadır.

Modelin tamamını eğitmek yerine, daha önceden eğitilmiş modeldeki bazı katmanlar tekrar eğitilerek (fine-tune), az verinin olduğu veri kümelerine uygun hale getirilmiştir. Deneylerde kullanılan modellerde, ilk katmanlarda herhangi bir eğitim yapılmamıştır. SquezeeNet modelinin, 1000 boyutundaki evrişimsel katmanı, Weizmann veri kümesinin sınıf sayısı olan 10 boyutundaki evrişimsel katman ile değiştirilmiştir. CNN

modeli eğitilirken sadece değiştirilen katman eğitilmiştir. Optimizer yöntemi olarak skolastik gradient descent kullanılmıştır. Eğitim oranı olarak 0.0001 ve momentum olarak da 0.9 seçilerek, eğitim yapılmıştır.

Weizmann veri kümesiyle yapılan diğer deneylerde, leave-one-out yöntemi seçilmişken, bu deneylerde, test ve eğitim verilerinin rastgele seçilerek, CNN modeli ile 5 kere eğitilip test edildiği bir çapraz doğrulama yöntemi kullanılmıştır. Bulunan sonuçların yine aritmetik ortalaması alınarak başarı oranları hesaplanmıştır.

Çizelge 5.12'de SquezeeNet modelindeki, son evrişimsel katman değiştirilip, eğitildikten sonra, HMM ile sınıflandırma yapılmıştır. Deneyler sırasında, 15 video karesinden oluşan klipler kullanılmıştır. Durum sayısı 4 ile 15 arasında değiştirilerek, çeşitli deneyler yapılmıştır.

Çizelge 5.12 Weizmann'da video kliplerin, CNN özniteliklerinin, HMM deneyleri sonuçları

Durum Sayısı	**HMM Tipi**	**Başarı Oranı %**
4	Soldan Sağa	32.82
4	Ergodik	63.15
7	Soldan Sağa	44.01
7	Ergodik	65.79
9	Soldan Sağa	38.93
9	Ergodik	67.93
12	Soldan Sağa	43.17
12	Ergodik	68.53
15	Soldan Sağa	46.12

Durum sayısı 15 ve HMM tipi ergodik seçilerek yapılan deneylerde en iyi sonuç olarak % 68.56 elde edilmiştir. Weizmann veri kümesinde 93 adet veri olduğu için, CNN modeli tam olarak eğitilememiştir.

CNN modelinin, daha iyi eğitilebilmesi için daha çok veriye ihtiyaç vardır. Veriyi çoğaltmak için augmentation yöntemi kullanılarak sınıflandırma başarısının artacağını,

Alex vd. (2012) çalışmasında göstermişlerdir.

Tez çalışmasında, Weizmann veri kümesindeki verileri çoğaltmak için resmin sol üst, sağ üst, sol alt, sağ alt ve orta kısma yakınlaştırma ile bunların dikey yansımaları kullanılarak, veri sayısı 12 katına çıkarılmıştır. Testler sırasında, 15 video karesinden oluşan klipler kullanılmıştır. Bu video kliplerin, augmentation yöntemiyle yapılan deneylerin sonuçları çizelge 5.13'de gösterilmiştir.

Çizelge 5.13 Weizmann'da video kliplerin, augmentation yöntemiyle çoğaltılarak, CNN özniteliklerinin HMM deneyleri sonuçları

Durum Sayısı	**HMM Tipi**	**Başarı Oranı %**
5	Soldan Sağa	39.32
5	Ergodik	89.66
9	Soldan Sağa	44.31
9	Ergodik	92.57
11	Soldan Sağa	51.31
11	Ergodik	91.97
13	Soldan Sağa	92.52
15	Ergodik	92.53
9	Soldan Sağa	44.31
9	Ergodik	92.57
11	Soldan Sağa	51.31
11	Ergodik	91.97
13	Soldan Sağa	92.52
15	Ergodik	92.53

Durum sayısı, 5 ile 15 arasından seçilerek yapılan deneylerde, en iyi sonuç, durum sayısı 15 ve HMM tipi ergodik olan deneyde, % 92.53 başarı oranı olarak elde edilmiştir. Augmention yöntemi ile veri çoğaltmasında başarı oranı artmış, ama MHI – Hu'nun 7 momentlerinin öznitelik olarak kullanıldığı deney sonuçlarını geçememiştir.

5.2 KTH Veri Kümesi ile Yapılan Deneyler

KTH veri kümesi, yavaş koşma, el çırpma, el sallama, koşma, yürüme, boks yapma olmak üzere 6 farklı insan aktivitesi sınıfı içermektedir. Bu hareketler, 25 farklı insan tarafından, dört farklı senaryo ile yapılarak toplamda 2391 adet video elde edilmiştir. Dört farklı senaryo şu şekildedir;

- Dışarıda ve farklı açıda,
- Dışarıda
- İçeride Dışarıda ve farklı kıyafetli

Bu ayarlarda, veri kümesi test ve eğitim olarak ikiye bölünmüştür. Bölme işlemi, insan bazında gerçekleştirilmiştir. Test verisinde, toplam 9 kişinin videoları bulunmaktadır. Bu kişiler 2, 3, 5, 6, 7, 8, 9, 10 ve 22 numaralı kişilerdir. Geriye kalan 16 kişinin videoları ise eğitim verisi olarak kullanılmaktadır. Eğitim verileri, klipler olarak bölündükten sonra rastgele olarak % 15 oranında validasyon verisi olarak bölünmüştür. KTH verisinin yayınlandığı makalede olduğu gibi, çok sınıflı bir sınıflandırıcıyı kullanılarak eğitilmekte ve değerlendirilmektedir. Deneylerde, performans ölçeği olarak, tüm sınıflar için yapılan deney sonuçlarının ortalaması kullanılmıştır. Şekil 5.3'de, KTH veri kümesindeki bazı videolardan örnek resimler gösterilmiştir.



Şekil 5.3 KTH veri kümesinden birkaç örnek çerçeve

Şekil 5.3'de, ilk satırdaki küçük resimler, soldan sağa doğru açıklanırsa; boks yapma, iki el çırpma, iki el sallama. İkinci satırdaki küçük resimler soldan sağa doğru açıklanırsa; tempolu yavaş koşma, koşma ve yürüme. Son satırdaki küçük resimler açıklanırsa; ikinci koşma, tempolu yavaş koşma ve yürüme.

5.2.1 Değiştirilmiş MHI ve Hu momentleri ile HMM

MMHI ile aktivitenin zaman modellemesi, 2D olarak gösterilebilmektedir. Bu bölümdeki deneylerde, videodaki aktivitelerin sınıflandırılması için gereken öznitelikler, MMHI uygulandıktan sonra Hu'nun 7 momentleri ile bulunmuştur. Hu 7 momentleri ile sayısı 7'ye düşürülen öznitelikler, normalizasyon yöntemleri ile ön işlemeden geçirilmiştir.

HMM, zamandaki olayların, sınıflandırılmasında başarılı olduğundan, video sınıflandırılması için de kullanılmıştır. Hu'nun 7 momentleri ile bulunan öznitelikler, ergodik HMM ile eğitildikten sonra, her bir sınıf için ayrı ayrı modellenmektedir

(KTH'da bulunan 6 sınıf için 6 farklı HMM oluşturulmaktadır.). Bu modellerin her biri, veri setinde bulunan ilgili sınıftaki verilerle eğilmiştir.

Test aşamasında 15 ve 30 karelik video klipler kullanılarak, başarı oranları bulunmuştur. Her video klip, her sınıfa ait olan modelden geçirildikten sonra, log olasılıkları (sınıfa olan benzerliği) hesaplanmaktadır. En yüksek olan sınıf, klibin ait olduğu sınıf olarak seçilmektedir. Çizelge 5.14'de HMM'lerde kullanılan farklı durum sayıları ile yapılan deneylerdeki başarı oranları gösterilmiştir.

Çizelge 5.14 KTH'da, video kliplerin, MMHI özniteliklerinin HMM deneyleri sonuçları

Durum Sayısı	**15 Karelik Klip**	**30 Karelik Klip**
3	70.14	-
5	74.94	-
8	86.42	91.29
11	89.15	-
15	91.65	94.58
17	90.97	94.38
20	92.77	95.45
50	95.29	96.73

Durum sayısı arttıkça, başarı oranının da arttığı gözlemlenmiştir. En iyi sonuç, durum sayısı 50 verildiğinde alınmıştır. 15 karelik video klip için yapılan deneylerde, en iyi başarı oranı % 95,29 olarak hesaplanmıştır. Video klipteki kare sayısı 30 olarak seçildiğinde ise % 96,73 başarı oranı elde edilmiştir. Video klibin uzunluğu arttıkça, başarı oranının da arttığı görülmüştür.

5.2.2 CNN ile HMM

El ile seçilmiş öznitelikleri kullanmak, her deney ortamında iyi çalışmayabilir. El ile seçilmiş öznitelikler kullanmak yerine, ortamdan bağımsız olarak öznitelik bulunması, daha iyidir. Çünkü bağımsız olarak bulunan öznitelikler, deney ortamları farklı olsa da çalışacaktır. Derin ağlar ile bulunan öznitelikler, el ile bulunan özniteliklerden daha iyi

sonuçlar verebildiği gözlenmiştir. Büyük sayıdaki veri ile eğitilmiş olan derin ağlar, resimlerdeki öznitelikleri, otomatik olarak çıkartıp, sınıflandırma yapabilmektedir.

Bu bölümde, öznitelikleri CNN ile çıkartılıp, sınıflandırma için çeşitli sınıflandırıcılar kullanılmıştır. KTH veri kümesinde, derin ağı eğitecek kadar veri bulunmadığından, daha önce büyük sayıdaki veriler ile eğitilmiş olan modellerde, iyileştirilme (fine-tuned) işlemi yapılmıştır.

Çeşitli başarılar yakalamış, çok sayıda derin ağ modelleri bulunmaktadır. İyileştirmek (fine-tuned) için, diğer modellere göre daha küçük, hızlı ve aynı zamanda, en az onlar kadar başarı oranı yakalayabilen SquezeeNet (Iandola vd. 2016) modeli kullanılmıştır. SquezeeNet modelinin, son evrişimsel katmanında bulunmakta olan 1000'lik evrişimsel katman, 6 adet (KTH'daki sınıf sayısı) evrişimsel katman ile değiştirilmiştir. İyileştirme aşamasında sadece değiştirilen katman eğitilmektedir. Bu sayede, önceden resimlerin tanınması için öğrenilmiş olan ağırlıklar, bozulmamaktadır. Öğrenme katsayısı olarak 0.0001 ve momentum olarak 0.9 seçilmiştir. Optimizer yöntemi olarak skolastik gradient descent kullanılmıştır.

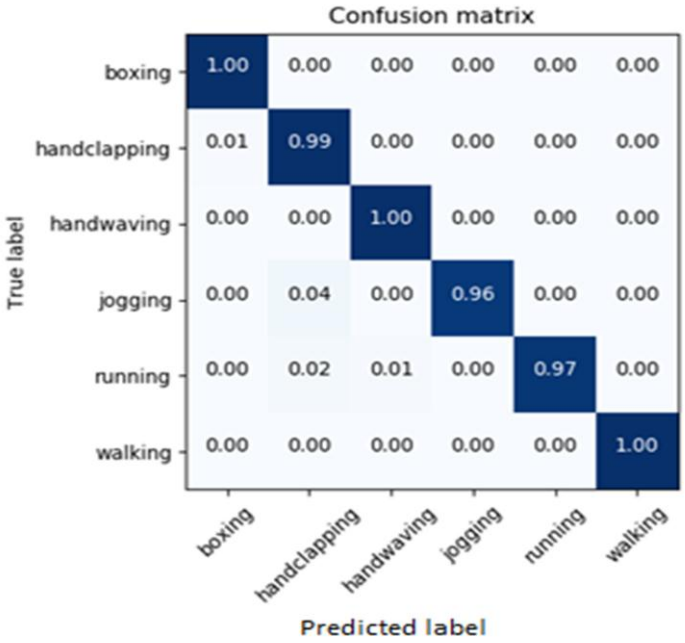
Videodaki her karenin öznitelikleri, CNN ile bulunduktan sonra, HMM sınıflandırıcısına verilmektedir. Önceki yöntemlerde olduğu gibi her sınıf için ayrı bir HMM oluşturulmuştur. Sınıfa ait olan videolar, kendi HMM'lerini eğitmek için kullanılmıştır. Eğitim aşamasında oluşmuş olan 6 farklı model test için kullanılmıştır. Video klipler, her sınıfa ait olan modellere uygunluğu Viterbi algoritması ile belirlenmiştir. Uygunluk değeri en yüksek olan, tahmin edilen sınıf olarak seçilmektedir. Ergodik HMM ile 15 ve 30 karelik klipler ile yapılan deneylerin başarı oranları çizelge 5.15'de gösterilmiştir.

Çizelge 5.15 KTH'da, video kliplerin, CNN özniteliklerinin, HMM deneyleri sonuçları

Durum Sayısı	**15 Karelik Klip**	**30 Karelik Klip**
3	87.62	-
5	92.45	-
7	95.78	98.25
8	97,41	98.78
11	97.38	-
15	97.58	99.22
17	97.69	99.30
20	95.54	97.71
30	97.84	99.18
50	98.08	99.30

Çıkan sonuçlara göre, CNN ile çıkarılan öznitelikler, MHI – Hu'nun 7 momentleri ile çıkarılan özniteliklerden daha iyi sonuçlar vermiştir. Klip uzunluğu arttıkça başarı oranı da artmaktadır. 15 karelik klip için en iyi sonuç, durum sayısı olarak 50 seçildiğinde, % 98.08 başarı oranı yakalanmıştır. 30 karelik video klip ile yapılan deneylerde, en iyi sonuç durum sayısı olarak 17 seçildiğinde ise % 99.30 başarı oranı yakalanmıştır.

Şekil 5.4'de, 30 karelik video klip kullanılarak yapılan deneyler arasında en yüksek sonuç 17 durum seçilerek yapılan deneyin, karmaşıklık matrisi gösterilmiştir. Binde 4 oranında yavaş koşma ve binde 3 oranında da koşmada hata alınmıştır.



Şekil 5.4 Video klip uzunluğu 30 ve 17 durum sayılı deneyin karmaşıklık matrisi

HMM sınıflandırıcısında, MHI – Hu'nun 7 momentlerinin öznitelik olarak kullanıldığı deneylerde en iyi sonuç alınmıştır. El ile özniteliklerin bulunduğu yöntemlerden farklı olarak özniteliklerin otomatik olarak bulunduğu yöntemlerden CNN ile iyi sonuçlar bulunmuştur. CNN ile özniteliklerin bulunarak LSTM ile sınıflandırılmasının yapıldığı deneyler bir sonraki bölümde açıklanmaktadır.

5.2.3 CNN ile LSTM

Zaman akışının sınıflandırılmasında, RNN, en çok kullanılan derin ağ yöntemidir. RNN mimarileri, arasında en yaygın olarak kullanılan mimarilerden birisi LSTM'dir. Daha önceki deneylerde, videodan sınıflandırılma yapıldığında, HMM kullanılmıştır. Bu bölümde, HMM ile LSTM sınıflandırıcılarının karşılaştırılabilmesi için, KTH veri kümesindeki CNN ile HMM bölümündeki deneylerde kullanılan öznitelikler ile aynı deneyler LSTM ile yapılmıştır.

Eğitilmiş bir model olan SquezeeNet ağı öznitelik çıkarılması amacıyla kullanılmıştır.

SqueezeNet ağının veri kümesine uygun hale getirilmesi için iyileştirmeler (fine-tuned) yapılarak eğitilmiştir. Eğitilmiş CNN modelinden, videonun her bir karesi için öznitelikleri çıkarılmıştır. Videodaki karelerin çıkarılan özniteliklerinin, sınıflandırılması için LSTM ağında eğitilip, test edilmiştir.

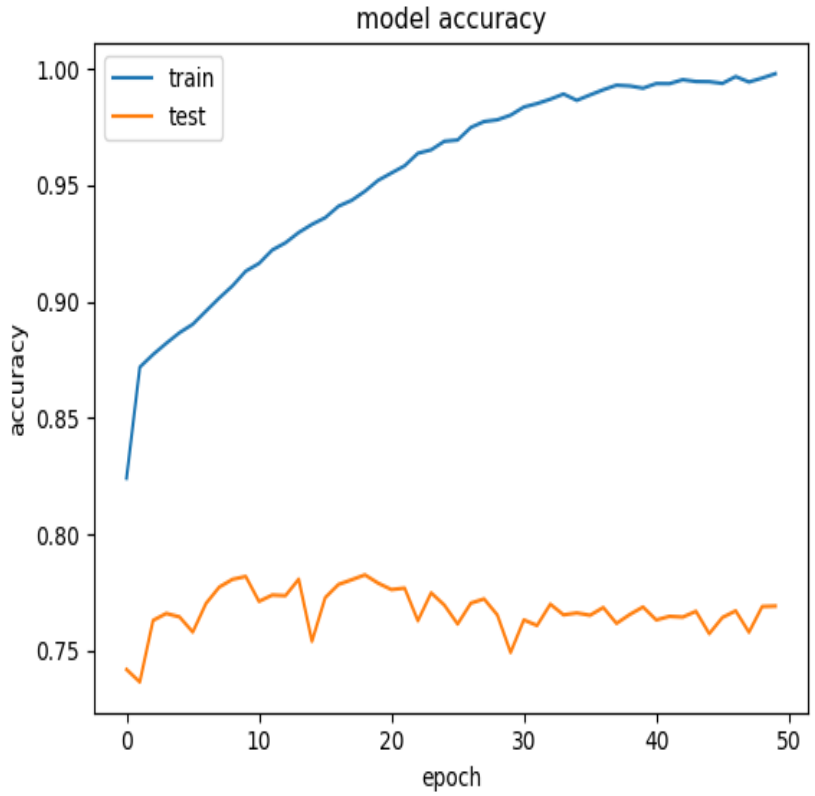
CNN ile eğitilmiş modelden çıkarılmış olan öznitelikler, çeşitli LSTM mimarilerine verilerek başarı oranları bulunmuştur. KTH veri kümesinin yayınlandığı makalede (Schuldt vd. 2004) belirtildiği gibi veriler, eğitim (16 kişi) ve test verisi olarak ayrıldıktan sonra kullanılmıştır. Video klipler oluşturulduktan sonra, eğitim videolarından, rastgele olarak % 15 oranında onaylama (validasyon) verisi olarak ayrılıp deneyler yapılmıştır. Onaylama verilerinden alınan onaylama sonuçları ile eğitim verileri ile alınmış olan eğitim başarı oranları çizelge 5.16'da gösterilmektedir.

Çizelge 5.16 KTH'da, video kliplerin, CNN özniteliklerinin, LSTM deneyleri sonuçları

LSTM Mimarisi	**Eğitim Başarı Oranı**	**Validasyon Başarı Oranı**
4-32	91.53	91.32
8	89.85	89.34
8-64	95.35	94.81
16	89.59	88.84
32	92.70	92.45
64	95.48	95.09
128	98.82	97.80
128-64	99.41	99.16
128-64-32	98.83	98.26

128-64 LSTM mimarisi kullanılarak, en yüksek onaylama başarı oranı olan % 99.16 elde edilmiştir. Eğitim başarı oranı olarak en yüksek sonuç 128-64 LSTM mimarisinde % 99.41 olarak alınmıştır. En yüksek validasyon başarı oranı yakalanan deney test verileri üzerinde denenmiş ve % 76.92 başarı oranı elde edilmiştir. Onaylama verilerinden alınan onaylama başarı oranına ve HMM ile yapılan deneylerde alınan başarı oranı daha yüksektir.

Şekil 5.5'de, validasyon başarı oranlarından, en iyi sonuç olan 128-64 mimarisindeki, onaylama ve eğitim başarı oranlarının, epok sayısı ile olan değişimi gösterilmiştir. Eğitim başarı oranı % 100'e yaklaşmakta iken, onaylama başarı oranı, yatay olarak ilerlemektedir. Bu da modelin görmediği veriler üzerinde tahmin yeteneğinin iyi olmadığını, ezberlediğini göstermektedir.



Şekil 5.5 KTH'da, 128-64 LSTM mimarisinde eğitim ve onaylama başarı oranları

Ham KTH verileri kullanılarak yapılan deneyde, pek başarılı sonuç alınmamasından dolayı, veriler, veri çoğaltma teknikleri kullanılarak, tekrar teste tabi tutulmuştur.

Kullanılan augmentation teknikleri, sol üst, sağ üst, sağ alt, sol alt tarafını kaydırma, ortadan yakınlaştırma ve bu alınanların dikey yansımaları kullanılmıştır. Bu yöntemler ile veriler 12 kata çıkarılmıştır. Veriler çoğaltıldıktan sonra, SqueezeNet modeli, tekrardan eğitim verileri ile eğitilip, öznitelikleri çıkarılmıştır. LSTM'in eğitim aşamasında çoğaltılan veriler kullanılmıştır. Test aşamasında, 9 kişinin videoları bu

yöntemlerle çoğaltılıp, LSTM mimarisi eğitilmiştir. Yakalanan başarı oranı, yeni oluşturulmuş olan videolar kullanılarak, doğrulama başarısı en yüksek olan LSTM mimarisi seçilerek yapılmıştır.

Bir videonun sınıfını bulmak için, 11 adet augmentation ile oluşturulmuş olan videolar ve videonun kendisi, LSTM ile sınıflandırılmadan geçirilerek, en çok seçilen sınıf videonun tahmin edilen sınıfı olmaktadır. Bu şekilde tüm klipler, test aşamasından geçirilmiştir ve % 81.92 başarı oranı elde edilmiştir.

KTH veri kümesinde, LSTM mimarisinin, HMM mimarisinden daha başarısız olduğunu görülmüştür. LSTM mimarisi, daha önceden görmüş olduğu veriler için başarılı olduğu görülmüştür. Hiç görmediği videolarda, yeterli bir tahmin yapamamıştır. Az sayıda veri olan durumlarda, HMM çok daha iyi sonuçlar vermiştir. Veriler augmentation ile çoğaltılarak, LSTM'den alınan başarı oranı yükselmiştir fakat yine de HMM'den alınan sonuç daha iyidir.

6. SONUÇLAR

Bu tez çalışmasında videolardan, çeşitli yollarla öznitelikler çıkarılıp, çeşitli makine öğrenmesi yöntemleri ile sınıflandırmalar yapılmıştır. Video sınıflandırmalarında, kamuya açık KTH ve Weizmann veri kümeleri kullanılmıştır. Weizmann ile yapılan deneyde % 99.40 başarı oranı elde edilirken (Alp ve Keleş 2017), literatürde alınan sonuçlar ile yarışabilir bir başarı yakalanmıştır. KTH veri kümesinde, % 99.30 başarı oranı elde edilmiştir (Alp ve Keleş 2018). KTH veri kümesinde, bu elde edilen başarı oranı, şimdiye kadar elde edilmiş, en iyi sonuçtur.

Öznitelik olarak MHI, Hu Momentleri, MMHI, CNN, siluet, kestirim (histogram) metotları kullanılmıştır. Değiştirilmiş MHI (MMHI) yöntemiyle, hareket geçişi görüntüsü hesaplanmaktadır. Bu yöntemle, hareket enerjisi görüntüsünü, hesaplamalar yapılmadan direk oluşturulmaktadır.

Son zamanlarda, resimlerin özniteliklerinin bulunmasında en çok kullanılan yöntemlerden biri derin ağlardır. Çalışmada, derin ağlar arasından CNN ağı, öznitelik seçiminde kullanılmıştır. Tüm bir ağı eğitmek için çok fazla veri gerektiğinden, daha önceden eğitilmiş ve başarılı sonuçlar alınmış olan SqueezeNet mimarisi kullanılmıştır. Bu mimari, çok küçük olduğu için eğitimi hızlı bir şekilde gerçekleştirilmiştir.

Zaman modellemesinde yaygın olarak kullanılmakta olan HMM, bu çalışmada videodaki aktivitelerin sınıflandırılması için kullanılmıştır. Daha önceden, çeşitli yöntemlerle oluşturulmuş olan öznitelikler, HMM üzerinde eğitilmiştir ve test edilmiştir. Çeşitli uzunluktaki video klipler, kayan pencere yöntemiyle oluşturulup, HMM ile sınıflandırılması yapılmıştır. Videolarda aktivite sınıflandırılmasında ise RNN yöntemlerinden, yaygın olarak kullanılmakta olan LSTM ağı kullanılmıştır. Videoların öznitelikleri çıkarıldıktan sonra, kayan pencere yöntemiyle videolar parçalanıp, sınıflandırma yapılmıştır. LSTM ile elde edilen başarı oranları, HMM ile elde edilen başarı oranlarından düşüktür. Bunun temel nedeninin, üzerinde çalışılan iki veri setinde de LSTM parametrelerini olgunlaştırmaya yetecek sayıda verinin olmadığı düşünülmektedir.

CNN kullanılarak yapılan deneylerde, videolar çeşitli yöntemlerle çoğaltılmıştır. Videolar 12 katına çıkarılarak, CNN modelinin daha iyi eğitilmesi sağlanmıştır. Veriler çoğaltılarak yapılan deneylerde daha iyi sonuçlar alınmıştır. Weizmann gibi veri sayısının yetersiz olduğu veri kümelerinde, CNN ile nitelikli öznitelik çıkarılamamaktadır. Verilerin az olduğu durumlarda, el ile seçilmiş olan öznitelikler, otomatik olarak seçilmiş olan özniteliklerden daha iyi sonuç vermektedir. Weizmann veri kümesinde, el ile seçilmiş olan özniteliklerden en iyi sonuçlar MMHI ve Hu momentlerinden alınmıştır. KTH gibi veri sayısının derin bir ağı eğitilebilecek kadar sayıda olmadığı veri kümelerinde ise CNN ile öznitelik bulmak daha iyi sonuç vermektedir.

Özniteliklerin ayrı olarak hesaplanıp, LSTM mimarisine verilen yöntemlerde başarılı sonuçlar alınamamıştır. Ancak şu anda yaygın olarak kullanılan, CNN ve LSTM mimarilerinin birleştirilerek oluşturulmuş olan mimarilerde daha iyi sonuçlar alınmıştır. Özniteliklerin ayrı olarak hesaplanıp, HMM ile yapılan sınıflandırmalarda, LSTM ile yapılan sınıflandırmalara göre daha iyi sonuçlar elde edilmektedir.

Bundan sonraki çalışmalarda, HMM ve CNN birleştirilerek oluşturulmuş olan bir mimari denenebilir. Bu sayede HMM mimarisinden çıkan sonuçlar geri besleme yöntemiyle CNN, HMM'e özel olarak eğitilebilir. Bu mimari daha fazla veri içeren veri kümeleriyle sınanabilir.

Weizmann veri kümesinde, daha önce yapılan çalışmalarda en iyi sonuç olarak % 100 başarı oranı yakalanmıştır. Tezde MMHI ve Hu moment özniteliği kullanılıp 6 durumlu HMM kullanıldığında en iyi sonuç olarak % 99.41 başarı oranı yakalanmıştır. KTH veri kümesinde, daha önceden en iyi sonuç olarak, Sadanand vd. (2012), % 98.20 başarı oranı yakalamıştır. Tezde, bu veri kümesi için alınan en iyi sonuç, öznitelik yöntemi olarak CNN ve 17 durumlu HMM kullanıldığında, % 99.30 başarı oranı yakalanmıştır. Daha önceden alınan en iyi sonucu geçmiştir.

KAYNAKLAR

Afsar, P., Cortez, P. and Santos, H. 2015. Automatic Human Action Recognition from Video Using Hidden Markov Model. 2015 IEEE 18th International Conference on Computational Science and Engineering. 105-109. Porto.

Ahad, M., Tan, J., Kim, H. and Ishikawa, S. 2009. Temporal Motion Recognition and Segmentation Approach. Int. J. Imaging Syst. Technol., 91-99.

Alp, E.C. and Keles, H. 2017. Action recognition using MHI based Hu moments with HMMs,. IEEE EUROCON 2017 -17th International Conference on Smart Technologies. Ohrid, Macedonia.

Alp, E.C. and Keles, H. 2018. A Comparative Study of HMMs and LSTMs on Action Classification with Limited Training Data,. 2018 SAI Intelligent Systems Conference, London.

Baum, L. 1972. An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. Inequalities III: Proceedings of the 3rd Symposium on Inequalities. Los Angeles.

Bobick, A. and Davis, J. 2001. The recognition of human movement using temporal templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(3), 257-267.

Cho, K., Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. Conference on Empirical Methods in Natural Language Processing (EMNLP 2014).

Donahue, J., Hendricks, L., Rohrbach, M., Venugopalan, S., Guadarrama, S., Saenko, K. and Darrell, T. 2017. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4), 677-691.

Gers, F. and Schmidhuber, J. 2000. Recurrent nets that time and count. Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium. Como.

Goodfellow, I., Bengio , Y., and Courville, A. 2016. Deep Learning. MIT Press.

Hsieh, C.H., Huang, P. and Tang, M.D. 2011. Human Action Recognition Using Silhouette Histogram. Proceedings of the Thirty-Fourth Australasian Computer Science Conference. Perth, Australia..

Iandola, F., Han, S., Moskewicz, M., Ashraf, K., Dally, W. and Keutzer, K. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB

model size. arXiv preprint arXiv:1602.07360.

Ji, S., Xu, W., Yang, M. and Kai, Y. 2013. 3D Convolutional Neural Networks for Human Action Recognition. IEEE Trans. Pattern Anal. Mach. Intell., 35, 221-231.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L. 2014. Large-Scale Video Classification with Convolutional Neural Networks. IEEE Conference on Computer Vision and Pattern Recognition. Columbus, OH.

Krizhevsky, A., Sutskever, I. and Hinton, G. 2012. Imagenet classification with deep convolutional. Proc. Advances in Neural Information Processing Systems (NIPS), 1097–1105.

Kuehne, H., Arslan, A. and Serre, T. 2014. The Language of Actions: Recovering the Syntax and Semantics of Goal-Directed Human Activities. Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, 780-787. Washington, DC, USA.

Kuhne, H., Jhuang, H., Garrote, E., Poggio, T. and Serre, T. 2011. Hmdb: A large video database for human motion recognition. IEEE International Conference on Computer Vision (ICCV).

Le, Q., Zou, W., Yeung, S. and Ng, A. 2011. Learning Hierarchical Invariant Spatio-temporal Features for Action Recognition with Independent Subspace Analysis. Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. Washington, DC, USA.

Lecun, Y., Bottou, L., Bengio, Y. and Haffner, P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

Lei, J., Li, G., Zhang, J., Guo, Q. and Tu, D. 2016. Continuous action segmentation and recognition using hybrid convolutional neural network-hidden Markov model model. IET Computer Vision, 10, 537-544.

Lloyd, S. 1982. Least squares quantization in PCM. IEEE Transactions on Information Theory, 129-137.

Martınez-Contreras, F., Orrite-Urunuela, C., Herrero-Jaraba, E., Ragheb, H. and Velastin, S. 2009. Recognizing Human Actions Using Silhouette-based HMM. 2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance. Genova.

Medoza, M. and Perez de la Blanca, N. 2007. HMM-Based Action Recognition Using Contour Histograms. Iberian Conference on Pattern Recognition and Image Analysis, 394-401. Girona- Spain.

Ming-Kuei, H. 1962. Visual pattern recognition by moment. Information Theory, IRE Transactions, 8, 179-187.

Ng, J.H., Hausknecht, M., Vijayanarasimhan, S., Vinyals, O., Monga , R. and Toderici, G. 2015. Beyond short snippets: Deep networks for video classification. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Boston, MA.

Nielsen, M. 2015. Deep Learning. Neural Networks and Deep Learning. Determination Press.

Rabiner, L.R. 1990. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Readings in Speech Recognition 267-296. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Russakovsky, O., Deng , J., Su, H., Krause , J., Satheesh , S., Ma , S. and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 211-252.

Sadanand, S. and Corso, J.J. 2014. Action bank: A high-level representation of activity in video. 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA.

Schuldt, C., Laptev, I. and Caputo, B. 2004. Recognizing Human Actions: A Local SVM Approach. Proceedings of the Pattern Recognition. Washington, DC, USA.

Simonyan, K. and Zisserman, A. 2014. Two-stream Convolutional Networks for Action Recognition in Videos. Proceedings of the 27th International Conference on Neural Information Processing Systems. Montreal, Canada.

Soomro, K., Zamir, A. and Shah, M. 2012. UCF101: a dataset of 101 human actions classes from videos in the wild. CoRR, abs/1212.0402.

Sundaresan, A., RoyChowdhury , A. and Chellappa, R. 2003. A hidden Markov model based framework for recognition of humans from gait sequences. Proceedings 2003 International Conference on Image Processing, 93-96.

Türkoğlu, M. 2013. Otomatik Kan Hücrelerinin Tanınması and Sınıflandırılmasına Dayalı Görüntü İşleme Yöntemlerinin Kullanılması. Elazığ: Fırat Üniversitesi Fen Bilimleri Enstitüsü.

Vezzani, R., Baltieri, D. and Cucchiara, R. 2010. HMM Based Action Recognition with Projection Histogram Features. roceedings of the 20th International Conference on Recognizing Patterns in Signals, Speech, Images, and Videos. Istanbul, Turkey.

Viterbi, A. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory, 260-269.

Weinland, D., Ronfard, R. and Boyer, E. 2011. A Survey of Vision-based Methods for Action Representation, Segmentation and Recognition. Comput. Vis. Image Underst., 224-241.

Yamato, J., Ohya, J. and Ishii, K. 1992. Recognizing human action in time-sequential images using hidden Markov model. Proceedings 1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 379-385. Champaign.

ÖZGEÇMİŞ

Adı Soyadı : Elit Cenk Alp

Doğum Tarihi :1989

Medeni Hali :Evli

Yabancı Dili :İngilizce

Eğitim Durumu

Lise :Nermin Mehmet Çekiç Anadolu Lisesi (2007)

Lisans :Ankara Üniversitesi, Bilgisayar Mühendisliği (2012)

Yüksek Lisans :Ankara Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği (Mayıs, 2018)

Çalıştığı Şirketler

GATE Elektronik A.Ş – Bilgisayar Mühendisi (Temmuz 2012 – Haziran 2015)

NETAŞ A.Ş – Yazılım Geliştirme Mühendisi (Temmuz 2015 – Devam ediyor)

Uluslararası Kongre Sunum

Alp, E.C. and Keles, H. 2017. Action recognition using MHI based Hu moments with HMMs,. IEEE EUROCON 2017 -17th International Conference on Smart Technologies. Ohrid, Macedonia.

Alp, E.C. and Keles, H. 2018. A Comparative Study of HMMs and LSTMs on Action Classification with Limited Training Data,. 2018 SAI Intelligent Systems Conference (IntelliSys), London.