

**RECOGNITION OF HUMAN INTERACTIONS USING
HISTOGRAM OF SEQUENCES**

**İNSAN ETKİLEŞİMLERİNİN DİZİLİMLERİN
HİSTOGRAMLARI İLE TANINMASI**

AYTAÇ ÇAVENT

ASSOC. PROF. DR. NAZLI İKİZLER CİNBİŞ

Supervisor

Submitted to Graduate School of Science and Engineering of Hacettepe University
as a Partial Fulfillment to the Requirements
for the Award of the Degree of Doctor of Philosophy
in Computer Engineering

Oct 2018

This work named "Recognition of Human Interactions using Histogram of Sequences" by AYTAÇ ÇAVENT has been approved as a thesis for the Degree of **DOCTOR OF PHILOSOPHY IN COMPUTER ENGINEERING** by the below mentioned Examining Committee Members.

Prof. Dr. A. Aydın ALATAN

Head

.....

Assoc. Prof. Dr. Nazlı İKİZLER CİNBİŞ

Supervisor

.....

Assoc. Prof. Dr. İbrahim Aykut ERDEM

Member

.....

Prof. Dr. Pınar KARAGÖZ

Member

.....

Assoc. Prof. Dr. Ahmet Burak CAN

Member

.....

This thesis has been approved as a thesis for the Degree of **DOCTOR OF PHILOSOPHY IN COMPUTER ENGINEERING** by Board of Directors of the Institute for Graduate School of Science and Engineering.

Prof. Dr. Menemşe GÜMÜŞDERELİOĞLU
Director of the Institute of
Graduate School of Science and Engineering

YAYINLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin / raporunun tamamını veya herhangi bir kısmını, basılı (kağıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe Üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alınarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim Kurulu tarafından yayınlanan “ **Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge**” kapsamında tezim aşağıda belirtilen koşullar haricinde YÖK Ulusal Tez Merkezi / H. Ü. Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

- o Enstitü / Fakülte yönetim kurulu kararı ile tezimin erişime açılması mezuniyet tarihinden itibaren 2 yıl ertelenmiştir. ⁽¹⁾
- o Enstitü / Fakülte yönetim kurulunun gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihinden itibaren Ay ertelenmiştir. ⁽²⁾
- o Tezimle ilgili gizlilik kararı verilmiştir. ⁽³⁾

04/10/2018

(İmza)

Aytaç CAVENT
Öğrencinin Adı SOYADI

“Lisansüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge”

- (1) Madde 6. 1. Lisansüstü teze ilgili patent başvurusu yapılması veya patent alma sürecinin devam etmesi durumunda, tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu iki yıl süre ile tezin erişime açılmasının ertelenmesine karar verebilir
- (2) Madde 6. 2. Yeni teknik, materyal ve metotların kullanıldığı, henüz makaleye dönüşmemiş veya patent gibi yöntemlerle korunmamış ve internetten paylaşılması durumunda 3. Şahıslara veya kurumlara haksız kazanç imkanı oluşturabilecek bilgi ve bulguları içeren tezler hakkında tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü ve fakülte yönetim kurulunun gerekçeli kararı ile altı ayı aşmamak üzere tezin erişime açılması engellenebilir.
- (3) Madde 7. 1. Ulusal çıkarları veya güvenliği ilgilendiren, emniyet, istihbarat, savunma ve güvenlik, sağlık vb. konulara ilişkin lisansüstü tezlerle ilgili gizlilik kararı, tezin yapıldığı kurum tarafından verilir*. Kurum ve kuruluşlarla yapılan işbirliği protokolü çerçevesinde hazırlanan lisansüstü tezlere ilişkin gizlilik kararı ise, ilgili kurum ve kuruluşun önerisi ile enstitü veya fakültenin uygun görüşü üzerine üniversite yönetim kurulu tarafından verilir. Gizlilik kararı verilen tezler Yükseköğretim Kuruluna bildirilir.
Madde 7. 2. Gizlilik kararı verilen tezler gizlilik süresince enstitü veya fakülte tarafından gizlilik kuralları çerçevesinde muhafaza edilir, gizlilik kararının kaldırılması halinde Tez Otomasyon Sistemine yüklenir.

* Tez danışmanının önerisi ve enstitü anabilim dalının uygun görüşü üzerine enstitü veya fakülte yönetim kurulu tarafından karar verilir.

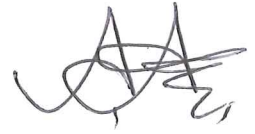
ETHICS

In this thesis study, prepared in accordance with the spelling rules of Institute of Graduate Studies in Science of Hacettepe University,

I declare that

- all the information and documents have been obtained in the base of the academic rules.
- all audio-visual and written information and results have been presented according to the rules of scientific ethics
- in case of using others works, related studies have been cited in accordance with the scientific standards
- all cited studies have been fully referenced
- I did not do any distortion in the data set
- and any part of this thesis has not been presented as another thesis study at this or any other university.

03/10/2018



AYTAÇ ÇAVENT

ABSTRACT

RECOGNITION OF HUMAN INTERACTIONS USING HISTOGRAM OF SEQUENCES

Aytaç ÇAVENT

Doctor of Philosophy, Computer Engineering Department

Supervisor: Assoc. Prof. Dr. Nazlı İKİZLER CİNBİŞ

October 2018, 116 pages

In recent years, many techniques have been proposed for recognition of the human interactions in the computer vision literature. To recognize human interactions, spatial and temporal descriptors have been used together. In modeling of the interactions in the spatial domain, remarkable progress has been made with the experience coming from image processing area. In recognition of the human interactions, modeling temporal information has critical importance, too. It is known that the successful methods that have been proposed in the recent years model both spatial and temporal structure of the motion. In this thesis, we propose a novel representation for human interaction recognition, namely histogram of temporal sequences. The proposed representation consists of the following steps: extracting basic feature descriptors, creating common sequential patterns for each class by using sequence mining, selecting discriminative sequential patterns, creating histograms of sequences from videos and classification. In the sequence mining step, by using hierarchical pyramid structures, the different temporal behaviors of the interactions at different temporal scales are included in the final model. The proposed method can also be used with different type of features. In accordance with this capability, we have tested the performance of the new representation with the deep features. Pre-trained deep features are fine-tuned on the dataset and extracted features are

used with the proposed method. We have provided the performance results on two publicly available datasets: UT Interactions Dataset and TV Human Interactions Dataset. In the UT Interactions Dataset, we have the best result on set#2 and the same performance with the best method on the set#1. In the TV Human Interactions Dataset, we have obtained the best results in the literature.

Keywords: interaction recognition, sequence mining, computer vision, video descriptors



ÖZET

İNSAN ETKİLEŞİMLERİNİN DİZİLİMLERİN HİSTOGRAMLARI İLE TANINMASI

Aytaç ÇAVENT

Doktora, Bilgisayar Mühendisliği

Danışman: Doç. Dr. Nazlı İKİZLER CİNBİŞ

Ekim 2018, 116 sayfa

Son yıllarda bilgisayarlı görü alanında insan etkileşimlerinin tanınması problemi üzerinde literatürde çok fazla sayıda yöntem önerilmiştir. İnsan etkileşimlerinin tanınması için hem uzamsal hem de zamansal öznitelikler beraber kullanılmıştır. Uzamsal alanda harekete ait bilgilerin modellenmesinde, görüntü işleme alanından elde edilen tecrübe ile önemli ilerlemeler sağlanmıştır. İnsan hareketlerin tanınması için ise zamansal bilginin modellenmesi de kritik öneme sahiptir. Son yıllarda önerilen başarılı yöntemler, hareketlerin hem zamansal hem de uzamsal yapısını birlikte modellemişlerdir. Bu tez kapsamında, karmaşık insan hareketlerinin tanınmasında, özniteliklerin zamansal dizilimlerinin histogramları isminde yeni bir model önerilmektedir. Önerdiğimiz yöntem: videolardan temel özniteliklerin çıkartılması, dizilim madenciliği ile sınıf içinde ortak olan dizilim örüntülerinin elde edilmesi, ayırt edici dizilim örüntülerinin seçilmesi, seçilen dizilimler ile videolar üzerinde histogramların elde edilmesi ve bilinen yöntemler ile sınıflandırma aşamalarından oluşmaktadır. Dizilimlerin elde edilmesi aşamasında zaman boyutunda hiyerarşik piramit yapısı kullanılarak insan etkileşimlerinin farklı zamansal ölçeklerdeki davranışları da modele dahil edilmektedir. Önerdiğimiz method farklı türdeki özniteliklerle de kullanılabilir. Bu kapsamda, önerdiğimiz yöntemin başarımını derin öznitelikler ile test ettik. Önceden eğitilmiş derin

ağları veri kümeleri üzerinde uyarladık ve elde edilen derin öznitelikleri önerdiğimiz yöntem ile denedik. Literatürde var olan UT Human Interactions ve TV Human Interactions veri kümelerinde yöntemimizin başarımını raporladık. UT Human Interactions veri kümesinde, ikinci grup üzerinde en iyi sonucu elde ettik, birinci grup üzerinde ise literatürdeki en iyi sonuçla aynı başarımı elde ettik. TV Human Interactions veri kümesinde ise literatürdeki en iyi sonucu elde ettik.

Anahtar Kelimeler: insan etkileşimleri tanıma, dizilim madenciliği, bilgisayarlı görü, video öznitelikleri

ACKNOWLEDGEMENTS

First and foremost, I would like to thank to my supervisor Assoc. Prof. Dr. Nazlı İKİLER CİNBIŞ for her valuable advice and guidance. At every stage of this thesis, she supported me with her knowledge, experience, motivation and encouragement.

Besides, I would like to thank to my thesis committee members, Prof. Dr. A. Aydın ALATAN, Assoc. Prof. Dr. İbrahim Aykut ERDEM, Prof. Dr. Pınar KARAGÖZ and Assoc. Prof. Dr. Ahmet Burak CAN for reviewing this thesis and giving insightful comments.

I would also thank to members of Hacettepe University Computer Vision Laboratory (HUCVL) for providing me a good atmosphere for working.

I would like to thank my parents Nazik and Ayhan, and also my sister Elif for supporting and encouraging me throughout my educational life.

Finally, I am deeply grateful to my wife Sümeyye for her best wishes and support.

This research was supported in part by TUBITAK Career Development Award 112E149.

GENİŞLETİLMİŞ ÖZET

Giriş

Bilgisayarlı görü alanında insan eylemlerinin tanınması problemi son yıllarda üzerinde çok çalışılan alanlardan birisidir. İnsan eylemlerinin tanınması için geliştirilen yöntemler, güvenlik, insan bilgisayar etkileşimi, spor ve eğlence gibi alanlarda kullanılabilmektedir. Bugüne kadar yapılan çalışmaların önemli bir bölümü daha basit insan eylemlerinin, bir insanın gerçekleştirdiği tek bir hareketin, tanınabilmesi üzerine odaklanmıştır. Gerçek dünya uygulamalarında ise birden çok insanın birden çok hareket içeren durumlar vardır.

İnsan eylemlerinin çeşitliliğinin çok fazla olması, insan eylemlerinin sınıflandırılması ihtiyacını ortaya koymuştur. Literatürde insan eylemlerini sınıflandırmak için birden çok sınıflandırma yöntemi önerilmiştir [1–4]. Önerilen yöntemlerin önemli bir bölümü eylemlerin zamansal olarak uzunluğunu, eylemlerin içerisindeki insanların sayısı ve tekrarlı hareketlerin olup olmaması gibi özellikleri insan eylemlerini sınıflandırmada dikkate almıştır.

Aggarwal ve Ryoo [1], insan eylemlerini karmaşıklığına göre, jest ve mimikler, insan hareketleri, insan etkileşimleri ve grup aktiviteleri olarak dört alt kategoriye ayırmıştır. Bu sınıflandırmaya göre, insan hareketleri bir insan tarafından gerçekleştirilen, birden çok alt jest ve mimikten oluşmaktadır. Koşma, yürüme ve itme hareketleri gibi tek bir kişi tarafından gerçekleştirilir. İnsan etkileşimleri, el sıkışma, sarılma ve öpüşme gibi birden fazla kişi tarafından gerçekleştirilen etkinlikleri içerir. Son olarak, grup aktiviteleri birden fazla kişi ve birden fazla nesne içerir. Toplantı, dövüş aktiviteleri grup etkinliği olarak kabul edilebilir.

Bu tez çalışması kapsamında, videolardan insan etkileşimlerinin tanınması problemi üzerinde çalışılmaktadır. İnsan etkileşimleri içeren videolarda, etkileşimleri tanımayı zorlaştıran bazı etmenler bulunmaktadır. Bu etmenlerden en önemlileri: karmaşık arka planlar, görüntü ölçeklerinin farklı olması, ortam ışığındaki farklılıklar, insanların görünümlerindeki farklılıklar, görüntüyü kaydeden kameranın hareketli olmasıdır. Tüm bu zorluklar, farklı adaptasyon yöntemleri ve dikkatlice tasarlanmış bir çözüm gerektirmektedir.

Basit ve iyi performans gösteren bir insan eylemlerini tanıma sisteminde, zamansal ve uzamsal özelliklerinin birlikte kullanılması bir gerekliliktir. Çünkü eylemler çoğunlukla hem zamansal hem de mekansal boyutlarda karakterize edilir. Örnek olarak, koşu ve yürüyüş eylemleri uzamsal olarak benzer görüntüler oluştursa da, zamansal olarak farklıdır. Bu gibi durumlarda, zamansal bilgi kullanılarak insan etkileşimleri daha iyi ayırt edilebilir. Sadece eylemlerdeki kısa süreli hareket parçalarının modellenmesi, uzun zaman aralıklarında devam eden karmaşık insan eylemlerinin ve etkileşimlerinin tanınmasında da sınırlı bir başarıya sahip olacaktır.

Etkileşimlerin daha uzun sürdüğü durumlarda, zaman boyutunda ölçeklenebilir modellere ihtiyaç vardır. Eylemlerin zamansal boyuttaki özelliklerinin modellenmesi için, mevcut çalışmaların çoğu, ilgili eylem türünü temsil eden, ortak alt parçaları bulmaya odaklanmıştır [2, 5]. Bu yaklaşımlar, olası alt video parçaları arasında en iyi bölümleri bulsa bile, hareketi oluşturan bileşenler arasında zamansal boyutta bölünmeler olabileceği için bilgi kaybına neden olabilir. Bu nedenle, eylemlerin zamansal boyuttaki özelliklerini daha iyi modelleyen yöntemlere ihtiyaç olduğuna düşünüyoruz.

Etkileşimlerin tanınabilmesi için bir arada bulunan yerel özniteliklerin kullanılması da önemli bir ayırt edici bilgi kaynağıdır. Şu anda literatürdeki başarılı yöntemler [6, 7], yerel özniteliklerin göreceli konumlarını göz ardı etmektedir. Bununla birlikte, özellikle görsel açıdan benzer eylemler arasında ayırım yapmak istendiğinde, bu özelliklerin göreceli zamansal konumları da önemlidir. Örneğin “oturmak” ve “kalkmak” eylemleri uzamsal boyutta benzer özelliklere sahip olsa da, zamansal alanda oldukça farklı özellikleri vardır.

Bu tez kapsamında, insan etkileşimi tanıma açısından yararlı olabilecek yeni bir yöntem öneriyoruz. Önerilen yöntem, karmaşık eylemlerin tanınması için de kullanılabilir. Önerdiğimiz yöntem, videoları *yerel öznitelik dizilimlerinin* histogramı olarak kodlar. Dizilimler, videolardan elde edilen yerel özniteliklerin, kümeleme yöntemi ile görsel kelimelere dönüştürülmesi ile oluşturulur. Görsel kelimeler videolarda hangi zamanda geçiyorsa, o zaman etiketi ile artarda gelerek, videoyu temsil eden dizilimi ortaya çıkarır. Görsel kelimelerden çok fazla

sayıda dizilim elde etmek mümkündür, bu sebeple her sınıfa ait sık geçen dizilimlerin bulunması önemlidir. Bu dizilimler arasında, ilgili sınıfa ait sık görülen dizilimler, bir veri madenciliği yöntemi olan *dizilim madenciliği* yöntemleri vasıtasıyla bulunur. Ayrıca aktivite sınıfları arasında daha iyi ayırt edici dizilimleri bulmak için, bir dizilim seçme işlemi uygulanır. Sonunda, her bir video, ayırt edici, sık görülen dizilimlerin bir histogramı olarak temsil edilir. Bu modelin bir uzantısı olarak, Choi ve diğ.'de [8] olduğu gibi bir zamansal piramit yapısı oluşturularak, daha zengin bilgi içeriğine sahip modeller oluşturulur. Zamansal piramidin yüksekliği, dizilerin zamansal ölçeklerini belirler ve her farklı seviyede zamansal olarak farklı çözünürlükteki eylemlere ait bilgiler modellenir.

Literatürde Var Olan Yöntemler

Literatürde insan etkileşimlerini ve eylemlerini tanıma problemi üzerinde, bilgisayarlı görüş alanında pek çok yöntem bulunmaktadır. Bu yöntemler üç başlık altında gruplanabilir. Birinci grupta, yerel öznitelik noktalarının ayırt edici sınıflandırma yöntemleri ile kullanılmasından oluşmaktadır. İkinci grupta, hareketin zamansal dinamik modelleri oluşturulmaya çalışılmaktadır. Bayesçi yöntemler, Gizli Markov modeller, sonlu durumlu makinalar kullanılabilir. Üçüncü grupta ise, anahtar bileşenlerden oluşmaktadır. Hareketi tanımlayan görüntü ve/veya öznitelik bölümlerinin zamansal olarak sırasını analiz etmektedirler. Bir de son yıllarda, grafik işlemci birimlerinin kullanılabilmesi ile beraber derin öğrenme yöntemleri ortaya çıkmıştır. Derin öğrenme yöntemlerinde videolar önerilen bir derin ağ yapısında eğitime tabi tutularak, öznitelikler öğrenilmektedir.

Yerel öznitelik bilgilerinin kullanıldığı pek çok yöntem [7, 9–18] bulunmaktadır. Bu yöntemlerde, yerel özniteliklerinin farklı modeller ile bir araya getirilerek videolar sınıflandırılmaktadır. Bu yöntemlerden en temeli, uzay-zaman ilgi noktalarıdır [9]. Uzay-zaman ilgi noktaları zamansal ve uzamsal olarak yüksek değişime sahip olan noktalardır. Bu noktalar etrafında ya hareket ya da kenar/köşe bölgesi olduğundan, hareketlerin modellenmesi için önem arz etmektedirler.

Uzay-zaman ilgi noktaları etrafından elde edilen yerel öznitelikler daha görsel kelime kümeleri yöntemleri ile kullanılmıştır [19]. Görsel kelimeler, yerel özniteliklerin kümeleme algoritmaları ile kümelenmesinden sonra, özniteliklerin elde edilen kümeler cinsinden ifade edilmesidir. Videolar elde edilen görsel kelimelerin histogramı olarak temsil edilir. Tespit edilen uzay-zaman ilgi noktalarında en çok kullanılan öznitelikler Histogram of Oriented Gradients (HOG) [12], Histogram of Optic Flows (HOF) [12] ve Motion Boundary Histogram (MBH) [12] öznitelikleridir. HOG özniteliği, resim üzerindeki bir bölgede bulunan görüntüyü, kenar/köşe cinsinden kodlamaktadır. HOF özniteliği, videolar üzerindeki yerel bir bölgedeki yerel hareketi (motion) kodlamaktadır. Bilgi kaynağı olarak optik akış bilgisi kullanılır. MBH özniteliği ise, HOF özniteliğine benzemektedir, doğrudan optik akış bilgisinin kullanılması yerine, optik akış bilgisinin türevi kullanılmaktadır. Bu sayede kamera hareketlerinden kaynaklanan, yanlış optik akış hesaplama problemine karşı bir miktar daha gürbüzdür.

Doğrudan görsel kelimelerin histogramlarını kullanmak yerine, görsel kelimeler arasındaki uzamsal ve zamansal ilişkiyi analiz eden yöntemlerin daha başarılı olduğu görülmüştür. Bu kapsamda yapılan çalışmalarda, çalışmaların bir kısmı [13–15], bir arada bulunan yerel özniteliklerini tespit etmiş, bir arada bulunma bilgisinin başarılı şekilde ayırt etme yeteneğinin olduğu anlaşılmıştır. Ayrıca, yerel ilgi noktalarının bir arada bulunma bilgisinin kullanımına ek olarak, videolardaki dağılımını bir çizge ve/veya çizge benzeri yapılar ile modelleyen yöntemler [17, 18] de bulunmaktadır.

Literatürde bulunan çalışmalardan, ikinci grupta, hareketin dinamik modelini oluşturmaya çalışan yöntemler [2, 5, 6, 14, 20–23] bulunmaktadır. Bu gruptaki yöntemlerin önemli bir bölümü, videolardaki uzun süreli izleri(trajectory) kullanmaktadır [6, 20–23]. Videolarda izleri oluşturmak için uzay-zaman noktaları, insan görüntüleri, insan kafası gibi değişik seviyedeki bilgiler takip edilip, takip edilen iz boyunca oluşturulan öznitelikler ile videolar sınıflandırılmıştır. Bu gruptaki diğer çalışmalar [2, 14] ise, videolardaki hareketi temsil eden en iyi zamansal parçaları bulmaya çalışmışlardır. Videolar eşleştirilirken sisteme esneklik katmak için, elde edilen zamansal konumlar bir Gauss Dağılımından çıkmış gibi varsayılmıştır. Bu çalışmalar ek olarak, videoları zamansal boyutta alt parçaların oluşturduğu ağaç vb. yapılar ile temsil eden yöntemler bulunmaktadır [5, 24, 25]. Bu yöntemlerde

videolardan, elde edilen bölütler arası ilişkiler incelenmiş, sık geçen ve ayırt edici segment grupları kullanılmıştır. Bu grupta ayrıca, insan hareketlerini sonlu durumlu otomatalar ile tanıyan yöntem vardır [26]. Hareketi oluşturan alt parçaların oluşturduğu ilişkiler, Gizli Markov Modelleri ile modellenmiştir. Test örnekleri sınıflandırılırken, öğrenilen sonlu durumlu otomatalar test örnekleri ile eşleştirilmektedir.

Literatürdeki bulunan çalışmalardan, üçüncü grupta, anahtar bileşenleri bulan yöntemler [27–30] vardır. Bu yöntemlerde, el ile seçilmiş video çerçevelerinden başlayarak, farklı öğrenme yöntemleri ile videolara ait ayırt edici çerçeveler seçilmiştir.

Literatürde bulunan bu üç gruba ek olarak, son yıllarda derin öğrenme yöntemleri [31–35] de insan hareketlerinin tanınması için kullanılmıştır. Bu yöntemlerde farklı mimarilerdeki derin ağlar ile uzamsal ve zamansal öznitelikler öğrenilmektedir. Yöntemlerden bazıları [33, 35] uzamsal ve zamansal bilgiyi, ayrı ayrı öğrenip daha sonradan birleştirmekte, diğer kısmı ise aynı anda hem uzamsal hem de zamansal [31] (3B) öznitelikleri öğrenmektedir. Yöntemlerden bir kısmı [32, 34] ise farklı çözünürlükte bilgiyi kullanmaktadır, yani hem zamansal ve uzamsal bilgiyi ayrı ayrı hem de beraber farklı çözünürlüklerde kullanmaktadır.

Önerilen Yöntem

Önerdiğimiz yöntem aşağıdaki adımlardan oluşmaktadır:

1. Yerel özniteliklerin elde edilmesi
2. Görsel kelimelerin oluşturulması
3. Videoların yerel özniteliklerin dizilimi olarak temsil edilmesi, bu aşamada hiyerarşik piramit yapısının oluşturulması,
4. Dizilim madenciliği ile sık geçen dizilimlerin elde edilmesi
5. Elde edilen dizilimlerden ayırt edici olanların seçilmesi
6. Videoların ayırt edici ve sık geçen örüntülerin histogramı ile temsil edilmesi

7. Elde edilen histogramlar ile sınıflandırma.

Videolardan yerel özniteliklerin elde edilmesi aşasında literatürde var olan uzay-zaman ilgi noktaları [9] kullanılmıştır. Elde edilen ilgi noktaları üzerinden HOG, HOF veya MBH öznitelikleri oluşturulmuştur. Sistemin yerel özniteliklerin dışında derin öznitelikler ile de çalışabildiğini göstermek adına, derin öznitelikler de [33] videolardan ayrıca elde edilmiştir. Yerel özniteliklerin kamera hareketinden yoğun şekilde etkilenmesi sebebi ile videolarda bölütleme algoritması [36] kullanılarak, harekete ait olmayan öznitelikler filtrelenmiştir. Filtreleme yapılırken, uzamsal yayılımı en geniş olan bölüt, arka plan olarak kabul edilmiş, arka plana düşen öznitelikler filtrelenmiştir. Bu yöntem her ne kadar mükemmel olmasa da, harekete ait olmayan pek çok sayıda özneliğin başarılı şekilde filtrelenmesini sağlamıştır.

Hesaplanan öznitelikler daha sonra görsel kelimelere dönüştürülmüştür. Görsel kelimeler oluşturulurken, öznitelikler k-means kümeleme algoritması ile kümelenecek, öznitelikleri küme merkezlerine olan uzaklığına göre görsel kelimeler elde edilmiştir. Bu aşamada her bir özneliğin tek bir küme merkezine eşleştirilmesi bilgi kaybına neden olduğundan, öznitelikler bir den çok küme merkezi ile uzaklık ilişkisine göre eşleştirilmiştir.

Videolarda geçen öznitelikler görsel kelimelere dönüştürüldükten sonra, içerisinde bulundukları çerçeve numarası ile etiketlenerek dizilimler oluşturulmuştur. Bu aşamada videolar, zamansal olarak hiyerarşik düzende bir den çok, farklı sayıda parçaya bölünmüştür. Örneğin, 6 seviyeli bir hiyerarşik piramit yapısında, videolar sırasıyla değişik seviyelerde 2, 3, 5, 8, 13 ve 21 eş parçaya bölünerek, her parçaya ait dizilimler oluşturulmuştur. Birinci seviyede yer alan, ilk parçada bulunan özniteliklerin zaman etiketi 1, ikinci parçadakilerin zaman etiketi ise 2 olmuştur. Yine benzer şekilde, altıncı seviyede, video 21 eş parçaya bölünmüş ve öznitelikler zamansal olarak geçtiği konuma göre, 1'den 21'e kadar zaman etiketi ile etiketlenmiştir.

Literatürde bulunan dizilim madenciliği yöntemi [37] ile elde edilen dizilimler üzerinde, her sınıf için ayrı ayrı dizilim örüntüleri elde edilmiştir. Örüntüler elde edilirken her sınıftan yaklaşık aynı sayıda örüntü elde edilecek şekilde parametreler kullanılmıştır. Bir sınıf için bulunan sık geçen bir örüntünün diğer sınıflarda da sık geçmesi, bu örüntünün yeterince

ayırt edici olmamasına sebep olduğundan dolayı, örüntüler içerisinde bir seçme işlemi uygulanmıştır. Seçme işlemi doğrusal *SVM* [38] kullanılarak, *SVM* ağırlığı en düşük dizilim örüntülerinin filtrelenmesi ile gerçekleştirilmiştir.

Daha sonra elde edilen dizilim örüntüleri videolarda aranarak, videolardaki bulunma sayıları ile histogram temsili oluşturulmuştur. Elde edilen histogramlar yine *SVM* [38] kullanılarak *one-vs-all* çok sınıflı sınıflandırma yöntemi ile sonuçlar elde edilmiştir.

Önerdiğimiz yöntemi iki farklı veri kümesi: *TV Human Interactions* [39] ve *UT Human Interactions* [40] üzerinde performansını deneyip, bu veri kümeleri üzerinde en iyi başarıyı gösteren diğer yöntemler ile karşılaştırılmıştır. Yöntemimizde kullanılan parametrelerin seçerken, eğitim kümesi üzerinde, yaptığımız denemeler sunulmuştur. Elde edilen sonuçlara göre, yöntemimiz *TV Human Interactions* veri kümesinde en iyi başarıya veya en iyi sonuçlarla benzer sonuca, *UT Human Interactions* veri kümesinde ise literatürdeki en iyi sonuçlarla benzer başarıya sahip olduğu görülmüştür.

Yerel özniteliklerin önerdiğimiz yöntem ile kullanımına ek olarak, derin öznitelikleri de yöntemimizde denedik. Bu kapsamda, daha önceden eğitilmiş olan [33] zamansal ve uzamsal VGG-16 derin ağlarını veri kümesi üzerinde uyarladık. Derin ağın FC6 ve FC7 katmanlarından öznitelikleri çıkarttık. Buradan elde edilen örüntüler ile sonuçları raporladık. Bu çalışmaya ek olarak, derin ağın daha farklı katmanlarından da (pool3_3, pool4_3, pool5_3) elde edilen özniteliklerin bir arada kullanılmasının etkisini de inceledik.

Önerdiğimiz yöntem ile dizilim madenciliği sonucu elde edilen örüntülerin ayırt edici olduğu anlaşılmıştır. Hiyerarşik piramit yapısı kullanıldığında, insan etkileşimlerinin farklı zamansal ölçeklerdeki yapısının daha iyi modellenerek, başarımın artırıldığı görülmüştür. Önerilen algoritma *IET Computer Vision* dergisinde yayınlanmıştır [41].

CONTENTS

	<u>Page</u>
ABSTRACT	i
ÖZET	iii
ACKNOWLEDGEMENTS	v
GENİŞLETİLMİŞ ÖZET	vi
CONTENTS	xiii
FIGURES	xix
TABLES	1
1 INTRODUCTION.....	1
1.1 Motivation	1
1.2 Contribution of the Proposed Method	4
1.3 Organization of the Thesis	4
2 BACKGROUND	6
2.1 Local Interest Points.....	6
2.1.1 Spatial Interest Points	6
2.1.2 Spatio-Temporal Interest Points	6
2.2 Bag of Words	8
2.3 Feature Descriptors	9
2.3.1 Histogram of Oriented Gradients (HOG).....	9
2.3.2 Histogram of Optic Flows (HOF).....	10
2.3.3 Motion Boundary Histogram (MBH).....	11
2.4 Support Vector Machines (SVM)	12
2.5 Deep Learning	12
2.6 Sequence Mining	15
2.7 Problem Definition	15
2.7.1 Example Execution of Sequence Mining Algorithm	20
2.8 Histogram Computation with Sequential Patterns	21
3 RELATED WORK	24

3.1	Local Interest Point Methods	24
3.1.1	Bag of Words and Interest Points	24
3.1.2	Co-Occurrence of Interest Points	24
3.1.3	Graph of Interest Points	27
3.1.4	Spatial and Temporal Relationship of Interest Points.....	28
3.2	Models Focused on Temporal Dynamics.....	29
3.3	Models Focused on Key Components	35
3.4	Deep Learning Methods.....	35
3.5	Mining Methods used in Computer Vision.....	37
4	HISTOGRAM OF SEQUENCES	40
4.1	Extracting Local Features	40
4.1.1	Filtering Interest Points	40
4.2	Extracting Deep Features	42
4.3	Visual Word Construction.....	44
4.4	Hierarchical Temporal Pyramid Structure.....	45
4.5	Mining Sequential Patterns	46
4.6	Calculation of Histogram of the Sequential Patterns	47
4.7	Sequential Pattern Selection	48
4.8	Building Up Histogram of Sequential Patterns	49
4.9	Classification	52
4.10	Deep Multi-Layer Histogram of Sequences.....	52
4.10.1	Features.....	53
4.10.2	Visual Word Formation	54
4.10.3	Hierarchical Temporal Pyramid Structure	55
4.10.4	Mining Sequential Patterns	55
4.10.5	Histogram Calculation and HoS Representation	55
5	EXPERIMENTAL EVALUATION.....	56
5.1	Datasets	56
5.2	Experimental Details	57
5.3	Evaluation Methods	59

5.4	Results on TV Human Interactions Dataset	60
5.4.1	Test with Single Level HoS with Local Features	60
5.4.2	Test with Hierarchical HoS with Local Features	62
5.4.3	Test with Hierarchical HoS with Deep Features	65
5.4.4	Comparison of the Best HoS Method with the State of the Art	73
5.5	Results on UT Human Interactions Dataset	75
5.6	Time and Space Complexity of the Proposed Method.....	76
5.6.1	Runtime Environment	77
5.6.2	List of References to Available Software Implementations Used in This Thesis...	78
6	DISCUSSION	79
6.1	Future Work.....	79
	REFERENCES	81

FIGURES

	<u>Page</u>
Figure 1.1 Categorization of human actions.....	2
Figure 1.2 Example human interaction scenarios.	2
Figure 2.1 Spatial Interest Points [42].	6
Figure 2.2 A representation of Space-Time Interest Points (STIPS) [9].	7
Figure 2.3 Space-Time Interest Points [9].	8
Figure 2.4 Histogram of visual words.	8
Figure 2.5 Different quantization techniques for assignment of features to cluster centers.	9
Figure 2.6 Example HOG Descriptors Extracted for 24x24 Patches.....	10
Figure 2.7 Setting up the histogram with the gradient data.....	10
Figure 2.8 An example optical flow.....	11
Figure 2.9 An example optical flow when there is a camera motion.....	11
Figure 2.10 Support Vector Machine - Classification of Non-Linear Problems. ...	12
Figure 2.11 VGG-16 network, input to output	13
Figure 2.12 A representation of VGG-16 deep network.	14
Figure 2.13 Sample mining steps for input database Table 2.1.	22
Figure 3.1 Spatio-temporal phrase representation [13].	25
Figure 3.2 A representation of 3-D volumes used for modeling co-occurrence information [14].	26
Figure 3.3 The representation of the “String of Feature Graph”model.	28
Figure 3.4 Video matching with dynamic time warping.....	28
Figure 3.5 A representation of Ryoo and Aggarwal’s [18] model.	29
Figure 3.6 A representation of dynamic temporal scale selection method.....	30
Figure 3.7 A representation of motion phrases.....	31
Figure 3.8 A representation of dense trajectories [20].	31

Figure 3.9	A representation of person-centered descriptor [21].	32
Figure 3.10	A representation of two stream model.	35
Figure 3.11	A representation of Feichtenhofer et al.'s [33] model.	36
Figure 4.1	Overall framework of the proposed method.	41
Figure 4.2	Example frames for the STIP filtering step on the TV Human Interaction dataset [39].	42
Figure 4.3	An example optical flow image from TV Human Interactions Dataset.	43
Figure 4.4	An example representation of temporal pyramid structure.	46
Figure 4.5	Pictorial representation of the discovered sequential patterns (when the temporal pyramid is not used).	47
Figure 4.6	Pictorial representation of the discovered sequential patterns (when the temporal pyramid is used).	48
Figure 4.7	Example patterns that are discovered using our approach from the UT Interactions dataset (Figure 4.7(a)- 4.7(b)) and TV Human Interactions dataset (Figure 4.7(c)- 4.7(d)).	50
Figure 4.8	Additional discovered sequential patterns.	51
Figure 4.9	Deep multi-layer Histogram of Sequences representation.	54
Figure 5.1	Example snapshots from the TV Human Interactions dataset [39].	57
Figure 5.2	Example snapshots from the UT Human Interactions dataset [39].	57
Figure 5.3	Local Features - The Effect of number of candidate sequential patterns when the temporal pyramid is not used on TV Human Interactions dataset.	60
Figure 5.4	Local Features - Effect of number of selected sequential patterns when the temporal pyramid is not used on TV Human Interactions dataset after the sequential pattern selection step.	61
Figure 5.5	Local Features - The effect of number of candidate sequential patterns for each class at every level of the pyramid for TV Human Interactions dataset.	62

Figure 5.6	Local Features - The effect of number of selected sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class after the sequential pattern selection step.....	63
Figure 5.7	Local Features - The effect of the number of pyramid levels on the recognition performance on TV Human Interactions dataset. See text for details	64
Figure 5.8	Local Features - Classwise mAP performance of the proposed HoS representations, together with its extensions on the TV Human Interactions dataset.	65
Figure 5.9	Deep Spatial Features - The effect of number of candidate sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class.....	66
Figure 5.10	Deep Spatial Features - The effect of number of selected sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class.....	67
Figure 5.11	Deep Spatial Features Extracted using Sliding Windows - The effect of number of candidate sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class.....	68
Figure 5.12	Deep Spatial Features Extracted using Sliding Windows - The effect of number of selected sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class.	68
Figure 5.13	Deep Spatial Features Extracted from Multiple Layers of Network - The effect of number of selected sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class.	70
Figure 5.14	Deep Temporal Features - The effect of number of candidate sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class.....	70
Figure 5.15	Deep Temporal Features - The effect of number of selected sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class.....	71

Figure 5.16 Classwise mAP performance of the HoS method with deep spatial features on the TV Human Interactions dataset.....	73
--	----



TABLES

Table 2.1	Example input sequences for the sequence mining algorithm.....	20
Table 2.2	Extracted sequential patterns from example input of Table 2.1.	20
Table 5.1	Performance of the Single Level HoS method using the local features on the TV Human Interactions dataset when the negative videos are included	61
Table 5.2	Performance of the single level HoS method using the local features on the TV Human Interactions dataset when the negative videos are excluded.....	62
Table 5.3	Performance of the HoS method using the local features on the TV Human Interactions dataset when the negative videos are included	64
Table 5.4	Performance of the single level HoS method using the local features on the TV Human Interactions dataset when the negative videos are excluded.....	64
Table 5.5	Performance of the HoS method using the deep features on the TV Human Interactions dataset when the negative videos are included	72
Table 5.6	Performance of the HoS method using the deep features on the TV Human Interactions dataset when the negative videos are excluded. ...	73
Table 5.7	Performance comparison of the HoS method using the local features on the TV Human Interactions dataset when the negative videos are included	74
Table 5.8	Performance comparison of the HoS method using the local features on the TV Human Interactions dataset when the negative videos are excluded.....	75
Table 5.9	Comparison of the proposed method with the state of the art methods on the UT Interaction dataset.	75
Table 5.10	Confusion table of UT Interactions dataset for HoS representation is used with soft clustering and Temporal Pyramid.....	76

1 INTRODUCTION

1.1 Motivation

Human action recognition problem is to understand what people are doing and it is an evolving topic in the computer vision community. The human action recognition methods can be used in various application domains such as surveillance, human-computer interaction, sport and entertainment analysis, and more...

Most of the research up to date has focused on the analysis of singleton human actions, where only a single person is performing a single action. However, real-world cases can be more complex. Complex activities are taking place in videos, and an important portion of possible human activities are human interactions, where more than one person is involved in a particular activity.

Since the diversity of the actions is so much to handle, actions have been grouped into different classes. In literature, there are many different types of classifications can be found, but most of them shares the common two features which are the duration of the action, periodicity of the action and the number of people or objects in the action. Recent studies [1–4] have been proposed to categorize the human actions. Aggarwal and Ryoo [1] categorized the human actions into four subcategories according to the complexity of the actions as: gestures, actions, interactions, and group activities. The gestures include the elementary motion of body parts. The actions consist of multiple gestures and performed by a single person, such as running, walking and pushing. The interactions include the activities that performed by multiple people such as handshaking, hugging and kissing. Finally, the group activities include multiple people and multiple objects. Meeting, fighting activities can be accepted as a group activity.

Niebles et al. [2] categorize the human actions in five categories according to the duration in the temporal domain as shown in Figure 1.1. According to the this categorization, the method for recognizing a “Snapshot” category should have different representation for “Long term event” due to the inherent structure of the actions.

In this thesis work, we aim to look at the problem of recognizing complex human actions, human-human interactions in particular, where the videos are taken in uncontrolled settings, and more than one actor, most of the time two people, are involved in the interaction. In this

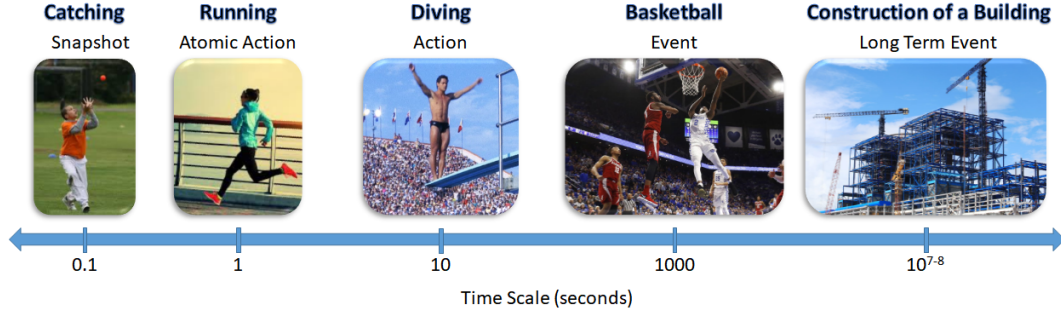


FIGURE 1.1: Categorization of human actions.

problem, there are many challenging issues to be dealt with. The videos have occlusions, variations in the appearance of the people, moving camera, scale variations, changes in light, cluttered backgrounds, and more. The Figure 1.2 shows example interaction scenarios those have clutters, occlusions variance in appearance of the people and more. These issues require many different adaptation techniques and a carefully designed framework.



FIGURE 1.2: Example human interaction scenarios.

For a successful activity recognition system, modeling spatial structures together with the temporal structures is a requirement, since the activities differ either by the differences in the spatial domain, temporal domain or in both of the domains. As an example, the running and walking actions can be better discriminated by using temporal information. Modeling short-term structure may have limited representative power in recognition of complex actions and interactions, which are sustained on longer time periods.

A temporally scalable model to recognize the used in long duration videos is needed. When we look at the existing models, most of the previous work [2, 5] focused on segmentation in finding representative common subsequences in the temporal domain, over the different types of actions, or finding one to one temporal matching between the basic sub-components of the actions [13, 17, 18]. Such approaches can cause information loss due to the presence of splits, or variation in the execution speed of the actions, even if the method finds best partitions or matches among the possible ones. Therefore, we think that a more general framework that does not propagate the error introduced in simple action matching phase is

would be beneficial. This framework should handle the whole sequence more effectively in the temporal domain.

The co-occurrence of the local features may also indicate important cues for accurate recognition of the interactions. Again, current successful representations [6, 7] based on bag-of-words of local features, mostly ignore the relative position of these features. However, the relative temporal locations of these features are important, especially if one wants to differentiate between visually similar actions, which have quite different characteristics on the temporal domain, e.g. “sitting up” and “sitting down” actions.

In this thesis, we propose a new feature representation for human interaction recognition. The suggested method can also be used for the recognition of complex actions. This feature representation encodes the videos as a *histrogram of local feature sequences*. The sequences correspond to the time-ordered list of the visual vocabulary elements where the dictionaries are constructed over the space-time interest points. Among these sequences, the frequently occurring sequences are found by means of a sequence mining approach. We further apply sequence selection for better discrimination between activity classes. In the end, each video is represented as a histogram over the selected discriminative frequent sequences.

As an extension to this model, we incorporated a temporal pyramid mechanization as in [8]. The temporal pyramid structure is a representation to better encode the temporal information in videos.

This new representation is likely to cover more complex spatio-temporal relationships by means of mining sequences of local interest points that frequently occur for each interaction at different time scales. The height of the temporal pyramid determines the temporal scales of the sequences. We think that the proposed method can also be used for recognition of human action recognition problems in addition to the human interaction problems, since there can be many sequential patterns between humans and the objects in the actions while performing actions.

We have also tested the new method with deep spatial and temporal features as an another extension the baseline model. The deep features are the best performing features in the recent studies for complex and hard recognition problems [35, 43–46].

1.2 Contribution of the Proposed Method

In this thesis, our contribution can be defined as following items:

1. We propose a novel representation, histogram of sequences (HoS) by combining sequence mining methods and local interest point methods to model the human interactions, we show that our final set of sequences not only frequent but also discriminative.
2. We showed that mining the frequently occurring sequential interest point patterns, and then representing this information by means of a histogram encoding has good recognition performance for human interaction recognition problem.
3. The performance of the proposed technique on two publicly available benchmark datasets are provided together with the comparison with the state of the art methods.
4. We show that combining sequence mining method with temporal pyramid structure (Section 4.4) provides better performance by capturing temporally multi-granular patterns.
5. We show the use of deep features with the proposed method to show the use of different features with the proposed method and propose a multi-layer version of HoS representation.
6. We have included a review of the state of the art methods into the thesis. The performance of more than 20 methods is compared for two publicly available datasets and compared the performance of our method with the recent state of the art methods

1.3 Organization of the Thesis

The thesis is organized as follows:

In Chapter 2, we provide a brief description of fundamental information used for action recognition in the computer vision area. We give a brief definition for data mining idea and give a formal definition of the sequence mining problem. As a supplement to this section, example sequence mining algorithm execution steps for further understanding of the algorithm are provided.

Chapter 3, we review existing human interaction and action recognition methods in the literature. Also, we provide a review of data mining methods used in the computer vision literature.

Chapter 4, we describe the HOS representation in detail step by step. The algorithms and pictorial representations are provided.

Chapter 5, we provide the experimental test results on the two publicly available datasets: UT Human Interactions [40] and TV Human Interactions 5.1 with quantitative measures. We compare the performance of the proposed method with the state of the art results in this chapter.

Finally, in Chapter 6, we discuss the outcomes of the proposed method and the future work that may be the direction to follow.

2 BACKGROUND

Recognition of human actions has been a promising topic for many years and there are many different methods proposed. For this purpose, in this section, we describe the basic background information that is used almost all across the action recognition community.

2.1 Local Interest Points

2.1.1 Spatial Interest Points

The local interest point methods focus on the points where the image have higher variation. The edges and corners are the most likely positions that can be detected as the local interest points.

The detected positions are assumed as key locations to discriminate an action from another. The locality and small scale of the interest points makes them very successful when the background changes or there is a occlusion.

The local interest points have robustness to the affine transformations such as rotation, scale and translation. The Figure 2.1² shows the example spatial interest points extracted with the method of Mikolajczyk and Schmid [42].



FIGURE 2.1: Spatial Interest Points [42].

2.1.2 Spatio-Temporal Interest Points

In the action recognition area, the temporal information has an importance in addition to the spatial information as we describe in Section 1.1. In some scenarios, even if there is no spatial change in the frames, the change in the temporal position of the object/parts means a different action.

²The Figure 2.1 is reprinted by permission from Springer Nature: Springer eBook [42], © Springer-Verlag Berlin Heidelberg (2001).

To model such information, space-time interest points (STIP) has been proposed by Laptev [9]. The STIP method uses similar method with the [42] and extends the 2-dimensional algorithm to the 3-dimensional space. In the STIP algorithm, local 3-dimensional regions having large variations along spatial and the temporal directions are detected as the space-time interest points.

Extraction of STIPs involves a space-time extension of the Harris Detector [47] and is characterized by a high variation of the image values in space and non-constant motion over time. The detected points have significant variation in the space-time neighborhood. For each video point, the spatio-temporal second-moment matrix is computed using independent temporal and spatial scale values. The Figure 2.2¹ shows the detected space-time interest points that have different spatial and temporal scales. The interest points are shown as ellipsoids.

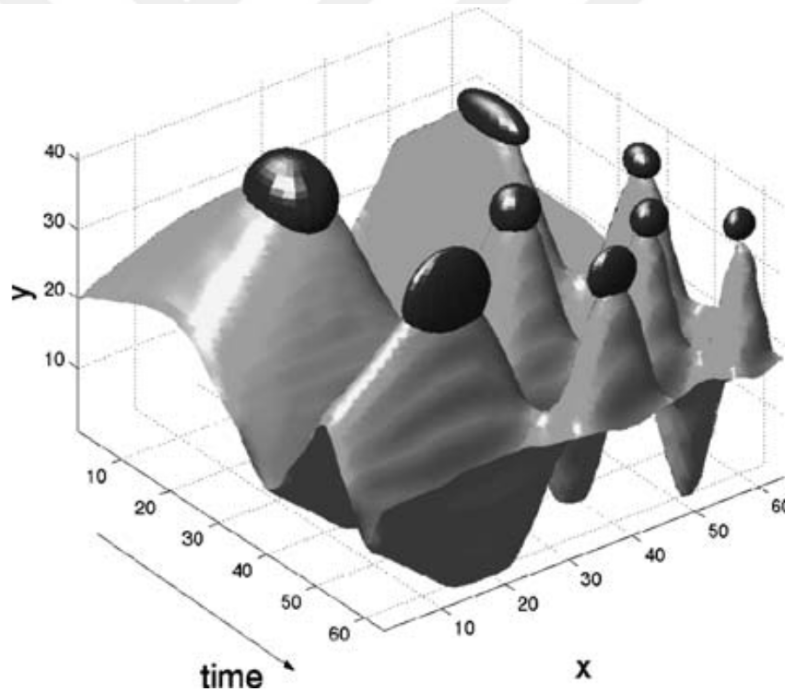


FIGURE 2.2: A representation of Space-Time Interest Points (STIPS) [9].

An automatic spatio-temporal scale selection for the space-time interest points are provided by finding the maximum of normalized Laplace operator from set of spatio-temporal scales.

¹The Figure 2.2 is reprinted by permission from Springer Nature: Springer eBook [9], © Springer-Verlag Berlin Heidelberg (2005).

The Figure 2.3¹ shows the example space-time interest points extracted with the method of Laptev [9].



FIGURE 2.3: Space-Time Interest Points [9].

2.2 Bag of Words

The bag-of-words method is based on the text-retrieval community. The idea is to create a *codebook* to represent videos as a histogram of codebook elements by mapping each interest point to a codebook element [19]. To create a codebook, at first, descriptors are used to model the interest points. The most used descriptors are Histogram of Oriented Gradients (HOG) [12], Histogram of Optic Flows (HOF) [12] and Motion Boundary Histogram (MBH) [12]. The codebook is created clustering the local feature descriptors. Then, each local feature point is assigned to a cluster center by using similarity metrics to form a histogram representation. The Figure 2.4 shows an example bag of visual words representation.

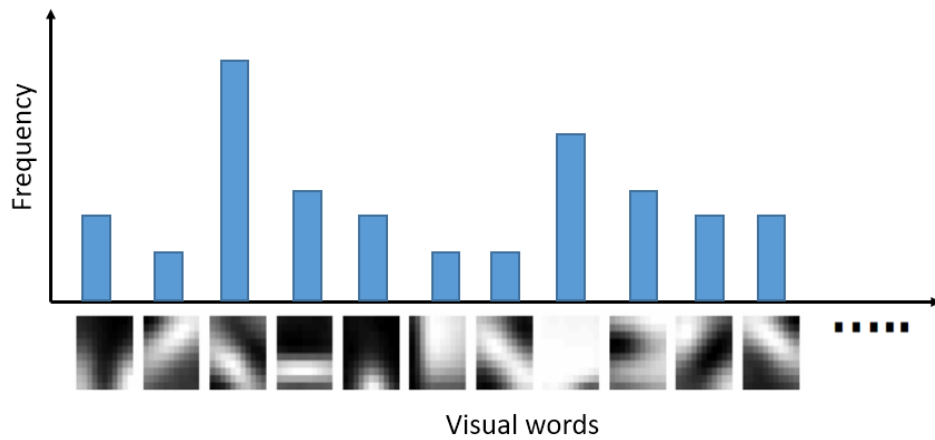


FIGURE 2.4: Histogram of visual words.

It has been shown that hard-assignment based quantization suffers significantly from the loss of information. When each data element is assigned to exactly one cluster, it is possible

¹The Figure 2.3 is reprinted by permission from Springer Nature: Springer eBook [9], © Springer-Verlag Berlin Heidelberg (2005).

to lose the similarity information between the data point and the other clusters. Several techniques such as sparse coding [48], locality-constraint linear coding (LLC) [49] have been proposed to deal with this situation. Instead of assigning a feature to a single cluster center, encoding the information between a feature and multiple cluster centers are shown to be better. The Figure 2.5 shows three different methods to assign features to cluster centers.

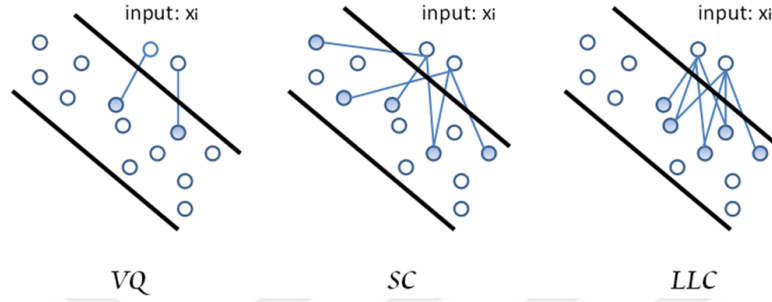


FIGURE 2.5: Different quantization techniques for assignment of features to cluster centers.

2.3 Feature Descriptors

2.3.1 Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients [50] (HOG) is the mostly used descriptor to model the spatial structure of an interested location in an image. The HOG descriptor was proposed to represent local appearance. The HOG descriptor comprises a 72-bins descriptor describing the local appearance, where a $(3 \times 3 \times 2)$ grid is utilized in the surrounding space-time area of interest and for each cell of the grid, 4-bin histogram of oriented gradients is computed. The Figure 2.6 shows an example HOG descriptors extracted for an image. As we can see from figure, the descriptor encodes the direction and the magnitude of the edges.

The gradients are calculated from the images as the spatial derivatives in different directions so that the value of the gradient is greater around the corners or edges. Since the gradients are calculated using the intensity values of the images, the output is sensitive to the lighting. The darker images result less gradient values. To gain some level of robustness to the variations in lighting, the intensity values of the images are normalized, generally between $[0, 1]$.

The histogram representation from the gradients are calculated by accumulating the magnitudes of the gradients into the direction bins. The Figure 2.7 shows an example process to calculate the histogram using the gradient data. In this example, there are 4 direction bins used and the final HOS descriptor is the concatenation of histograms within a local area.

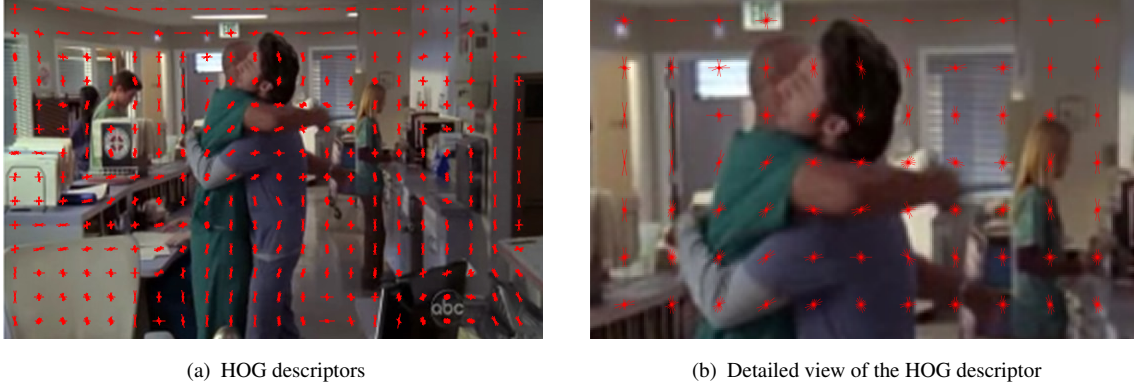


FIGURE 2.6: Example HOG Descriptors Extracted for 24x24 Patches.

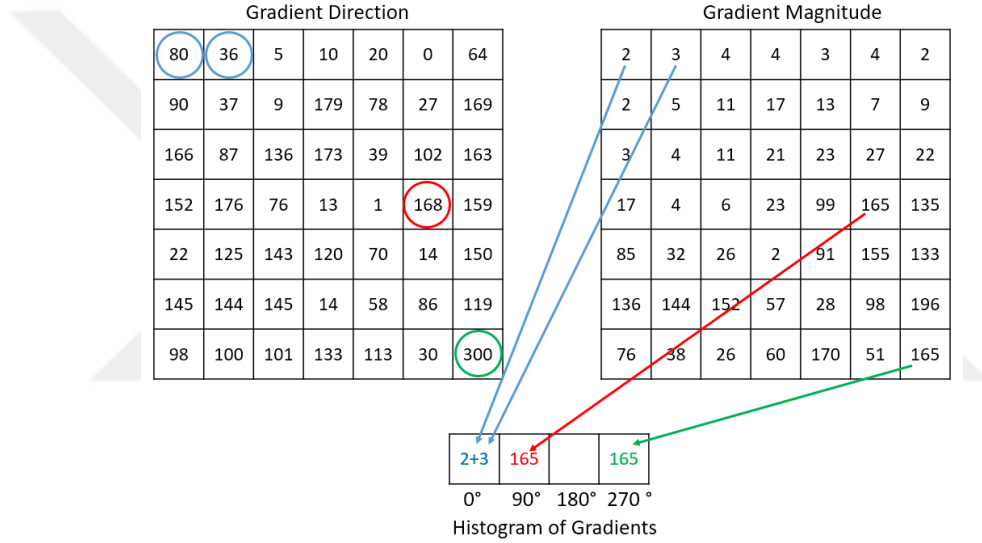


FIGURE 2.7: Setting up the histogram with the gradient data.

2.3.2 Histogram of Optic Flows (HOF)

Histogram of Optic Flows [12] (HOF) is the mostly used descriptors to model the temporal structure of an interested location in a video frame. The HOF descriptor was proposed to represent local motion. The HOF descriptor comprises a 90-bins descriptor for encoding the local motion. In the formation of this descriptor, a $(3 \times 3 \times 2)$ grid is placed around the encompassing space-time area and a 5-bins histogram of optical flow is computed for each cell of the grid. The Figure 2.8 show the calculated optical flows from a video.

The flows are calculated using the temporal differences between the consecutive video frames. Similar to the HOG descriptor, the histogram representation is created by accumulating the magnitudes of the flows into the direction bins.



FIGURE 2.8: An example optical flow.

The optical flow is highly affected by the camera motion since the camera motion results a difference between consecutive frames due to the changes in the perspective. The Figure 2.9 shows example optical flows, when there is camera motion. We can see that almost of the edges in the frames results in high optical flow values with the camera motion.



FIGURE 2.9: An example optical flow when there is a camera motion.

2.3.3 Motion Boundary Histogram (MBH)

The MBH descriptors [12] are used to represent the feature points, which are the derivatives of the optical flow and similar to HOF and it is computed over a $(3 \times 2 \times 2)$ grid with 8 bins. The MBH descriptors are shown to be effective descriptors for action recognition [6, 20]. The difference between HOF and MBH is that the MBH descriptors have horizontal and vertical optical flow components separately, but the HOF combines them in a quantization

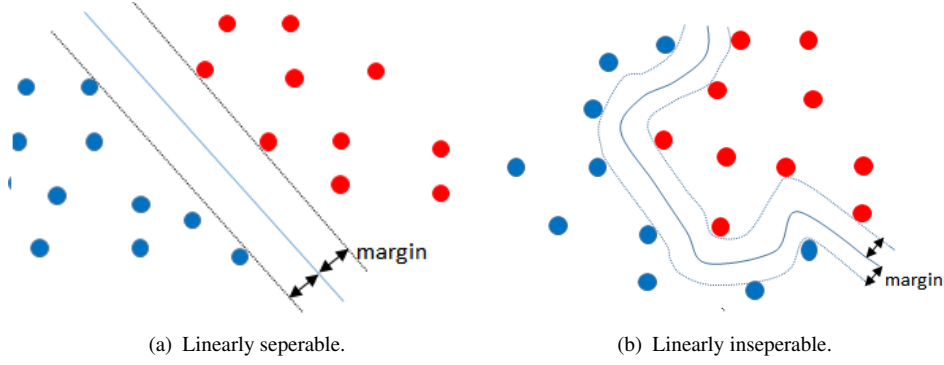


FIGURE 2.10: Support Vector Machine - Classification of Non-Linear Problems.

step. Since the HOF descriptors are highly affected by the camera motion, MBH descriptors can be used since it has some level of robustness to camera motion, as the positive effect of using derivatives of the optical flow.

Similar to the HOG&HOF descriptor, the histogram representation is created by accumulating the magnitudes of the derivatives of the flows into the direction bins. The dimension of the MBH feature (96x2) is likely to be greater than the dimension of the HOG&HOF features.

2.4 Support Vector Machines (SVM)

SVM is a classification technique that is based on max-margin hyperplanes. SVMs find a hyperplane that separates the positive and negative samples with a maximum margin. The margin is simply the total distance of example data points to the hyperplane. It is often the case that the positive and negative examples are not linearly separable with a hyperplane. To allow separation of non-linear examples, kernel trick [51] is used to map the input data to a high dimensional space and separate with non-linear functions. Figure 2.10 shows a simple example.

2.5 Deep Learning

With the advancement of parallel processing power (GPUs, CPU clusters), the convolutional neural networks (CNNs) started to be successfully employed for the computer vision problems [35, 43–46]. The convolutional neural networks are similar to the neural system's structure where the neurons are hierarchically organized so that each neuron transforms the

output of previous neuron with transformation function. The convolutional neural networks consist of multiple type of layers such as convolutional layer, pooling layer, normalization layer and fully-connected layer.

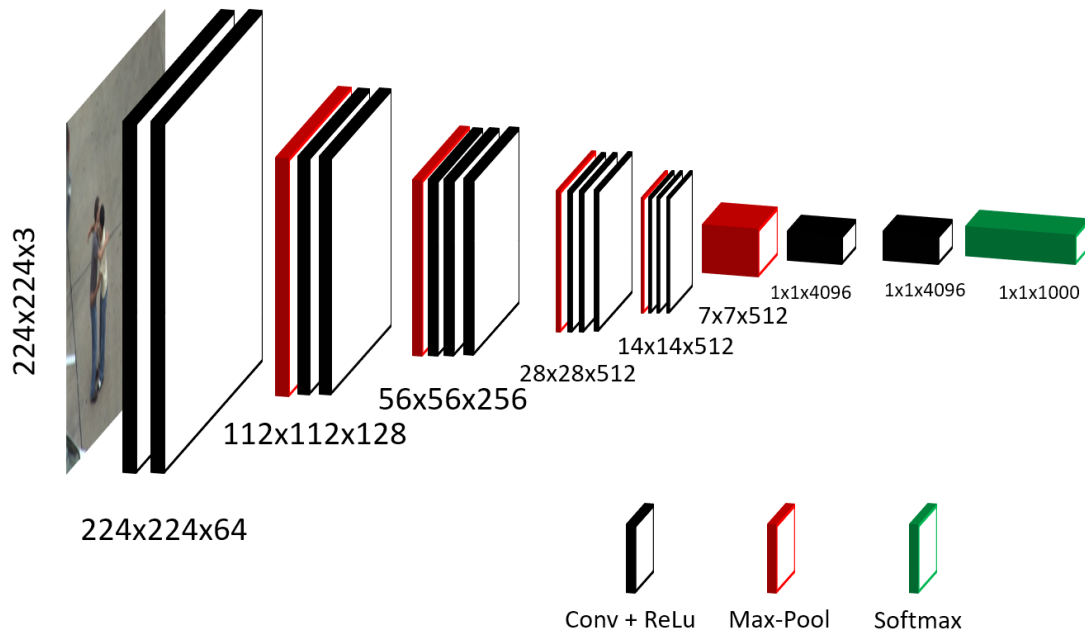


FIGURE 2.11: VGG-16 network, input to output .

The convolutional layers are the core of the convolutional neural networks. It has a learnable filter. The size of the filter changes from very small patches (e.g. 5x5x3) to the almost full size of the input data. During the forward pass, the convolutional filters are applied (dot product) to input data with a sliding window method so that a response map is produced.

One of the most used deep network is the VGG [52] network. The Figure 2.11 shows the architecture of the VGG-16 network. For the VGG-16 network, the input is an RGB image at a resolution of 224x224. Before the input is provided to the network, mean subtraction is suggested [35] to normalize the intensity values. The mean value is single value calculated from the training dataset. After the input layer, convolution filters are applied. The size of the convolution layers are all 3x3 and the stride is 1. The convolutional layers at the beginning of the network tends to find-out the small edge/corner fields whereas the convolutional layers to the end of the network laerns higher level information.

The pooling layers are commonly inserted between the successive convolutional layers. The pooling layers reduce the size of the data that need to be processed by the convolutional layers. The simplest pooling functions are max- pooling, average-pooling. The pooling layer

also reduces over-fitting. Some of the convolution layers are followed by the pooling layers as shown in Figure 2.11. The pooling is applied at 2x2 pixel window and with a stride of 2 pixels to reduce the total amount of data.

The normalization layers are used for bounding activations of the convolutional neural networks. Sometimes, the convolution operation results in unbounded response, in such cases, the normalization layers regularize the responses in a local neighborhood by damping the responses. All of the convolutional layers are followed with ReLU activation function in the VGG-16 network. The Relu activation function is defined as $f(x) = \max(0, x)$. Simply the output is 0 when the input x is negative.

The fully connected layer is also a specific type of convolutional layer. The convolutional layers operates over a local region in the input data, whereas the fully connected layers are connected to all outputs in the preceding layer. In the VGG-16 network, the convolutional layers are followed by the two Fully-Connected (FC) layers and the last layer is the soft-max layer. The soft-max layer is used to produce class probabilities.

The filters in the convolutional layers are learned with gradient based learning methods [53]. The loss function is minimized by analyzing the impact of input on the variation of the loss. The gradient is a measurement of the parameter's impact to the loss. The gradients are calculated using back propagation [54] from output layers to the input layers.

When the trained network is obtained, the testing is much faster than the training since there is no back propagation required. Once the learned filters are applied the classwise score is obtained from the soft-max layer. The size of the soft-max layer should be same as the number of the classes in the dataset.

The Figure 2.12 shows the VGG-16 deep network architecture with layer names.

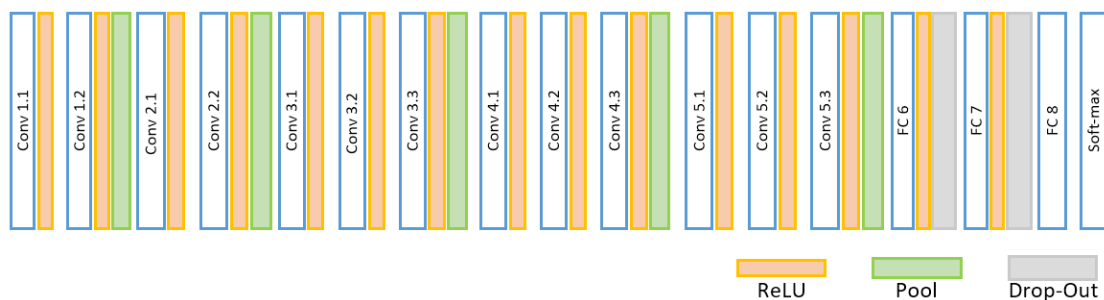


FIGURE 2.12: A representation of VGG-16 deep network.

2.6 Sequence Mining

2.7 Problem Definition

With the advance of computer vision methods and technologies, the variety and complexity of the computer vision problems are getting bigger and bigger. The discovered frequent patterns from datasets with the data mining methods, are being used in computer vision area for classification, detection and analysis of different objects, actions in the videos or still images.

The topic of data mining is basically working on associative rule mining from a database of unordered transaction records labeled with a time tag or a unique number. Data mining algorithms answer “Which two elements often coexist?” question. Sequence mining is a subdivision of the field of data mining and tries to derive more general rules taking into account the temporal relationship between transactions.

The use sequence mining is promising for the recognition of actions from videos, since the videos are set of ordered frames and the sequence mining methods can be used for problems that consists of temporally ordered set of sub-components.

The sequential pattern mining paradigm was first introduced by Agrawal and Srikant [55] and it is a subject of data mining. The sequential pattern mining methods find frequently occurring patterns where the values in the transactions are in a sequential form and have associated temporal labels. The problem is to find the most co-occurring closed-sequences when the dataset contains time ordered set of features. Sequential pattern mining has been successfully used in many applications that involve temporal data points.

One of the first method proposed for the sequence mining problem is the AprioriAll algorithm [55], which is developed based on the Apriori [56] algorithm, one of the basic data mining methods. “All subclusters of frequent patterns also frequently occur in the database”. When the think about the inverse of this idea, “all upper clusters of a non-frequent pattern are also not frequent”, which means that there is no need to use them as candidate patterns. For example, $\langle ade \rangle$ is a subset of $\langle acde \rangle$, and it is clear that $\langle acde \rangle$ can not be a frequent sequential pattern in a database where $\langle ade \rangle$ is not a frequent pattern in that database. With this information obtained, the Apriori algorithm is used to find the patterns by generating longer candidate patterns by using shorter patterns and frequent items.

Candidate trimming is one of the most vital process for reducing the search area. While searching for patterns, first, pattern candidates (starting from single item) are created, then frequent candidates are selected from these candidates, pattern candidates of length 2 are created, and the pattern search process is continued. It is a low performance algorithm because the number of candidate patterns increases rapidly in each step and the database needs to be re-scanned frequently.

Formally, the sequential pattern mining is defined as follows [55] : Let $I = \{i_1, i_2, \dots, i_n\}$ be the universal set of items, where i_x is an item. X is defined as an itemset, if $X \subseteq I$. A sequence $s_m = \langle X_1, X_2, \dots, X_m \rangle$ is defined as an ordered list of itemsets. A sequence $s_1 = \langle A_1, A_2, \dots, A_n \rangle$ is said to be contained in another sequence, $s_2 = \langle B_1, B_2, \dots, B_m \rangle$, if there exists integers $1 \leq i_1 < i_2 < \dots < i_n \leq m$ such that $A_1 \subseteq B_{i_1}, A_2 \subseteq B_{i_2}, \dots, A_n \subseteq B_{i_n}$, where A and B are itemsets.

Given a sequence database $SDB = \{s_1, s_2, \dots, s_k\}$, which is a set of sequences, the problem of sequential pattern mining is to find the sequential patterns that have support values greater or equal to a user specified minimum *support*. The *support* for sequential pattern candidate $support_{SDB}(s)$ is the fraction of total sequences that include that particular sequential pattern candidate such that:

$$support_{SDB}(s) = \frac{|T \mid T \in SDB, s \subseteq T|}{k} \in [0, 1] \quad (1)$$

The original problem of sequential pattern mining considers only the order of item occurrences, but not the intervals between two items. The time interval between two items has an effect on the meaning of the sequence and should be taken into account. Hirate and Yamana [37] proposed a method called Generalized Sequential Pattern Mining with Time Intervals (GSPM) that employs time labels for each itemset and brings additional constraints to address time interval requirements in sequence mining.

The sequences that contain time labeled itemsets are called as interval extended sequence (*is*) and the sequential database that is a set of interval extended sequences are called as interval extended sequence database (*ISDB*). Four types of additional constraints are proposed in *support* calculation. The constraint C1 is the minimum time interval required between two adjacent itemsets. The constraint C2 is the maximum time interval required between two adjacent itemsets. The constraint C3 is the minimum time interval required between the head

and tail of a sequence. The constraint C4 is the maximum time interval required between the head and tail of a sequence. For example, consider the interval extended sequence $is = \langle (0; 1) (1; 1, 2) (3; 1, 2, 3) \rangle$. The time interval between the head and tail of the sequence is is 3 and the time interval between the second and the third itemset is 2.

To find the frequent interval extended sequences, the algorithm GSPM [37] makes recursive projections of the interval extended sequence database with prefixes. This method is called as Prefix-Span [57]. In the Prefix-Span method, for each pattern candidate, a subset of the global database is created, which is called as pseudo-projection (pseudo-projection). Computations were performed on this smaller database. In this way, both less data is processed and unnecessary pattern candidates that could be created in the future were avoided. Section 3.5 describes the difference of the most known sequence mining algorithms. The shortest prefix is a single item i with a time label 1.

Formally, let's assume $\alpha = \langle (t_{1,1}, A_1), (t_{1,2}, A_2), \dots, (t_{1,m}, A_m) \rangle$ and $\beta = \langle (t'_{1,1}, B_1), (t'_{1,2}, B_2), \dots, (t'_{1,m}, B_m) \rangle$ be interval extended sequences where $t_{x,y}$ is the difference of the occurrence time of the A_x and A_y . The α is said to be contained in β if there exists integers such that $1 \leq i_1 < i_2 < \dots < i_n \leq m$ such that $A_1 \subseteq B_{i_1}, A_2 \subseteq B_{i_2}, \dots, A_n \subseteq B_{i_n}$ and $t_{1,i} = t'_{1,i_1}, t_{1,2} = t'_{1,i_2}, \dots, t_{1,n} = t'_{1,i_n}$. The prefix of interval extended sequence α based on the (t_β, A_β) is defined as following:

$$prefix(\alpha, A_\beta, t_{1,\beta}) = \langle (t_{1,1}, A_1), (t_{1,2}, A_2), \dots, (t_{1,j}, A_j) \rangle \quad (2)$$

Postfix of interval extended sequence α with regard to (t_β, A_β) is defined as following:

$$postfix(\alpha, A_\beta, t_{1,\beta}) = \langle (t_{j,j}, A'_j), (t_{j,j+1}, A_{j+2}), \dots, (t_{j,m}, A_m) \rangle \quad (3)$$

where A'_j contains the items in A_j and doesn't include the items in A_β . When there exists no integer j , postfix of α with regard to (t_β, A_β) becomes the following:

$$\begin{aligned}
prefix(\alpha, A_\beta, t_{1,\beta}) &= \emptyset \\
postfix(\alpha, A_\beta, t_{1,\beta}) &= \emptyset
\end{aligned} \tag{4}$$

The patterns are calculated by finding the frequent items first. For each frequent item a , where $support_{ISDB}(1, a) > min_{support}$, the initial interval extended sequence α is created as $\langle (0, a) \rangle$ where $a \in A_i$. Then, $ISDB|\alpha$ is created. The $ISDB|\alpha$ is the projection of the $ISDB$ with regard to α and defined as:

$$ISDB|\alpha = \{is | is \neq \emptyset \wedge is = postfix(\gamma, \alpha, 0)\} \tag{5}$$

where $\gamma \in ISDB$.

Similarly, in the next level of projection, all possible pairs $(t_{1,i}, a)$ in $ISDB|\alpha$ are generated. For each pair satisfying the minimum support constraint, β is defined as $prefix(\alpha, a, t_{1,i})$ and the projected interval extended sequence database $ISDB|\beta$ becomes a collection of postfixes of sequences in $ISDB$ with regard to β .

$$\begin{aligned}
ISDB|\beta &= \{is | is \neq \emptyset \wedge is = postfix(\gamma, a, t_{1,i}) \\
&\quad \wedge support_{ISDB|\alpha}(t_{1,i}, \beta) \geq min_{support} \\
&\quad \wedge \beta \text{ satisfies } C_1, C_2, C_3 \text{ and } C_4\}
\end{aligned} \tag{6}$$

where $\gamma \in ISDB$. The pseudocode of this algorithm is given in Algorithm 1. For more details, please refer to [37].

When we execute the sequential pattern mining algorithm GSPM [37], we observe that there are many redundant sequential patterns that are included in other sequential patterns that have same support value. As an example, the sequential patterns: $sp_1 = \langle (0; 1, 2, 3) \rangle$ and $sp_2 = \langle (0; 1, 3) \rangle$ have support value of 75 and sp_2 is contained in sp_1 , so there is no additional information gained by use of sp_2 . Elimination of these sequences is done by use of *Closed Sequence Mining* methods. A closed sequential pattern is a frequent sequential pattern that is not included in another sequential pattern having exactly the same support. In our case,

Algorithm 1 Sequential Pattern Mining Algorithm [37].

Input: $ISDB \neq \emptyset, min_{support} \geq 0, C1 \geq 0, C2 \geq 0, C3 \geq 0, C4 \geq 0$

Output: R {sequential patterns those satisfies constraints}

$R \leftarrow \emptyset$

for $i \in ISDB$ **do** {where i is an item}

if $support_{ISDB}((0, i)) \geq min_{support}$ **then**

$R = \{R, \langle(0, i)\rangle\}$

$R = projectWithPrefix (ISDB \setminus \langle(0, i)\rangle, R, min_sup, C1, C2, C3, C4)$
 (Algorithm 2)

end if

end for

return R

Algorithm 2 *projectWithPrefix*

Input: $ISDB|prefix \neq \emptyset, R, min_{support} \geq 0, C1 \geq 0, C2 \geq 0, C3 \geq 0, C4 \geq 0$

Output: R

for $(t, i) \in ISDB|prefix$ **do**

if $support_{ISDB}((t, i)) \geq min_{support}$ **then**

if (t, i) satisfies $C1$ and $C2$ **then**

$newPrefix := concatenate(prefix, (t, i))$

if $newPrefix$ satisfies $C3$ and $C4$ **then**

$R = \{R, newPrefix\}$

if size of $ISDB|newPrefix \geq min_{support}$ **then**

$projectWithPrefix (ISDB|newPrefix, R, min_sup, C1, C2, C3, C4)$

end if

end if

end if

end if

end for

we adopt the sequence mining algorithm of Hirate and Yamana [37] with the BI-Directional Extension [58] (BIDE). BIDE checks if a sequential pattern is closed or not, with no need to maintain the set of historical closed patterns.

To give more insight on the problem, an example set of sequences is shown in Table 2.1. In this example, there are four input sequences with three, four, two and two itemsets respectively. Each itemset consists of one or more elements and labeled with a tag to account for the time of occurrence. Table 2.2 shows some sequential patterns extracted by applying sequence mining to the set of example sequences provided in Table 2.1.

TABLE 2.1: Example input sequences for the sequence mining algorithm.

ID	Sequences
Seq1	$\langle(0; 1) (1; 1, 2, 3) (2; 1, 3)\rangle$
Seq2	$\langle(0; 1) (1; 1, 2) (2; 1, 2, 3) (3; 1, 2, 3)\rangle$
Seq3	$\langle(0; 1\ 2) (1; 1, 2)\rangle$
Seq4	$\langle(0; 2) (1; 1, 2, 3)\rangle$

TABLE 2.2: Extracted sequential patterns from example input of Table 2.1.

ID	Sequential Patterns	Support
P2	$\langle(0; 1, 2)\rangle$	100%
P5	$\langle(0; 2) (1; 1)\rangle$	100%
P1	$\langle(0; 1, 2, 3)\rangle$	75%
P6	$\langle(0; 1) (1; 1, 2)\rangle$	75%
P3	$\langle(0; 2) (1; 1, 2)\rangle$	75%
P4	$\langle(0; 2) (1; 1, 3)\rangle$	75%
P7	$\langle(0; 1, 2) (1; 1)\rangle$	75%

2.7.1 Example Execution of Sequence Mining Algorithm

Here we provide a running example of the Algorithm 1 for the input sequences defined in Table 2.1 with the following parameters: $min_{support} = \%80$, $C1 = 0$, $C2 = 2$, $C3 = 0$ and $C4 = 9999(infinity)$. Execution steps can be viewed in Figure 2.13 and they are given as a list below:

1. Scan $ISDB$ to find the frequent items 1, 2 and create initial sequences as: $\langle(0; 1)\rangle$ and $\langle(0; 2)\rangle$. These two sequences are added to R .
2. Select the most frequent item 1, and create projected database $ISDB|\langle(0; 1)\rangle >$. (Step 1)
3. Scan $ISDB|\langle(0; 1)\rangle$ to find pairs that satisfy the $min_{support}$, $C1$ and $C2$. It is only $\langle(0; 2)\rangle$.
4. Append $\langle(0; 1, 2)\rangle$ to R since they respect rules $C3$ and $C4$.
5. Generate $ISDB|\langle(0; 1, 2)\rangle$ to find pairs that satisfy the $min_{support}$, $C1$ and $C2$ (Step 2). There are no frequent pairs in $ISDB|\langle(0; 1, 2)\rangle$. Terminate mining frequent sequences that contain $\langle(0; 1, 2)\rangle$ as a prefix.

6. Terminate mining frequent sequences that contain $\langle(0; 1)\rangle$ as prefix.
7. Select the second most frequent item 2 and create projected database $ISDB|\langle(0; 2)\rangle$. (Step 3)
8. Scan $ISDB|\langle(0; 2)\rangle$ to find pairs that satisfy the $min_{support}$, $C1$ and $C2$. It is only $\langle(1; 1)\rangle$.
9. Append $\langle(0; 2)(1; 1)\rangle$ to R since they respect rules $C3$ and $C4$.
10. Generate $ISDB|\langle(0; 2)(1; 1)\rangle$ to find pairs that satisfy the $min_{support}$, $C1$ and $C2$ (Step 4). There are no frequent pairs in $ISDB|\langle(0; 2)(1; 1)\rangle$. Terminate mining frequent sequences that contain $\langle(0; 2)(1; 1)\rangle$ as a prefix.
11. Terminate mining frequent sequences that contain $\langle(0; 2)\rangle$ as prefix.
12. Select the third most frequent item 3, and create projected database $ISDB|\langle(0; 3)\rangle$. (Step 5)
13. Scan $ISDB|\langle(0; 3)\rangle$ to find pairs that satisfy the $min_{support}$, $C1$ and $C2$. However, there is no pair satisfying min_sup criteria.
14. Output R and terminate.

2.8 Histogram Computation with Sequential Patterns

The histograms are constructed using the occurrence of the sequential patterns within the video sequences. The occurrence of a pattern in a sequence is defined in Section 2.6. The pseudo-code of the algorithm for computing these occurrences is given in Algorithm 3.

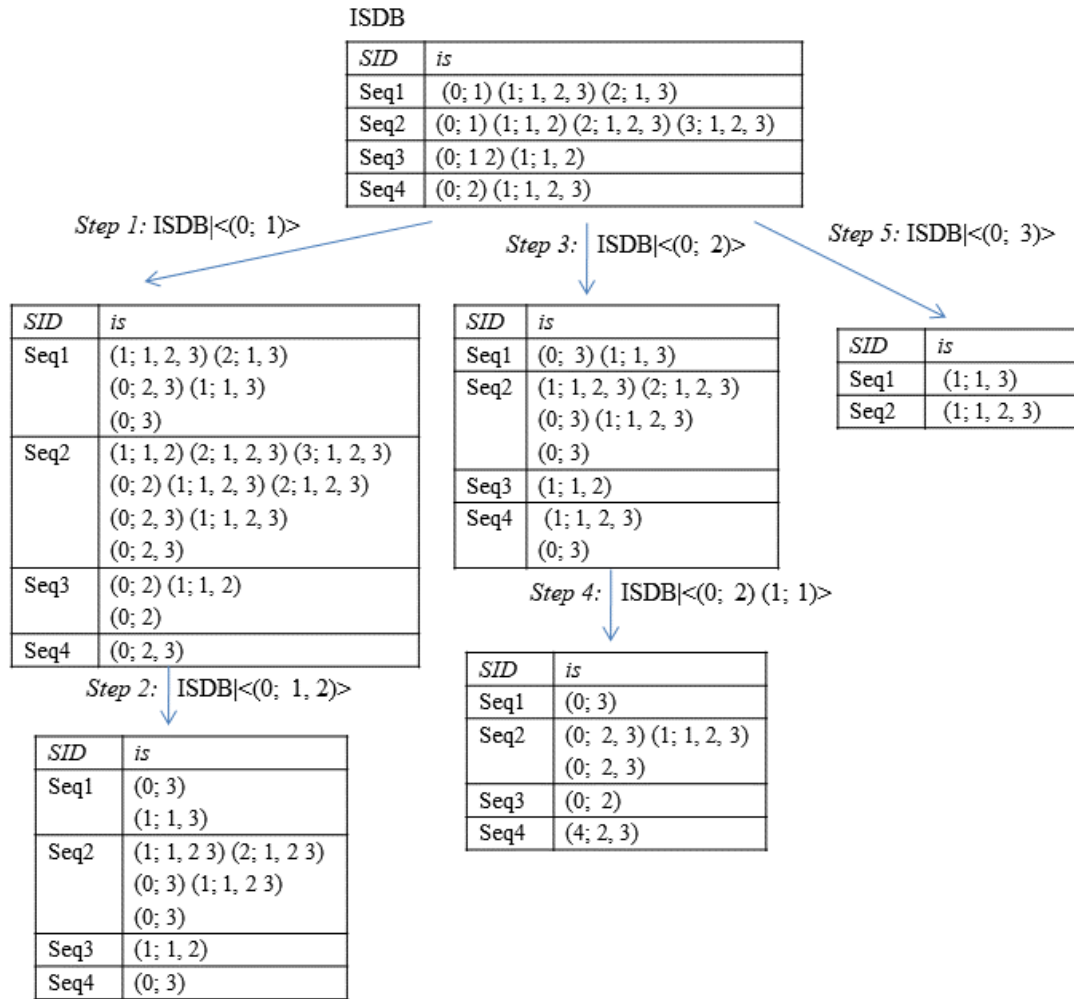


FIGURE 2.13: Sample mining steps for input database Table 2.1.

Algorithm 3 *countPatternOccurrence*

Input: $seq \neq \emptyset, pttrn \neq \emptyset, pttrn_idx \geq 0, seq_idx \geq 0, num_occur \geq 0$

Output: $num_occur \geq 0$ {number of occurrences}

```
1: for  $i$  in  $seq[seq\_idx] : seq[last\_idx]$  do {loop on itemsets}
2:   if  $t(i) = t(pttrn[pttrn\_idx])$  then
3:      $pttrn\_item \leftarrow firstItem(pttrn[pttrn\_idx])$ 
4:     for  $j$  in  $i$  do {loop on items in itemset  $i$ }
5:       if  $pttrn\_item = j$  then
6:         if  $pttrn\_item = lastItem(pttrn)$  then
7:            $num\_occur \leftarrow num\_occur + 1$ 
8:         else if  $pttrn\_item = lastItem(pttrn[pttrn\_idx])$  then
9:            $num\_occur \leftarrow num\_occur + countPatternOccurrence(seq, pttrn,$ 
             $pttrn\_idx + 1, seq\_idx + 1, num\_occur).$ 
10:        else
11:           $pttrn\_item \leftarrow nextItem(pttrn[pttrn\_idx])$ 
12:        end if
13:      end if
14:    end for
15:  end if
16: end for
17: return  $num\_occur$ 
```

3 RELATED WORK

There are many works in the literature of computer vision that aims at recognizing actions. The first line of work is based on the local spatio-temporal features leveraged with a discriminative classifier. Second line of work uses temporal dynamic models, such as Hidden Markov Model, Bayesian Model and Finite State Models. Third line of work uses the key components that extracted from the training samples. They analyze the temporal ordering of these key components. In the recent years, deep learning methods are proposed with the availability of the high processing power of the GPUs. The following sections provide detailed explanation of the methods.

3.1 Local Interest Point Methods

3.1.1 Bag of Words and Interest Points

The bag-of-words (BOW) with STIP is employed by a number of approaches to represent the human action. The bag-of-words representation is mostly combined with discriminative classifiers [7, 9, 10]. The Support Vector Machine [59] (SVM) is one of the most used classifier that used with the histogram representation.

Marín-Jiménez et al. [11] reported the performance of the bag of space-time interest points on human interaction datasets. According to their results, the combination of bag of words and STIP was still a competitive approach in human interaction recognition problems and the improvement of the STIP representation an important issue for the future work. They also found out that when modeling interactions between persons, the context does not help in the recognition process and, therefore, it is needed to reject STIP densely sampled outside the person region. Despite the computational efficiency of the bag of STIPs method, the bag of words model discards spatial and temporal structure.

3.1.2 Co-Occurrence of Interest Points

Zhang et al. [13] proposed a more successful method, which captures co-occurrence information. Spatial and temporal distance between two local features are used as the core of the information. The videos are expressed as histograms of spatio-temporal phrases. First, they

extracted the local interest points from the videos and found the visual word with the bag-of-words representation. Then, between two training videos, they encode spatial and temporal distance of the co-occurring visual words by marking the distances in a three-dimensional offset plane of Δx - Δy - Δt . Each point in the resulting three-dimensional space forms a space-time expression, and the value of that point specify the length of the space-time expression. An exemplary space-time expression appears in Figure 3.1¹. The circles in the figure represent the local interest points and the word assignments are represented with colors.

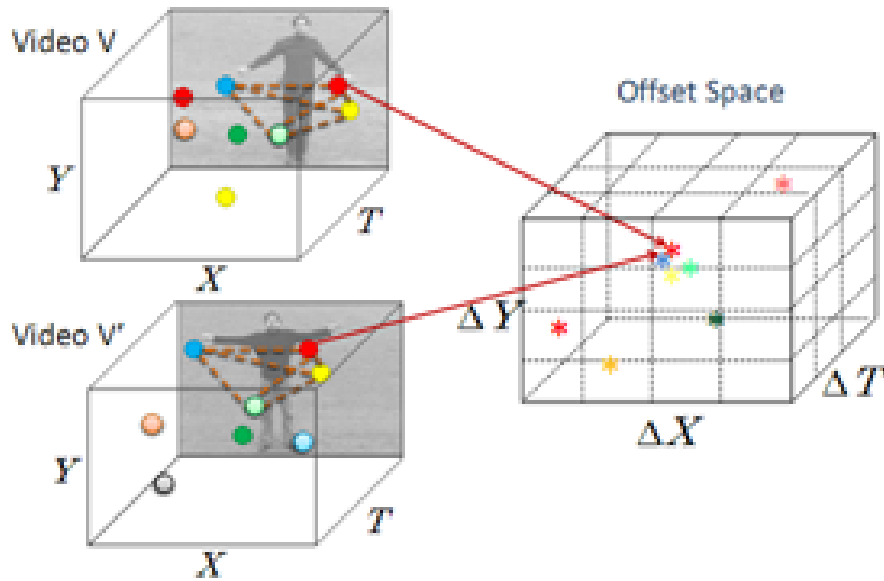


FIGURE 3.1: Spatio-temporal phrase representation [13].

In order to adapt the method against the speed and scale differences of the movement, they created a five dimensional distance universe by adding different spatial and temporal scale values to the three dimensional offset distance space. They stated that this design achieved adaptation at a certain level against the execution speed of the action and the changes in the scale. The obtained distance space was used for classification with SVM. In the worst case, the method requires $O(n^2)$ computation time where n is the number of local features in the video. However, the algorithm complexity is linearly related to the number of local features in practice. However, it can be seen that the performance of Zhang et al. [13]’s method in a real-life video may not be that successful because the space-time interest points are highly affected by the camera motion. When there is a camera movement, all of the edge points in

¹The Figure 3.1 is reprinted by permission from Springer Nature: Springer eBook [13], © Springer-Verlag Berlin Heidelberg (2012).

a frame may be accepted as a local space-time interest point, even if there is no real motion such as the background objects.

Slimani et al. [15] proposed a method that extracts a descriptor that captures the co-occurrence of the local interest points found in 3D XYT spatio-temporal volume for each interacting person. Then the co-occurrence descriptor is used in standard discriminative classifiers.

Savarese et al. [14] used the co-occurrences within local spatio-temporal regions to encode the local motion and appearance and also they used the probabilistic latent semantic analysis (pLSA) to learn the action classes. Similar to the Niebles et al.'s [2] method, they work on finding best 3D spatio-temporal volumes that contain most discriminative local interest points. Figure 3.2¹ shows the 3D slices of the proposed model. Classes were learned by the pLSA method on the histogram matrix created for every 3D slice and were used for the testing. This method increased performance by about %10 in the KTH dataset [19] when used with histograms of basic visual words.

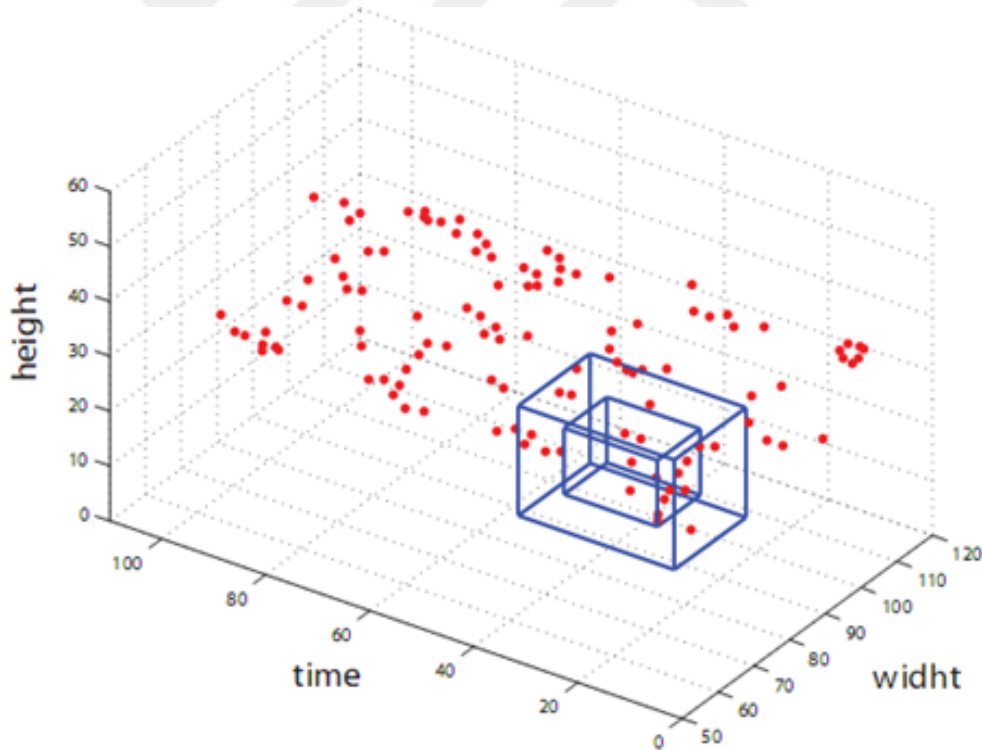


FIGURE 3.2: A representation of 3-D volumes used for modeling co-occurrence information [14].

¹The Figure 3.2 is reprinted, with permission, from [14] ©2008 IEEE.

The local features, especially STIP, are highly affected by the camera movements since they have no adaptation for external movements. To deal with this problem, Liu et al. [60] proposed a set of filters that removes the detected features according to the statistical behaviors of them.

Khokher et al. [16] proposed an improved spatio-temporal interest point detector by extracting salient interest points by taking into account long-term temporal interactions and camera motion. They encode the local features in the form of tensors in order to retain the spatio-temporal structure. To reduce the size of the tensor, they selected the features by Fisher ranking.

3.1.3 Graph of Interest Points

Gaur et al. [17] represented the videos as a string of feature graphs of local interest points scattered in 3D space. They have also transformed the problem of identifying the occurrence time of the action into a graph matching problem and modelled accordingly. Locating local points of interest from the videos, the local feature points are considered as vertices, and the distances between the local feature points as edges. An example illustration is shown in Figure 3.3. In the case of matching the graphics sequences of two videos, the similarities of the scattered graphs in different frames were first calculated, and then the similarity of the temporal relationships between these graphs was examined. Flexibility is provided by using the Dynamic Time Warping (DTW) method, since it is thought that exact matching of individual graphs in the time domain will damage the generalization of the model. With DTW, even though the times of occurrence of the two graphs differ, graphs may be accepted as “matched” since the actions may be performed at different speeds. Gaur et al. [17] thought that the two graphs are “matched” as long as the temporal order of the local features do not change.

Figure 3.4 shows an example of how Dynamic Time Warping is used. With this model, it is possible to find out where the desired movements are recognized in the video (Subsequence DTW method [61]) without using training videos. Even if the proposed method of Gaur et al. [17] models the interrelationships between local feature graphs, it ignores interrelationships between local attributes that can be created by multiple actors and are not in the same temporal position.

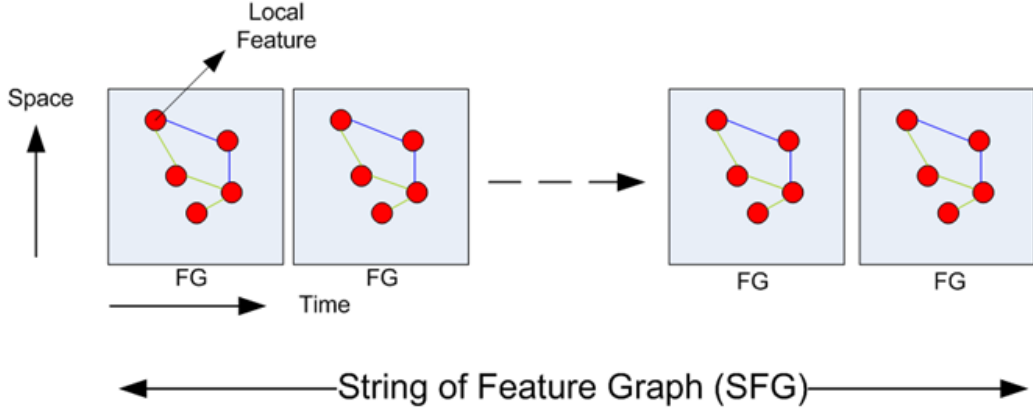


FIGURE 3.3: The representation of the “String of Feature Graph” model.

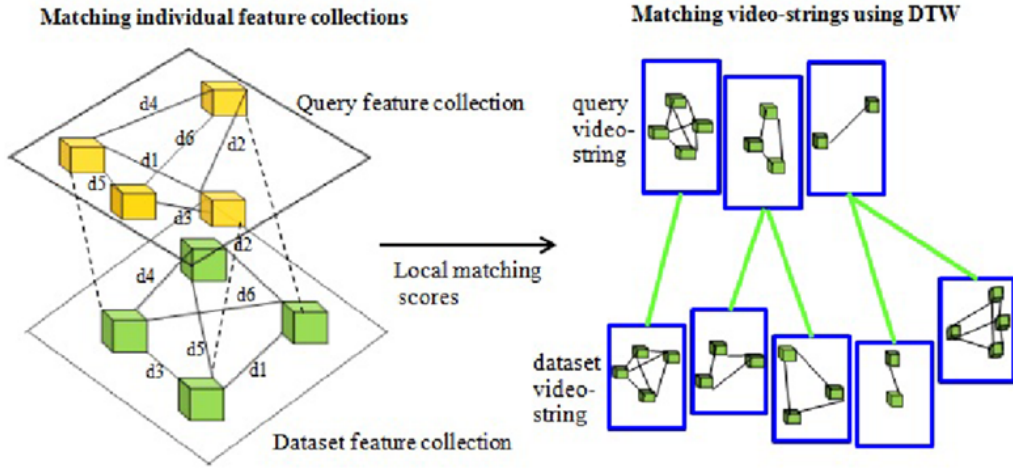


FIGURE 3.4: Video matching with dynamic time warping.

3.1.4 Spatial and Temporal Relationship of Interest Points

Ryoo and Aggarwal [18] introduced a new method to compare the structure of videos by using the spatio-temporal relationships between the features with the temporal and spatial predicates (before, meets, overlaps, far, near and more). Their approach detects and localizes all occurring activities.

Instead of directly classifying the videos directly, the proposed method finds out where the structural similarity starts and ends. For each type of feature, Ryoo and Aggarwal have created two 3D histograms, one of them spatial and the other one id temporal. In these histograms, there is information about the existence of a relationship between feature types and how they relate to each other. The generated histograms measure the similarity between the

two videos that will observe both spatial and temporal relationships with a kernel similar to the histogram intersection kernel structure. The Figure 3.5¹ shows the 3D histograms and matching scenario. This finding of similarity works with $O(n^2)$ time complexity because it compares the inter-type relationships of different attributes, where the number of "n" specifies the common feature types that common between the two videos. The authors stated that, the method may also use the result of any tracking algorithm.

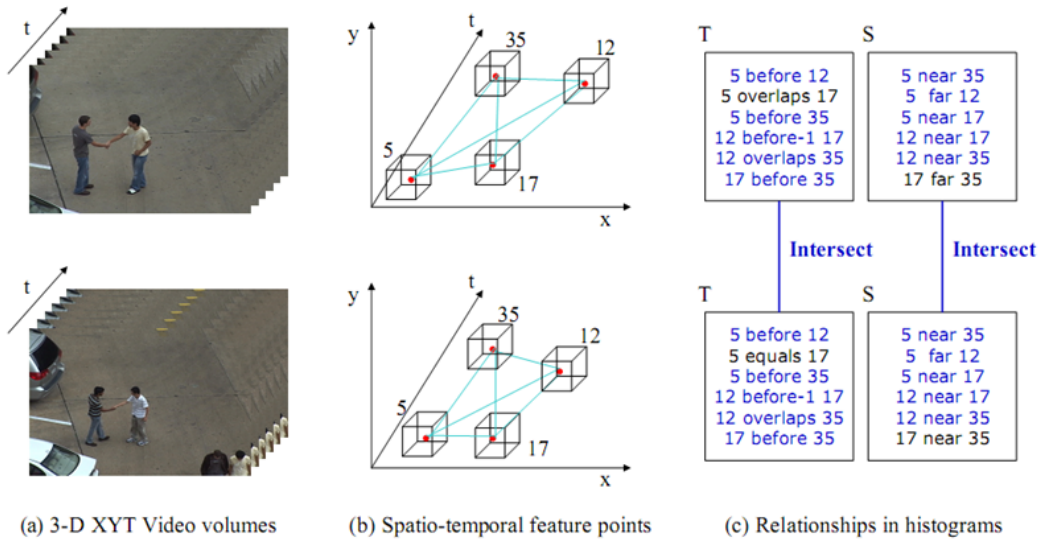


FIGURE 3.5: A representation of Ryoo and Aggarwal's model [18].

3.2 Models Focused on Temporal Dynamics

Niebles et al. [2] improved the temporal pyramid matching idea and propose a dynamic temporal scale selection method rather than using fixed length temporal parts. To select the best temporal representation, Niebles et al. [2] generate a set of random temporal scales and train classifiers for each temporal partitions, where the best temporal scales are selected according to the recognition performance. As opposed to finding best partitions, our work operates on finding the most frequently occurring feature sequences at different temporal scales. The Figure 3.6 shows the suggested model. For each motion segment obtained with this model, a flexibility is provided by a Gaussian distribution in time dimension. In addition to Niebles et al.'s [2] method, Savarese et al.'s [14] method uses both temporal and spatial segments instead of using only temporal segments.

¹The Figure 3.5 is reprinted, with permission, from [18] ©2009 IEEE.

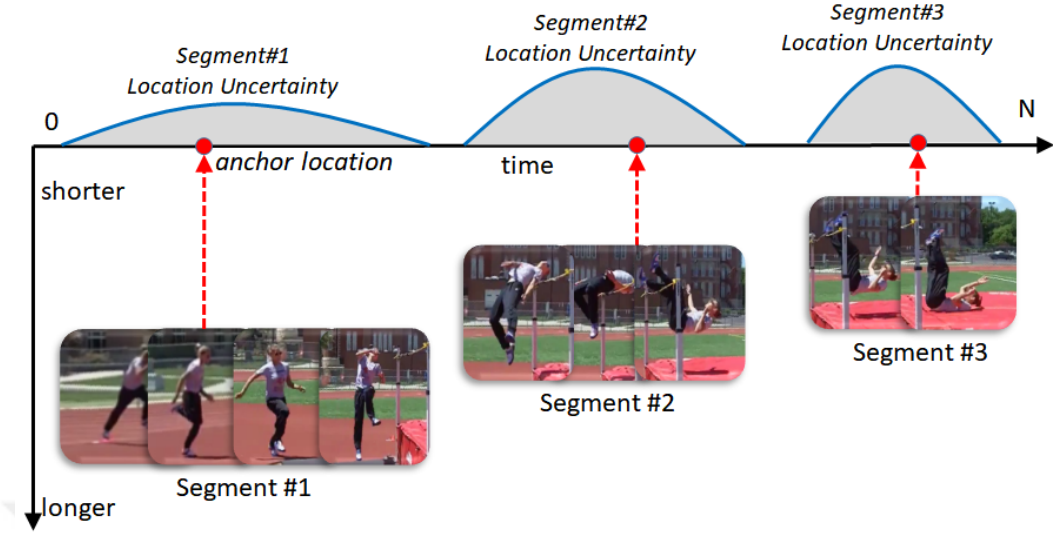


FIGURE 3.6: A representation of dynamic temporal scale selection method.

LiMin et al. [5] suggest a method called as *motion phrases* that can fill in the gap to recognize complex movements with low level information. In this method [5], each training video is represent by the information exacted from equal length segments. Video segments are selected according the with dense motion trajectories [20] and the segments are expressed in HOG / HOF and MBH descriptors. Then, these segments are clustered by the clustering algorithm and an SVM classifiers are trained on the cluster results. The SVM classifiers were run on the training set to determine the clusters with the highest performance. To increase the generalization capability of the model, additional cluster centers representing the entire training set were included for the samples with low SVM score, so that each training video could be adequately represented. These cluster centers are called motion atoms.

Since the motion atoms obtained exists in temporally different locations, they suggested to combine the motion atoms in a way that recognizes complex movements with better capability to capture higher-level information. By using AND and OR structures, motion atoms in the similar locations and same clusters are combined with the OR operator, and motion atoms in different clusters in different places are combined with the AND operator to create movement expressions.

The example motion representation is shown in Figure 3.7. When the candidate movements obtained, the discriminative motion expressions are mined with Apriori [62] algorithm. In the Apriori algorithm, the most frequent association rules in the entire data set are found, but the found patterns don't have to be discriminative, so the authors modified the Apriori algorithm

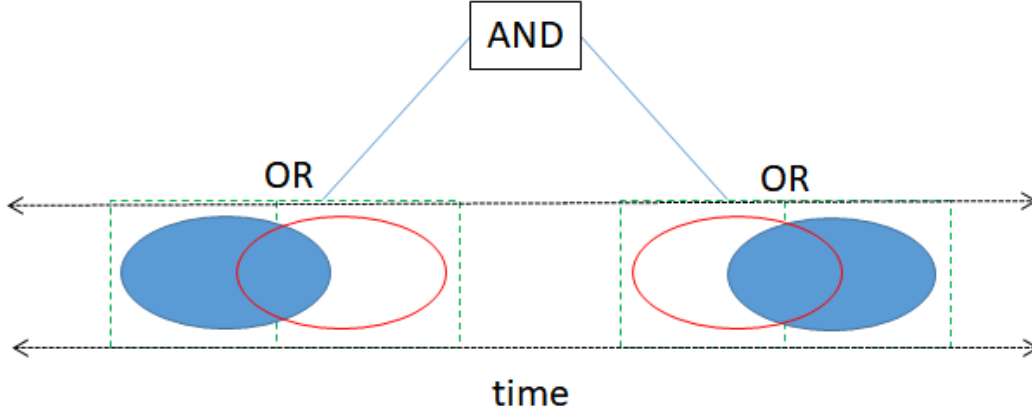


FIGURE 3.7: A representation of motion phrases.

to select distinctive motion expressions for each class. The obtained motion patterns, motion atoms and dense motion trajectories are combined into a single histogram representation for classification.

Wang et al. [20] proposed the use of dense trajectories to model the motion information. The dense trajectories also have some level of robustness to camera motion since the trajectories are based on optical flow of points. The MBH descriptors are used along the trajectories to model the actions. The Figure 3.8¹ shows the process of Wang et al.'s [20] feature extraction method. Zhang et al. [63] follows the similar method with the Wang et al. [20] so that they cluster trajectories using the coherent filtering and employ a multiple instance learning (C-KNN). Wang and Schmid [6] have also provided an upgrade to Wang et al. [20] by estimating the camera motion and filtering out the background trajectories.

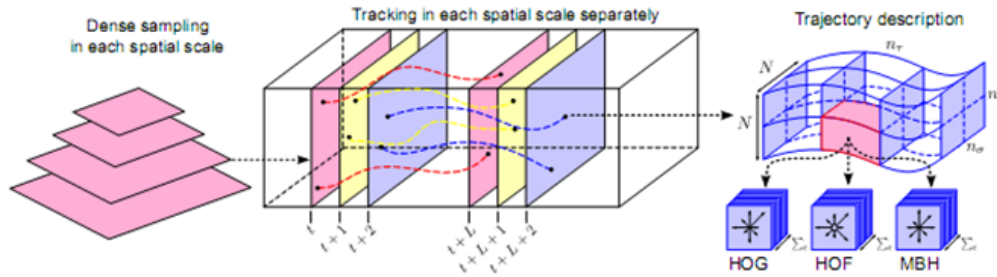


FIGURE 3.8: A representation of dense trajectories [20].

Since MBH attributes are the derivative of optical flow information, only the locations of motion is highly responsive, thus the extracted attributes are more successful in representing

¹The Figure 3.8 is reprinted, with permission, from [20] ©2011 IEEE.



FIGURE 3.9: A representation of person-centered descriptor [21].

the motion. However, in some cases, appearance information is more important than motion information in recognizing human movements, so Wang et al. [20] included the HOG / HOF information into the last attribute set in addition to the MBH for better recognition. The method is tested on four different sets of data and it is seen that performance is much better than the standard bag-of-video-words method.

Patron et al. [21] build a person-centered descriptor and use an upper body detector to find the people in every frame. The found detections are clustered to form tracks. Then, they calculate descriptors along the tracks and use SVM classifier for each interaction class by using these descriptors. In addition to these attributes, head orientation information is also thought to be useful for recognition by adding to the attribute, assuming that people are facing / looking at each other when movements take place. Figure 3.9¹ shows an example descriptor.

Hoai et al. [22] also proposed a method that finds the tracks of the upper bodies of the people and for each upper body track, a track-focused descriptor is then computed based on dense trajectory descriptors (DTD) [20] which encode gradient and motion cues along the trajectories. They also compute a HOG-based scene descriptor, which is the average of HOG descriptors computed on key frames. Thus, a video is represented by a human-focused descriptor and a HOG-based scene descriptor. Then, they use SVM classifier for each interaction class by using both scene and motion along the upper body.

Ikizler and Forsyth [26] suggested a finite state machine to model the human actions. They track the body limbs (arms, legs) and lift the tracks to the 3-D space and build a model of the action of each limb (arms, legs) for each action using Hidden Markov Model (HMM). The

¹The Figure 3.9 is reprinted, with permission, from [21] ©2012 IEEE.

smaller action models are combined with larger HMMs to represent complex actions. Since the action models are based on the tracking performance, the tracking performance effects the action recognition performance.

Y. Kong and Y. Fu [64] proposed the max-margin action prediction machine for recognizing actions in incomplete videos. They formulate the action prediction task as a structured SVM learning problem by using the dense trajectories and interest points, and incorporate composite kernels to capture nonlinear classification boundaries in the prediction task. Their method requires bounding boxes or a person detector.

Gaidon et al. [23] represented the videos as a tree of clustered point trajectories. The authors clustered the trajectories into representative motion parts by building a cluster tree and they embedded the cluster tree into their descriptors. Burghouts and Schutte [65] modeled the spatio-temporal layout of 48 actions by using the location of STIP features. They generated six different layouts based on the combination of three different coordinate systems, use of feature posterior probabilities (which are generated from learned probability density functions of the feature locations) and use of the standard histogram representation with BOW.

Binlong et al. [66] proposed a method to exploit the temporal structure by comparing the principal angles between subspaces representing the activity types. They generated Hankel matrices by use of trajectories to model the temporal information.

Yu et al. [67] employed a Hough-transform based voting method to recognize human actions. Since the performance of the Hough voting is highly dependent on the quality and the amount of the input data, the authors suggested to use propagative Hough voting by using random projection trees (RPT). With their method, feature voting is done by using RPT rather than using the votes of local features individually.

Vahdat et al. [68] proposed a model that represents actions as a sequence of poses. The method requires complete tracks of actors across the entire sequence and the occlusions may cause problems. Since our model uses local features, occlusions have minimal effect on the performance. Ma et al. [24] proposed a method that discovers a compact set of hierarchical space-time tree structures of human actions from training videos by using the hierarchical space-time segments [69]. Using an ensemble of the discovered trees, or in combination with simpler action words and pairwise structures, they build action classifiers that achieve state-of-the-art performance on two challenging datasets: High Five [39] and UCF-Sports [70]. Our method mines the local interest points instead of higher level space-time segments and

we use discrete sequence mining methods instead of tree mining and our method can capture lower level features by using the local features.

Ma et al. [24]’s method is similar to our method in terms of discovering space-time patterns. Our method is different than the method of Ma et al. [24] in 3 different ways:

1. Our method mines the local interest points instead of the high level space-time segments and we use discrete sequence mining methods instead of tree mining.
2. By using the local features, our method can capture lower level features and combine the lower level information with the higher level temporal patterns. According to the results on the High Five dataset, our method outperforms the Ma et al. [24]’s methods.
3. With the hierarchical temporal pyramid structure representation as defined in Section 4.4, our method can model long-term and short-term temporal structure of the actions.

Recently, Liang et al. [25] proposed hierarchical feature representation structure for affective interaction recognition based on the local feature and mid-level feature descriptions. They extract local features and contours from the spatio-temporal segments of the video and represented as trajectories. Then, the local features are clustered to form a dictionary. The final video descriptor is represented as the concatenation of the spatio-temporal histograms of the video segments using the calculated dictionary. Our method is different than the method of Liang et al. [25] in the descriptor design. We exploit the pairwise spatio-temporal relations between the local features via data mining methods instead of trajectory generation.

When we look at the methods described in this section and Section 3.1, it was seen that a significant part of the methods tries to find the temporal relation between the segments forming the action and the frequent and distinctive patterns within each class. In fact, it is known that the field of data mining is related with finding common sequences that frequently resulting from a set of actions labeled with time tags. In the section 2.6, it will be explained how the literature study on the subject of data mining can be used in the recognition of human actions.

3.3 Models Focused on Key Components

Key frames are also commonly used for action recognition, Carlsson et al. [27] use a manually selected frame for classification of non-complex actions. Liu et al. [28] used Adaboost to select the key frames. Raptis and Sigal [29] proposed to use a more compact key frame subset (up to 4 frames or %4 of frames).

Sefidgar et al. [30] proposed a key component model that selects a set of key-components, discriminative moments in a video sequence that are important evidence for the presence of a particular interaction. They use object trackers and find out the interaction between the objects.

3.4 Deep Learning Methods

In the computer vision area, Tran et al. [31] proposed a spatio-temporal feature learning method using deep three dimensional CNNs. Karpathy et al. [32] studied the performance of convolutional neural networks in large-scale video classification, using one million videos in 487 categories (Sports-1M dataset) obtained from YouTube videos.

There are different methods [33–35] proposed to model the temporal dynamics of the actions with deep learning methods. Simonyan and Zisserman [35] capture separate spatial and temporal recognition streams based on CNNs. Figure 3.10 shows the network structure together with the network inputs and outputs. There are two network streams: one of them is for spatial stream and the other one is temporal stream. The spatial stream accepts a single RGB frame input and the temporal stream accepts 10 frames optical flow figures. They combine the spatial and temporal networks at the soft-max layer either averaging or a linear SVM.

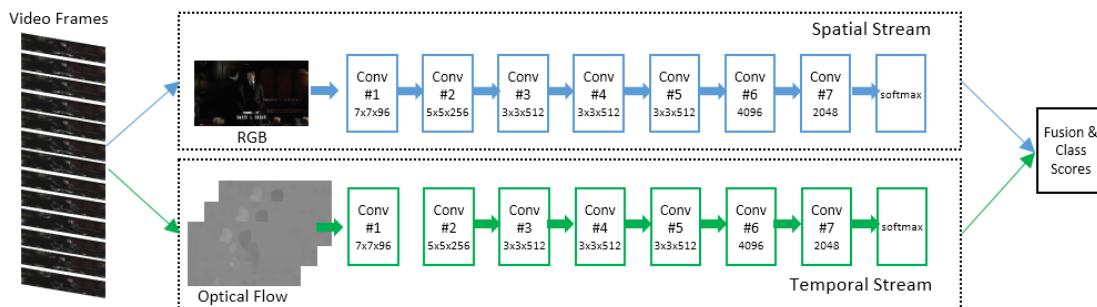


FIGURE 3.10: A representation of two stream model.

Feichtenhofer et al. [33] proposed an improvement to [35] that is able to fuse spatial and temporal cues at several levels of granularity in feature abstraction. Fusing spatial and temporal network in the early levels of the network provides the capability to discover spatial and temporal relations of action components. According to the results, they showed that learning correspondences between spatial and temporal networks has great importance for action recognition. Figure 3.11¹ shows an example network architecture for fusion of spatial and temporal network at different layers. They also analyzed the method for fusion strategies such as: max, dot product, mean and sum. They also showed the using improved dense trajectories [6] improves the final recognition performance.

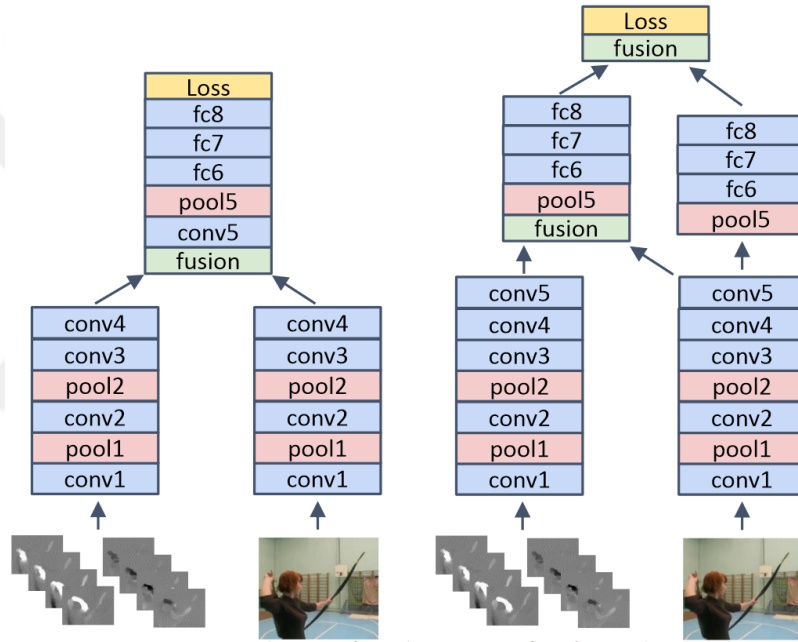


FIGURE 3.11: A representation of Feichtenhofer et al.'s [33] model.

Li et al. [34] presented a multi-granular deep architecture, which is able to incorporate information at a multitude of granularity, including frame, consecutive frames (motion), clip and the entire video. The frames and consecutive frames (motion) are provided to 2D convolutional neural networks. The clips and the entire video is provided to the 3D convolutional neural networks. They employed LSTM networks to incorporate long-term temporal modeling based on the granularity features. They fuse the scores from each network by using a weight. The weights of the networks are again learned via the stochastic gradient descent method. Their results showed that including multi-granular data improves the recognition performance since it captures the hierarchical structure of the actions.

¹The Figure 3.11 is reprinted, with permission, from [33] ©2016 IEEE.

3.5 Mining Methods used in Computer Vision

Li et al. [71] uses association rule patterns of mid-level image patches to classify the objects and scenes. They extract convolutional neural network (CNN) activations from images with a sliding window method. The images are represented as a transaction which consists of multiple convolutional neural network activations and an additional class label. They use association rule mining methods to find out the patterns of mid-level image patches for each class. The association rule mining algorithm finds out the sets of mid-level image patches that leads to interested class. The representativeness and discriminativeness of the discovered patterns are satisfied by using *support* and *confidence* criteria. The *support* related with the representativeness and the *confidence* relates with the discriminativeness. For multi-class recognition problems, having an association rule that has a confidence value greater than the $min_{confidence}$ does not necessarily mean that it is a discriminative rule, since that association rule may also be common for all other classes.

In the action recognition problems, Vito et al. [72] proposed a method called as *Histogram of Patterns* to recognize the human actions from 3D human skeleton data. They use a modified version of Apriori [56] algorithm to discover the frequent patterns. They changed the Apriori [56] algorithm to discover the sequential patterns of 3D human skeletons by using the sliding windows method on the videos. They try to find out the temporal windows that contain frequent skeleton patterns. Vito et al. [72]’s method is the closest method to our method. Our method is different than the method of Vito et al. [72] in 4 different ways:

1. They claim that the regular data mining algorithms discard the order of transactions (which is correct, for association rule mining), so that there is a need to modify the association rule mining algorithms to in order to find out the frequent ordered pattern. In our method, we consider videos as a time ordered series of frames and frames has a linearly increasing time-tag. With this representation, we can use the sequential-pattern mining methods. The method of Vito et al. [72] can capture only the perfectly ordered series where there should not be any temporal gap within the pattern, whereas our method can capture any sequential pattern either there is a temporal gap between the interested items (e.g. Local feature points or body parts)
2. Our method can capture co-occurring feature points. The co-occurrence information is important to discover the pairwise relations between the parts of actions, images or scenes.

3. With the hierarchical temporal pyramid structure representation as defined in Section 4.4, our method can model long-term and short-term temporal structure of the actions.
4. Our method finds the not only the frequent patterns, but only the discriminative patterns by applying a sequential pattern selection after generation of frequent patterns. The method of Vito et al. [72] selects the mined patterns based on the frequency of the patterns. They show that recall values can decrease with the use of frequent patterns and they try to find a cut-off frequency for patterns which has the best performance for discrimination of classes. We think that selecting cut-off frequency to filter the patterns is not a good method since in some scenarios, it is acceptable to have the greatest frequency for class specific patterns.

Mabroukeh and Ezeife [73] reviewed the literature in 2010 by categorizing the algorithms. Mabroukeh and Ezeife [73] sum up the algorithms of sequence mining in three groups: Apriori-based algorithms, early pruning algorithms, and pattern growth.

Apriori-based algorithms first generate pattern candidates, and then calculate the frequencies on the database and pruning the patterns, since the breadth-first search is performed. The Generalized Sequential Pattern Mining (GSP) [74] algorithm suggests that candidate patterns should be generated from the existing set of items rather than using all items for efficient utilization of the memory. Even if GSP is faster than AprioriAll, there are many sectors to be improved. As an improvement to the GSP algorithm, Ayres et al. [75] have made the GSP algorithm even more efficient by converting the GSP [74] algorithm to depth first and providing some improvements.

The main purpose of the pattern generation algorithms is to reduce the size of the search space rather than determining candidate patterns. For this purpose, Pei et al. [57], for each pattern candidate, they created a subset of the global database. This method is called as pseudo-projection (pseudo-projection) from the database (projected according to the pattern candidate) and computations were performed on this smaller database. In this way, both less data is processed and unnecessary pattern candidates that could be created in the future were avoided.

In the early-pruning algorithms, some preliminary checks are performed on the database, and some statistical metrics are collected for each element, and this information is used for

pruning to reduce the search space. Yang et al. [76] suggests LAPIN algorithm and LAPIN algorithm suggests that for each element search for the most recent and first position in the corresponding sequence should be determined, and search only among these positions by pattern growth algorithms. In this way, they have achieved a considerable increase in performance on sparse databases. The fact that a database is sparse or dense is not related to the number of records, but to the number of unique elements it contains, or to the length of the sequences.

Apart from the methods described above, hybrid methods such as SPADE [77] (Apriori and pattern growth) and PLWAP [78] (pattern growth and early pruning) were also suggested and succeeded. Since these methods have basically different methods of calculating each other, they will produce the same output when the time and memory requirements are met. In this context, it is appropriate to our method to use a mining method which are not too complicated, but exhibit sufficient performance to be used in computer vision problems. Among these algorithms, the PrefixSpan [57] algorithm is one of the suitable algorithms to use because it can be seen that it works more efficiently than the other methods with high amount of data. Less complexity of the PrefixSpan algorithm than the other methods is an important advantage on the others.

4 HISTOGRAM OF SEQUENCES

Our proposed framework involves i) extraction spatio-temporal interest points, ii) filtering interest points, iii) visual word construction, iv) hierarchical temporal pyramid construction, v) sequence mining and vi) classification steps. Below, we describe each of these steps in detail. The feature mining approach and construction of this new representation is illustrated in Figure 4.1.

4.1 Extracting Local Features

In computer vision community, local feature points have been proven to have good performance on matching and recognition. The attractiveness of local features comes from the fact that they are invariant to scale and occlusions, due to their locality nature. For this reason, as the low-level representation, we choose to use space-time interest points (STIPs) [9]. The detail of the STIPs are provide in Section 2.1.2. In this work, we use the provided STIP detector of [9] to detect the local feature points.

4.1.1 Filtering Interest Points

Since the spatio-temporal interest point (STIP) extraction procedure looks for high variations in both spatial and temporal domain, it is not robust to the camera movements. When the camera is moving, the points that have high variation in the spatial domain are likely to have variation in the temporal domain as well. Especially, in the videos with complex backgrounds which are likely to have many corner-like structures, the number of detected feature points increases enormously.

To filter the interest points that belong to the background, we make use of a motion segmentation algorithm which analyzes long term point trajectories and applies spectral clustering [36] to find the motion segments within a video. Brox’s motion segmentation algorithm [36] follows long term point trajectories rather than 2-3 frame motions by employing a point tracker [79].

Once the motion segments are obtained, we assume that the segment that has the maximum spatial deviation is the background segment and assume that the remaining segments are the foreground segments. By identifying the background segment in this way, we can filter out the interest points that fall into background segment. Example results of the filtering step

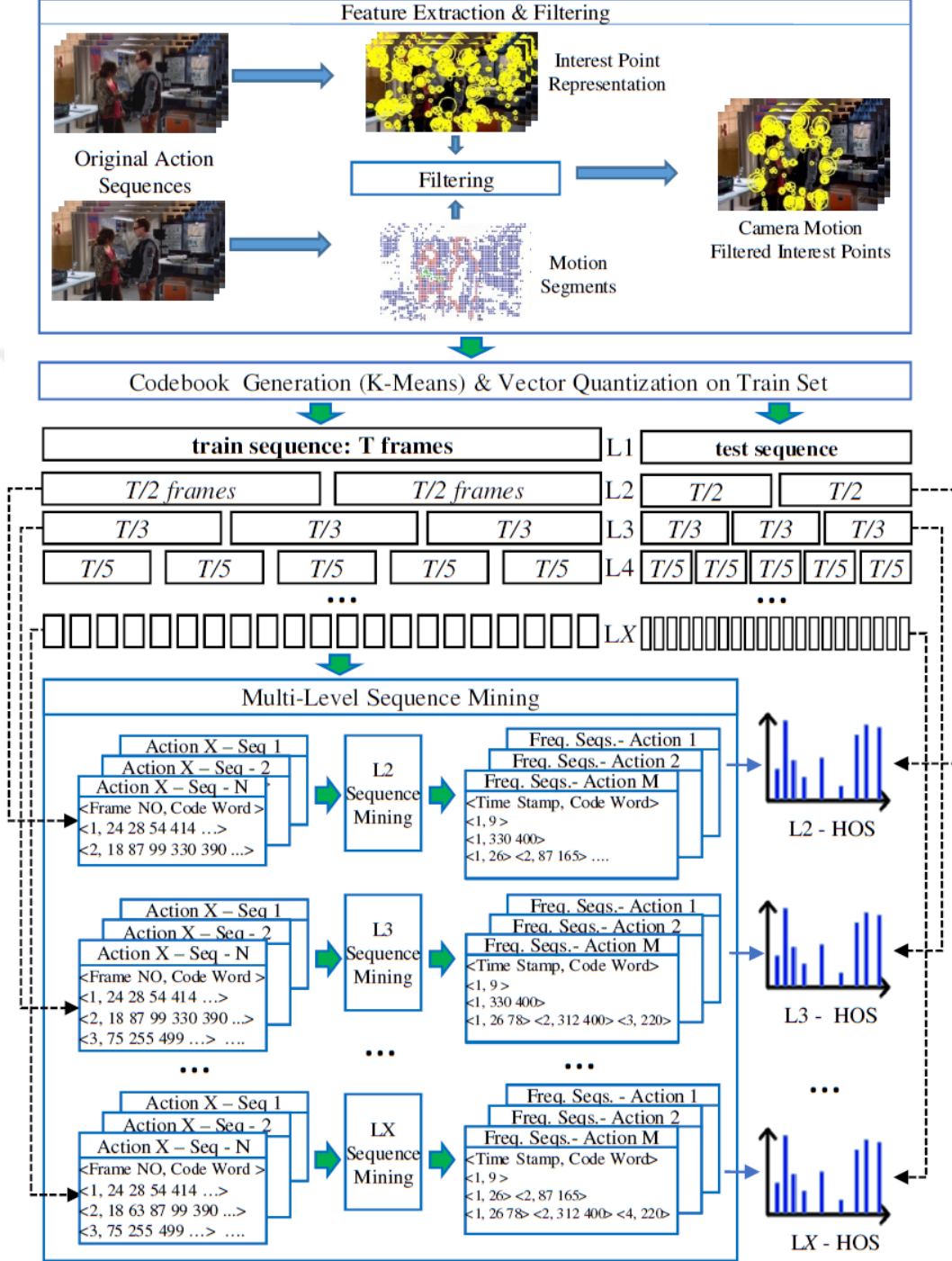


FIGURE 4.1: Overall framework of the proposed method.

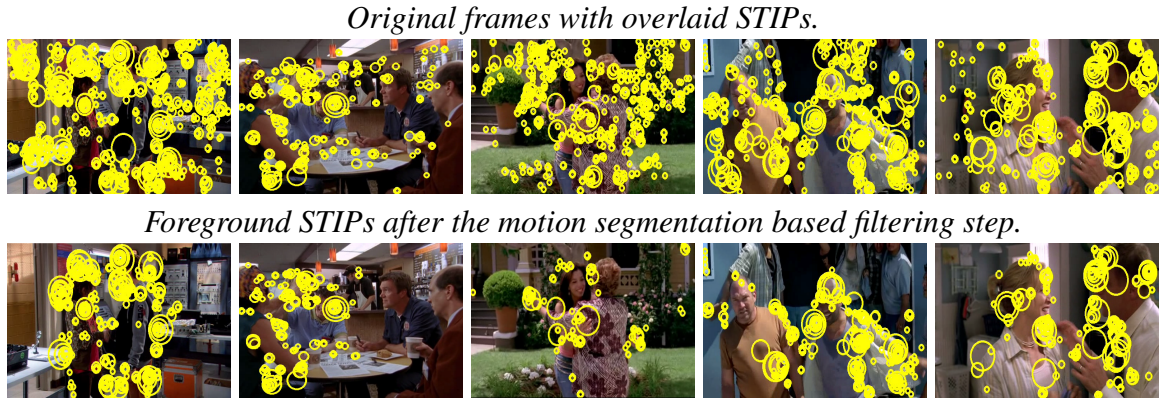


FIGURE 4.2: Example frames for the STIP filtering step on the TV Human Interaction dataset [39].

are shown in Figure 4.2. While not perfect, a great portion of the STIPs that fall onto the background region are eliminated this way. Original frames are shown to the left of the figure and the filtered STIPs as the result of background segment identification is shown on the right. The spatio-temporal interest points that lie on the detected background segment is filtered out, and only foreground STIPs that lie on the foreground segments remain.

After extracting and filtering local STIPs, local feature descriptors are extracted around each STIP. In order to capture both the shape and the motion information around the STIPs, we choose to extract the motion boundary histogram (MBH) features around the STIPs. Section 2.3.3 describes the MBH features.

The temporal and spatial scale values of the STIPs are also used in MBH computation to form the encompassing grid of interest. We have combined the implementation of the STIP [80] and MBH [6] provided by the authors.

4.2 Extracting Deep Features

The deep features can also be used with our method. Feichtenhofer et al. [33] trained one spatial and one temporal VGG-16 network on the UCF-101 dataset. We have used the pre-trained VGG-16 [52] networks from the Feichtenhofer et al.’s [33] method.

The UCF101 Dataset [81]: The UCF-101 is an action recognition data set of realistic action videos, collected from YouTube, having 101 action categories. With 13320 videos from 101 action categories, UCF-101 is diverse in terms of actions and with the presence of large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered



FIGURE 4.3: An example optical flow image from TV Human Interactions Dataset.

background, illumination conditions. The action categories are divided into five types: 1) Human-Object Interaction 2) Body-Motion Only 3) Human-Human Interaction 4) Playing Musical Instruments 5) Sports.

The spatial network uses RGB frames of the videos at a resolution of $224 \times 224 \times 3$ where 3 is the color channels. When the size of input frame is different than the 224×224 , it is rescaled to 224×224 . The mean of all frames is subtracted from the inputs to have zero centered inputs.

The temporal network uses RGB frames generated from the optical flow of the videos at a resolution of 224×224 . And again, when the size of input frame is different than the 224×224 , it is rescaled to 224×224 . The optical flow is densely extracted using the Farnebäck optical flow algorithm [82] (the implementation is available in the OpenCV library) and linearly scaled to $[0, 255]$ range and compressed as JPEG files. The Figure 4.3 shows an example flow image.

Ten horizontal and vertical optical flow components are stacked and provided to the temporal network as suggested by Feichtenhofer et al. [33].

We have fine-tuned the pre-trained VGG-16 spatial and temporal networks on the TV Human Interactions dataset. While fine-tuning, we have replaced the last fully connected layer with a new fully connected layer that outputs class scores for each class. The learning rate for the previous layers are decreased exponentially according to the distance from the last fully connected layer. To increase the number of training examples, offline data augmentation (flipping, scaling&cropping) is applied.

The features are extracted using the pre-trained spatial and temporal networks. For the spatial network, to encode more local data, the features are also extracted by using a sliding window method with a stride of 75 pixels which corresponds to the 1/3 of the original network input image size. We have extracted features from FC6 and FC7 layers. The size of the features are 4096 for both layers.

The temporal network outputs a single feature for ten consecutive frames, which is not sufficient for sequence mining. To increase the number of features we have extracted features with a temporal sliding window with a stride of 1 frame so that at each frame we have extracted a single descriptor.

4.3 Visual Word Construction

The visual vocabulary is the set of words in our model, which provides an easy way for representing a frame of a video. One can use the all feature points in comparison to test the new videos if the data is small enough to process. However, since the dimensionality is very large, it is hard to directly use and compare all feature points. The number of feature points computed for all video sequences is so much and it takes so long time to classify new videos. To represent videos we used the time labeled bag of words model inspired from text retrieval, visual vocabulary is constructed by quantizing the feature points into smaller clusters. For this purpose, we adopt the classical vector quantization procedure that uses k-means to cluster extracted descriptors into $k = 1000$ words, where cluster centers become the visual words of the final visual vocabulary.

Since we need a discrete representation to apply the subsequent sequence mining procedure, we employ soft clustering in order to compensate for information loss because of the hard-assignment based vector quantization. In this formulation, rather than mapping a visual feature to a cluster center, we map it to the closest k clusters. We select the closest clusters having a distance less than a threshold. The threshold $\tau = c * \sigma$ is defined as a multiple

of the standard deviation σ of all distances to all cluster centers. At the end, each frame is vector-quantized via this soft-clustering procedure and represented with the set of visual words that appear on that frame.

As an example, let's assume $s = \langle (5, 10\ 200\ 311), (7, 55) \rangle$ is a video representation without soft clustering, which have three local features at time 5 and a local feature at time 7. Now, the new representation is, $s' = \langle (5, 10\ 65\ 200\ 220\ 224\ 311), (7, 55\ 70\ 400) \rangle$, where the clusters 65, 220, 224 in frame 5 and the clusters 70 400 in frame 7 are now added since the distances of the cluster centers to the feature points are less than the threshold τ .

4.4 Hierarchical Temporal Pyramid Structure

When we look at the variations in the speed and the duration of actions, a single short-term temporal model may not cover the temporal structure of the actions. Some actions are sustained on the longer time period, whereas some actions are performed within a short time. The local features do not explain long term actions well and the sequence mining procedure is not robust to small changes in the speed of execution of the actions. To solve this problem, we propose a temporal pyramid construction.

Temporal pyramid construction is done as follows: The videos are divided into fixed size subsequences at the temporal dimension. At each level of the temporal pyramid, we divide the video into the T parts where the T is the corresponding Fibonacci number for that level, such as 2, 3, 5, 8, 13, 21 and so on. The Figure 4.4 shows an example pyramid structure.

The features in the same temporal window are assumed to occur at the same time interval and therefore assigned the same temporal label in the mining process. This process is illustrated in Fig. 4.4. As an example, at level 4, the videos are split into 5 temporal segments and the time label of the features are assigned to values from 1 to 5 based on the temporal location. The depth of the temporal pyramid defines the temporal subsequence size. At the higher levels of the pyramid (like $L2$ in Fig. 4.1), the mining algorithm finds the sequential patterns that are common in the long term. At the lower levels of the pyramid (like LX in Fig. 4.1), the mining algorithm finds the common short term sequential patterns.



FIGURE 4.4: An example representation of temporal pyramid structure.

4.5 Mining Sequential Patterns

In our case, the aim is to identify the frequently appearing sequential patterns for human interactions in videos. To this end, we need to run the mining algorithm to find frequent sequential patterns for each class. In this action recognition domain, the *ISDB* (defined in Section 2.6) corresponds to set of class specific video sequences V_i such that $ISDB = \{V_1, V_2, \dots, V_m\}$. A video sequence is an ordered set of frames and correspond to the interval extended sequence such that $V = \langle (t_{1,1}, F_1), (t_{1,2}, F_2), \dots, (t_{1,n}, F_n) \rangle$, where F_i is a video frame and $t_{1,i}$ is the frame number. A frame F_i corresponds to an itemset A , where each detected visual word correspond to items of the itemsets. When the hierarchical temporal pyramid structure is used, the sequential mining constraints

A fair representation of each class in the final set of patterns can be guaranteed by the equal number of participants from each class. For finding an equal number of candidate sequences from each class, we utilize the $min_{support}$ threshold. In our experiments, the $min_{support}$ values are automatically selected so that requested number of sequential patterns are mined for each class.

Figure 4.5 shows extracted sequential patterns pictorially. Each pattern may have a different temporal length. There may be sequential patterns at length 1 and they represent co-occurring features in the same frame. Also, same pattern may exist in different temporal locations in a

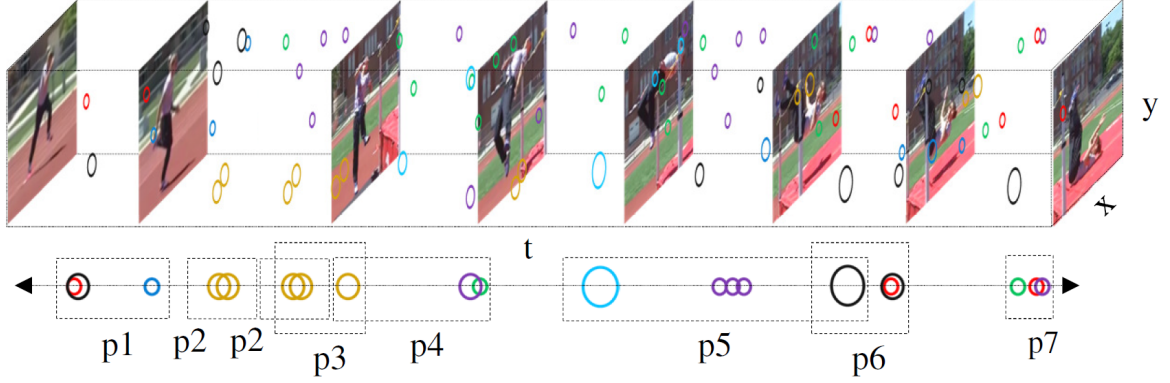


FIGURE 4.5: Pictorial representation of the discovered sequential patterns (when the temporal pyramid is not used).

video (e.g. p_2 in the Figure 4.5). The circles in the figure represent the local interest points and the color identifies different types of interest points in terms of visual words. The size of the circles represents the spatio-temporal scale of the features. Each pattern is identified with a number (pX) and the features in a discovered sequential pattern are grouped within a box. The length of the box determines the length of the sequential pattern. There may be overlapping features within a frame set (e.g. p_2 , p_3 and p_4).

Figure 4.6 shows extracted sequential patterns pictorially when the hierarchical temporal pyramid structure is used. L_1 through L_4 represents the levels in the pyramid (we have limited the number of levels to 4 for clarity of the figure). The maximum length of sequential patterns can be 2, 3, 5, 8 and 13 at levels 1, 2, 3, 4 and 5, respectively. There may be sequential patterns at length 1 such that they represent co-occurring features in the same temporal slice. $T_1, T_2, \dots, T_x$ labels represent the temporal label of that temporal part. Also, there may be different sequential patterns at each level.

4.6 Calculation of Histogram of the Sequential Patterns

In our framework, each frame is represented as a time-labeled discrete set of code-words. The set of code-words occurring in each frame is considered as the itemset for that frame, and each itemset is labeled with the time of occurrence. The obtained frequent feature sequences are used as a codebook and each video will be represented as the histogram of codebook elements. The $pttrn$ in Algorithm *countPatternOccurrence* 3 in section 2.8 corresponds to a sequential pattern. The size of histogram is same as the number of extracted patterns.

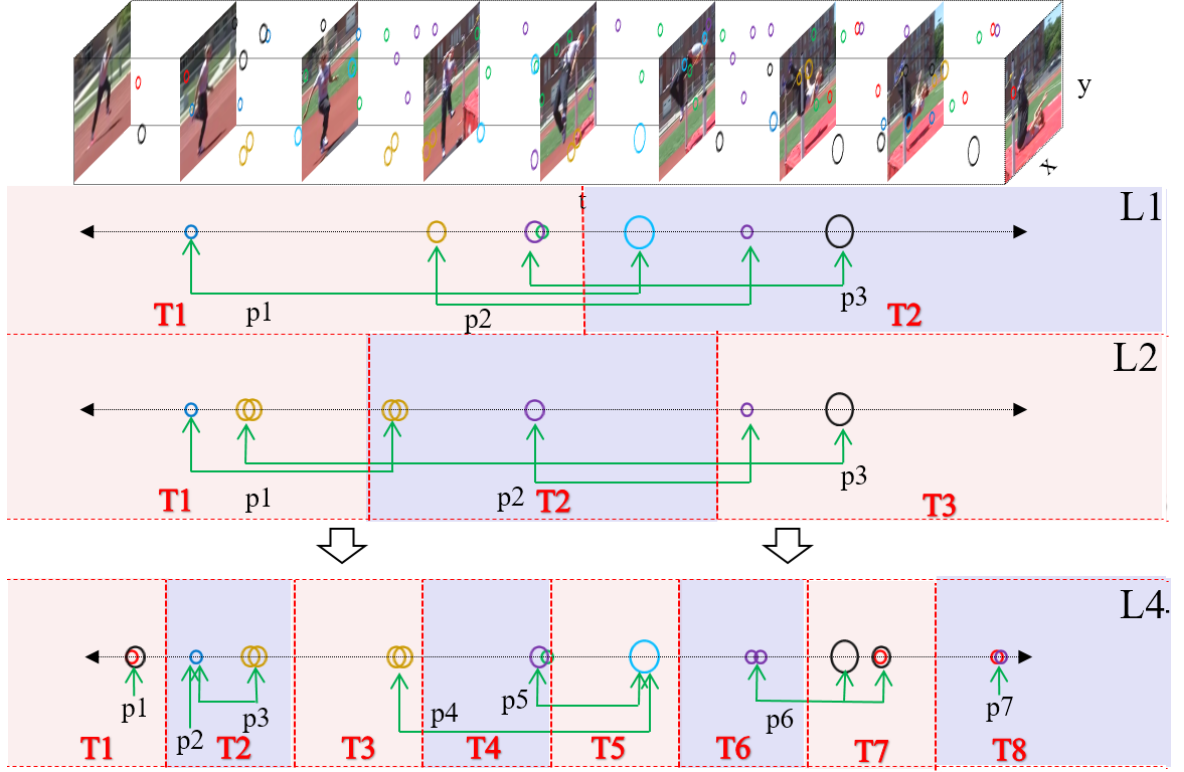


FIGURE 4.6: Pictorial representation of the discovered sequential patterns (when the temporal pyramid is used).

The complexity of this algorithm is $O(k * m * n)$ where k is the number of the sequences, m is the length of the sequence and n is the length of the pattern.

4.7 Sequential Pattern Selection

For the application of sequence or frequent itemset mining algorithms, there is a well known trade-off: If the minimum support value $min_{support}$ is kept large, co-occurring sets of items can be very limited and may not cover all the interesting sub-patterns. On the contrary, if the $min_{support}$ is kept low, the number of found sequential patterns can be prohibitively large and it becomes infeasible to process them all.

In order to have a more compact and discriminative set of sequences, we employ a selection procedure. The main idea of this selection is to discard the sequential patterns which are not discriminative enough. With this purpose, we first generate a maximum set of sequential patterns by running the mining algorithm for each class and collecting all patterns together. Then, for each class, we train a linear SVM on the calculated histograms using the generated

patterns. The sequential patterns that acquire large weights in these linear SVMs are selected as the final set of sequential patterns. Figure 4.7 shows some example sequences found after using the sequence mining and sequence selection steps. The local feature points are displayed as yellow circles and the size of the circle represents the spatial scale of that local feature.

Figure 4.7(a)- 4.7(b) contains occurrences of a sequential pattern selected $\langle(0; 261) (1; 261) (3; 121) (6; 77)\rangle$ from the UT Interactions dataset, where the temporal pyramid structure is not used. The length of the sequential pattern is 7 frames and the time intervals between the sequential items are: 1 frame, 2 frames and 3 frames as shown in the figure, respectively.

Figure 4.7(c)- 4.7(d) contains a sequential pattern $\langle(0; 51) (1; 61)\rangle$ from TV Human Interactions dataset hugging videos, where the temporal pyramid structure is not used. The length of the sequential pattern is 2 frames and the time interval between sequential items is one frame.

Figure 4.8(a)- 4.8(b) contains a sequential pattern $\langle(0; 451) (4; 307) (5; 390)\rangle$ from UT Interactions Dataset handshake videos. The length of the sequential pattern is 5 frames and the time intervals between sequential items are: 4 frames and 1 frame as shown in the figure, respectively.

Figure 4.8(c)- 4.8(d) contains a sequential pattern $\langle(0; 50) (4; 32)\rangle$ from TV Human Interactions handshake videos. The length of the sequential pattern is 5 frames and the time intervals between sequential items is 4 frames as shown in the figure, respectively.

4.8 Building Up Histogram of Sequential Patterns

After selection step, we have a set of discriminative sequential patterns for each class and the videos are represented as a histogram of sequences (HoS) where the attributes are the final set of selected sequential patterns and the values are the normalized number of occurrences of those sequential patterns. More specifically, let $P = \{p_1, p_2, \dots, p_k\}$ be a set of sequential patterns which includes all patterns selected for each class and k is the number of sequential patterns. The histogram representation is:

$$H_\alpha = [f(\alpha, p_1, 0, 0, 0), f(\alpha, p_2, 0, 0, 0), \dots, f(\alpha, p_k, 0, 0, 0)] \quad (7)$$



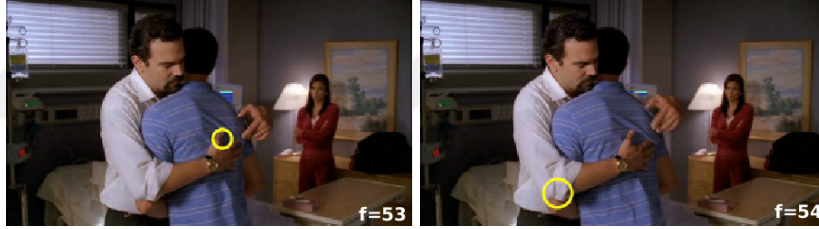
(a) *Pushing-1*



(b) *Pushing-2*



(c) *Hugging-1*



(d) *Hugging-2*

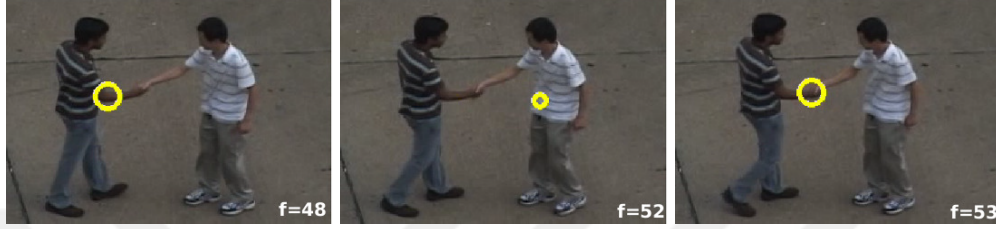
FIGURE 4.7: Example patterns that are discovered using our approach from the UT Interactions dataset (Figure 4.7(a)- 4.7(b)) and TV Human Interactions dataset (Figure 4.7(c)- 4.7(d)).

where α a sequence, f function counts the number of pattern p_i occurrences in α as defined in Algorithm 3 in section 2.8. The $pttrn_idx$, seq_idx and num_occur values are set to 0 as the initial values in Algorithm 3 in Section 2.8.

When the temporal pyramid structure is used, we concatenate the histograms from each class at each temporal pyramid level into a single histogram and the same classification method is applied. If we assume the L as the number of levels in the temporal pyramid, there will be $2^L - 1$ temporal segments. The video sequence a is temporally split into 2^{i-1} parts at level i such that $\alpha^i = \langle (1, A_1^i), (2, A_2^i), \dots, (2^{i-1}, A_{2^{i-1}}^i) \rangle$ where $0 < i \leq L$ and A is the



(a) Handshaking-1



(b) Handshaking-2



(c) Handshaking-3



(d) Handshaking-4

FIGURE 4.8: Additional discovered sequential patterns.

itemset within that split. Also, the itemsets at level i will be the concatenation of lower level itemsets: $A_j^i = [A_{2*j-1}^{i+1} A_{2*j}^{i+1}]$ where $0 < j \leq 2^{L-1}$. Then, the histogram at level i is:

$$H_\alpha^i = [f(\alpha^i, p_1^i, 0, 0, 0), f(\alpha^i, p_2^i, 0, 0, 0), \dots, f(\alpha^i, p_k^i, 0, 0, 0)] \quad (8)$$

After the histograms at each level is concatenated, the final histogram pyramid for sequence α becomes concatenation of the histograms at all levels:

$$HoS_{\alpha} = [H_{\alpha}^1 H_{\alpha}^2 \cdots H_{\alpha}^L] \quad (9)$$

This becomes the representation for each sequence and video sequences are classified using this compact representation.

4.9 Classification

Once the histograms of the sequences are calculated, the classification of the interactions can be made with many different types of classifiers. We have selected the Support Vector Machines [59] (SVM) for classification.

Since our histogram of sequences representation contains high-level encoded data, we use the SVM classifiers with a nonlinear kernel which is Radial Basis Function (RBF) kernel.

The SVMs are basically designed for binary classification. There are different methods (e.g. one-vs-all, all-vs-all) for the multi-class classification problems. We have used one-vs-all classification for simplicity. We have trained an SVM classifier for each class and compared the each classifier with the others to determine the predicted class label which is called as one-vs-all. As an example, for a 5 category classification problem, we have 5 different classifier trained each one is C_{θ}^i where $i \in \{1, 2, 3, 4, 5\}$. We calculate class probability as $C_{\theta}^i(x) = P(y = i|x; \theta)$ where $i = 1, 2, 3, 4, 5$. Once the class probability of each class is calculated we have $C_{\theta}(x) = [C_{\theta}^1(x), C_{\theta}^2(x), C_{\theta}^3(x), C_{\theta}^4(x), C_{\theta}^5(x)]$. The predicted class i becomes $\max_i(C_{\theta}^i(x))$.

4.10 Deep Multi-Layer Histogram of Sequences

When using features from deep neural networks, using features extracted from different layers of the network seems successful [34]. The features extracted from the beginning and middle layers of the deep network may improve the performance since the network is structured hierarchically and modeling more information may result performance increase. The features from the beginning layers of the network tend to be very primitive responses to the small filters such as the corners and the edges. As we continue to the end of the network, the

features model high level information, beginning from part of body objects to complete action at the end. Combining low level and high level information in a representation is likely to increase the representative power of the model since it provides opportunity to discover pairwise relations between high level and low level information.

The features at different granularity can also be beneficial in our method. For this purpose, we decide to use features from multiple layers as follows:

- Extract features from multiple layers
- Extract codebooks of different type of features
- Create temporal pyramid structure using the combined visual words by different type of features
- Mine sequential patterns
- Calculate Histogram of Sequences
- Classify

The Figure 4.9 shows the Histogram of Sequences representation when the features from different layers of the deep networks are used.

4.10.1 Features

Discovering pairwise relations between multi-granular features enriches the modeling capability of the proposed representation. To this end, we have extracted from pool3_3, pool4_3, pool5_3 in addition to the features from FC6 and FC7. The size of the pool3_3, pool4_3 and pool5_3 features are $28 \times 28 \times 256$, $14 \times 14 \times 512$ and $7 \times 7 \times 512$ respectively. We choose the pool3_3, pool4_3 and pool5_3 features to evenly capture features from the network. While pool3_3 features capture lower level information such as patches of body or movement, pool5_3 features tend to capture pose like features.

We do not choose the pool2_2 and pool1_2 layers: Firstly, they are at the very beginning of the network and tends to capture corner like points which is similar to the local features and secondly, the total amount of the data needed to be processed for the beginning layers of the network is greater than the current capabilities of our test environment to perform tests in reasonable time.

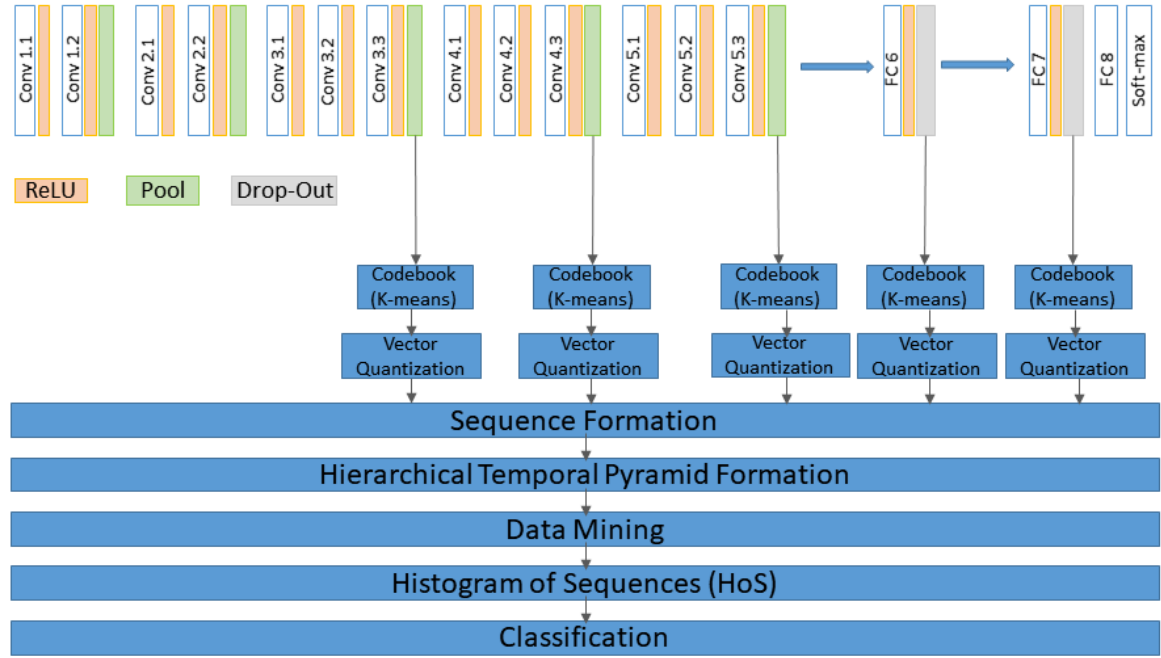


FIGURE 4.9: Deep multi-layer Histogram of Sequences representation.

4.10.2 Visual Word Formation

The visual words are created similar to Section 4.3. The features from different layers of the network are clustered separately via k-means clustering method and visual word representation for each type of features are created. While formation of sequences, features from different layers are numbered (visual words) with an offset to differentiate different layers. The number of clusters is selected as 8000, empirically and soft clustering method is applied as in the baseline HoS method (defined in Section 4.3).

While clustering pool3_3, pool4_3 and pool5_3 features we have grouped the responses of filters to reduce the number of feature types. As an example, each of 512 pool3_3 filters provide 28×28 size response for each frame. We have converted totally $28 \times 28 \times 512$ features to $112 \times 112 \times 32$ by combining 16 (this number is determined empirically.) consecutive filter responses to a single response. The reason behind this grouping, is the lack of data in comparison to the number features. If we would treat each filter response as different features, that would result $512 \times TotalNumberofFrames$ items at size 28×28 to cluster. With grouping filter responses, we think that the information is better captured in the clustering phase.

4.10.3 Hierarchical Temporal Pyramid Structure

Once the visual words for each type of features are created, the temporal pyramid structure is created with the same method as defined in Section 4.4. The video is temporally split into different parts at different levels and time tags of the features are generated according to the temporal split number at each level, as we did in the baseline method.

4.10.4 Mining Sequential Patterns

In the mining step, the features from different layers of the deep network are provided to the mining algorithm so that the sequential patterns those includes features from different layers of the network can be discovered. As an example, a feature from FC6 layer (may be a pose for pointing person) with a feature (may be a corner from knee of other person's body) from pool3_3 layer may be a key sequence for a specific interaction. Same mining method (as defined in Section 4.5) is applied for mining Deep Multi-Layer features.

4.10.5 Histogram Calculation and HoS Representation

There is no special processing required for Deep Multi-Layer Histogram of Sequences. Same method for Histogram Calculation and HoS Representation is applied as defined in Sections 4.6 and 4.8, respectively. Once the Histogram of Sequences representation is obtained for the Deep Multi-Layer features, we followed the same classification method as defined in Section 4.9.

5 EXPERIMENTAL EVALUATION

In this section, we present our experiments for evaluating the recognition performance of the proposed Histogram of Sequences representation for interaction recognition.

5.1 Datasets

In literature, there are many different human interaction recognition datasets. Chaquet et al. [83] provided a survey to provide a complete description of the most important public datasets. We have tested our method on two benchmark datasets of human interactions. These are UT Interactions Dataset [40] and TV Human Interactions dataset [39]. Both of these datasets are collected from sources of real-world videos. These datasets are particularly suitable for evaluating the recognition of more complex activities that involve more than one person or more than a singleton action.

TV Human Interactions Dataset: This dataset contains a total of 300 videos collected from TV programs [39]. There are 200 videos containing four different classes of daily interactions, namely hugging, high five, kissing and handshaking, and 100 videos that do not belong to any class, hence labeled as negative. The Figure 5.1 shows example snapshots from the TV Human Interactions dataset.

While using this dataset, we follow the training and testing split provided by [39]. The dataset is split into two sets, each set containing 25 video clips for each interaction and 50 video clips for negative samples. In this dataset, the human body bounding box annotations are provided, but our method does not require any of such annotations, so we do not use any annotations, other than the class labels in training.

UT Interactions: The UT Interactions Dataset [40] contains videos of continuous executions of 6 classes of human-human interactions: shake-hands, point, hug, push, kick and punch. For the classification task, the dataset provides 120 video segments divided into two sets. The set#1 and set#2 are both composed of 10 video sequences taken in a parking lot, where each sequence contains at least one execution per action class. The Figure 5.2 shows example snapshots from the UT Human Interactions dataset.

The videos of set#1 are taken with slightly different zoom rate, and their backgrounds are mostly static with little camera jitter. The videos in set#2 are taken on a lawn in a windy day.



FIGURE 5.1: Example snapshots from the TV Human Interactions dataset [39]. Note the clutters, camera views, variations of the actions and actors.

Background is moving slightly and the videos contain more camera jitters. For performance comparison, 10-fold cross validation method is used as suggested by [40] for each set. Again, we do not make use of any annotations.

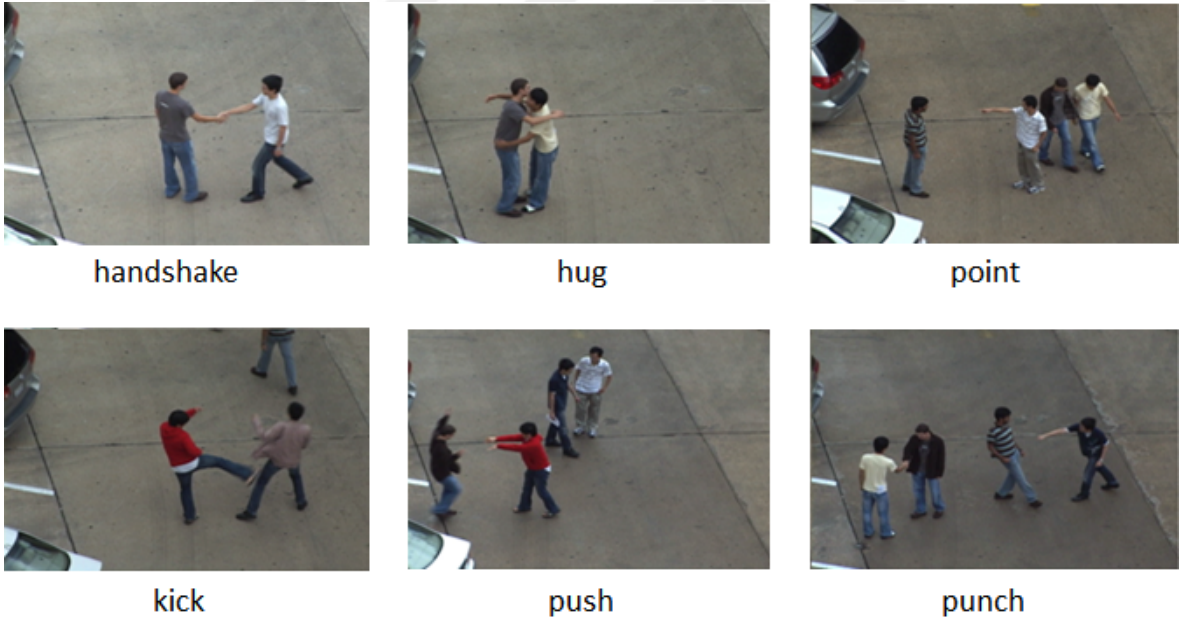


FIGURE 5.2: Example snapshots from the UT Human Interactions dataset [40].

5.2 Experimental Details

After the frequent sequence extraction, since there can be duplicate sequences across the different classes, the duplicate sequences are removed first. The total number of unique

sequential patterns for both datasets becomes less than 3000. The length of the longest sequential pattern at the highest level is set as 6 for the TV Human Interactions dataset and as 16 for UT Interactions dataset. The length of the shortest sequential pattern at the lowest level is 1 for both of the datasets. We observe that the found patterns in TV Human Interactions dataset are shorter than those of UT Interactions dataset. Here, the length corresponds to number of unique frame numbers found in a sequence.

We set the distance threshold (as defined in Section 4.3) for the soft clustering as the standard deviation of all the distances to the all cluster centers.

The sequential pattern mining parameters are set as follows when the temporal pyramid structure is not used: $C1$ (minimum interval between two itemsets) is 0 to cover co-occurrence information as much as possible, $C2$ (maximum interval between two itemsets) is 5 which is found by experimental analysis on the TV Human Interactions dataset, $C3$ (minimum whole interval) is 0 and $C4$ is ∞ to cover all sequential patterns at all lengths. When the temporal pyramid structure is used, we do not put any restriction on the length of the found sequential patterns since the temporal structure inherently provides structure for finding reasonable length sequential patterns.

The UT Interactions dataset has small number of videos and the 10-fold cross validation is used for the final performance on the test set. We have performed a 3-fold cross validation on the training set to select the number of levels in the temporal pyramid and the number of patterns. Since the training set is small, %100 accuracy is obtained for most of the runs. For this reason, we omit parameter selection in this dataset and use the same set of parameters with the TV Human interactions dataset.

We have tested the performance of the HOG/HOF descriptors and the MBH descriptors on both datasets. In the TV Human Interactions dataset, the MBH descriptors perform %2 better than the HOG/HOF descriptors in the total classification accuracy. In the UT Interactions dataset, the MBH descriptors perform %7 better than the HOG/HOF descriptors in the retrieval performance (mAP). The results provided in the following sections are obtained by use of MBH descriptors.

In addition to working with local STIP features, we also experiment with CNN features on the TV Human Interactions dataset as an input to the HoS method. For this purpose, we use the pre-trained VGG-16 based spatial network (split#1 of the UCF-101 dataset) of Feichtenhofer et al. [33] and we fine-tune this network on the TV Human Interactions dataset. We extract

FC6 and FC7 outputs as the input features to sequence mining and evaluate the results. When working with CNN features, we apply the same pipeline except that there is no interest point filtering applied. To see the performance of the features extracted from the different layers of the we have extracted pool3_3, pool4_3 and pool5_3 features and used together with FC6 and FC7 outputs.

Sequence mining with multi-level deep features resulted more sequential patterns in comparison with the other features. This may be due to the pairwise relations between the features at different layers. For this reason, we use the same set of candidate patterns for feature selection without finding number best number of candidate patterns in this setup.

5.3 Evaluation Methods

The results in the literature for the TV Human Interactions dataset is provided as mean average precision value (mAP). The mean average precision is a metric to measure the retrieval performance of a system. The average precision is the area under the precision-recall curve for a given query. It regards the order of the returned results such that when more relevant results are returned at the top of list, the average precision becomes greater. The mean average precision is the mean of average precision values for all test samples in all classes. We have used the implementation of the vlfeat library [84] to calculate the mean average precision value. Formally, the mean average precision AP is defined as:

$$precision = \frac{tp}{tp + fp} \quad (10)$$

where tp is the number of true positives and fp is the number of false positives.

$$AP = \frac{1}{m} \sum_{k=1}^N (P(k)r(k)) \quad (11)$$

where $r(k) \in \{0, 1\}$ is an relevance indicator, $P(k)$ is the precision at a given rank, m is the number of documents and N is the number of retrieved items.

$$mAP = \text{mean}(AP_1, AP_2, \dots, AP_n) \quad (12)$$

where n is the number of classes.

The results in the literature for the UT Human Interactions dataset is provided as accuracy value. The accuracy is the ratio of true positive results to the total test samples when the all test samples have an associated class label (when there is no negative samples).

5.4 Results on TV Human Interactions Dataset

5.4.1 Test with Single Level HoS with Local Features

5.4.1.1 Parameter Selection

There are two important parameters in forming the HoS descriptor: i) the number of class specific mined patterns for each pyramid level, ii) the number of selected class specific patterns for each level of the pyramid after the sequence selection step. We do cross-validation on the training datasets to select these two parameters.

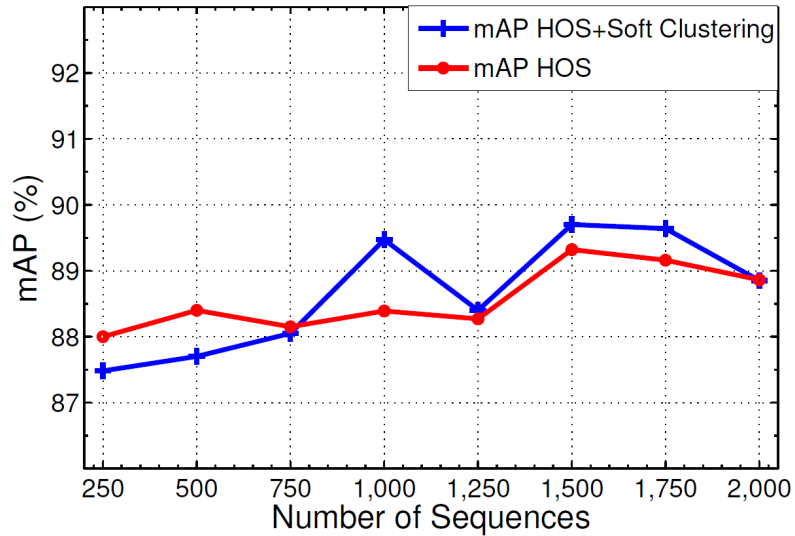


FIGURE 5.3: Local Features - The Effect of number of candidate sequential patterns when the temporal pyramid is not used on TV Human Interactions dataset.

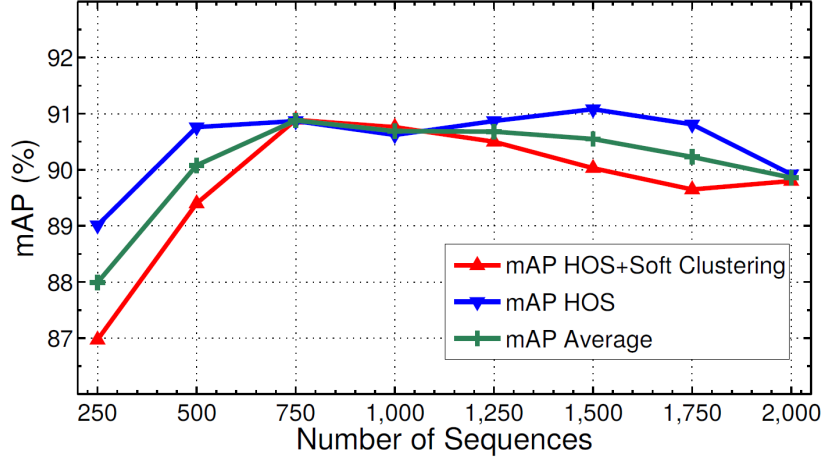


FIGURE 5.4: Local Features - Effect of number of selected sequential patterns when the temporal pyramid is not used on TV Human Interactions dataset after the sequential pattern selection step.

Method	Annotation	mAP
HoS	Not Required	%38.2
HoS (SoftClust)	Not Required	%43.9

TABLE 5.1: Performance of the Single Level HoS method on the TV Human Interactions dataset when the negative videos are included in the retrieval task.

Figure 5.3 shows the effect of the number generated candidate sequential patterns (as described in Section 4.5 before the sequence selection step) on the performance (mAP), when the temporal pyramid is not used, i.e. single level HoS. According to Figure 5.3, the best results are obtained when the histogram size is 1500. Figure 5.4 shows the effect of the number of the class specific selected patterns on the performance after the sequence selection step on the training dataset. The number of the class specific patterns is determined at the sequential patterns selection step as defined in Section 4.7.

5.4.1.2 Performance on Test Set

Table 5.1 and Table 5.2 shows the performance of the single level HoS method on the test set. Table 5.1 includes the results when the negative videos are included. When we look at the results, we see that soft clustering has nearly %6 positive effect on the average recognition performance.

Method	Annotation	mAP
HoS	Not Required	%51.4
HoS (SoftClust)	Not Required	%57.4

TABLE 5.2: Performance of the single level HoS method on the TV Human Interactions dataset when the negative videos are excluded.

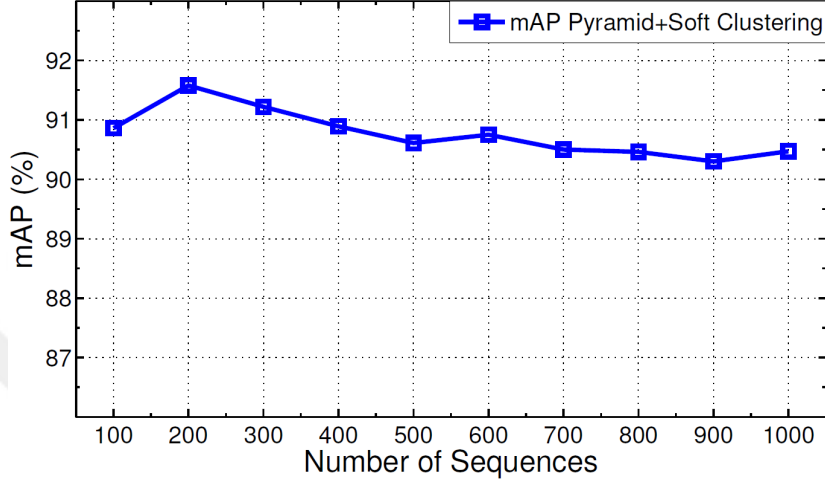


FIGURE 5.5: Local Features - The effect of number of candidate sequential patterns for each class at every level of the pyramid for TV Human Interactions dataset.

5.4.2 Test with Hierarchical HoS with Local Features

5.4.2.1 Parameter Selection

Figure 5.5 shows the performance (mAP) of the number of the generated patterns for every level of the pyramid on the training datasets as described in Section 4.5 before the sequence selection step. The performance is provided with no selection on the features. We have used %20 of the train data for validation. We extract same number of patterns for each level of pyramid in the experiments.

Our intent was to first find the best number of initial candidate sequential patterns and then apply sequential pattern selection. The best performance is found when the number of the generated patterns is 200 which means that the initial descriptor length is $200 * 5 = 1000$ (where 5 is the number of classes) for each level.

The effect of the number of the class specific selected patterns is shown in Figure 5.6. The number of the class specific patterns is determined at the sequential pattern selection step

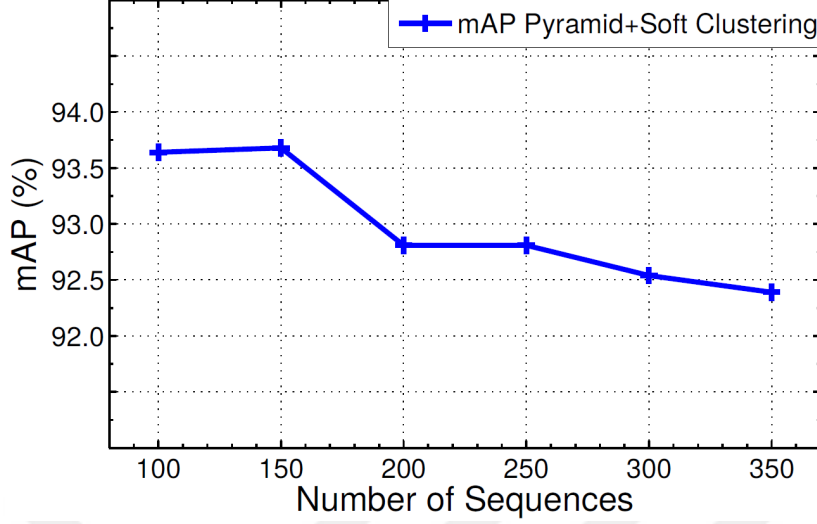


FIGURE 5.6: Local Features - The effect of number of selected sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class after the sequential pattern selection step.

as defined in Section 4.7. The best result is obtained when 150 class specific patterns are selected from each level of the temporal pyramid.

Selecting the number of pyramid levels is another important factor of our HoS representation. For this purpose, we perform 5-fold cross validation on the training set to select the number of pyramid levels. We construct a temporal pyramid up to depth 9. Figure 5.7 shows the contribution of cumulative pyramid levels. $L1$ to LX represents that the histogram of sequences (HoS) representation has been formed using levels 1 to level X . We observe that the recognition performance increases up to level 6, due to the additional information acquired from lower granularity of data in each subsequent temporal level. However, we observe that, after level 6, the recognition performance drops. This is mostly due to the large variations in the short-term activities and the noise caused by short-term motions present in the videos.

5.4.2.2 Performance on Test Set

Table 5.3 and Table 5.4 shows the performance of the HoS method on the test set. Table 5.3 includes the results when the negative videos are included. When we look at the results, the performance gain when using the hierarchical temporal pyramid is nearly +%13 (when the negative videos are included) and +%12 (when the negative videos are excluded) in the

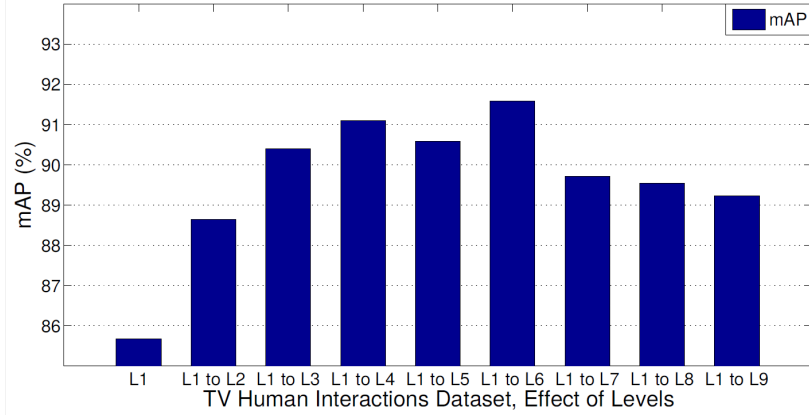


FIGURE 5.7: Local Features - The effect of the number of pyramid levels on the recognition performance on TV Human Interactions dataset. See text for details

Method	Annotation	mAP
HoS	Not Required	%38.2
HoS (SoftClust)	Not Required	%43.9
HoS (SoftClust + TempPyrd)	Not Required	%57.4

TABLE 5.3: Performance of the HoS method on the TV Human Interactions dataset when the negative videos are included in the retrieval task.

Method	Annotation	mAP
HoS	Not Required	%51.4
HoS (SoftClust)	Not Required	%57.4
HoS (SoftClust + TempPyrd)	Not Required	%69.2

TABLE 5.4: Performance of the single level HoS method on the TV Human Interactions dataset when the negative videos are excluded.

TV Human Interaction dataset. We observe that the best performance is achieved by the proposed HoS formulation using soft clustering and hierarchical temporal pyramid.

The Figure 5.8 shows the recognition performance (mAP) of the proposed method and its extensions for each class. We observe that soft clustering has a greater contribution for *Handshake* class than the others, due to the high spatial variability in the execution of this interaction. We observe that temporal pyramid has a greater contribution for *Hug* class, and this is largely due to the temporal extent of the *Hug* interaction videos, i.e. they tend to be longer than other class instances. *Kiss* class as relatively lower performance and when we analyze the reason, we observe that a smaller number of local features have been generated since there is not much significant movement in the execution of this interaction.

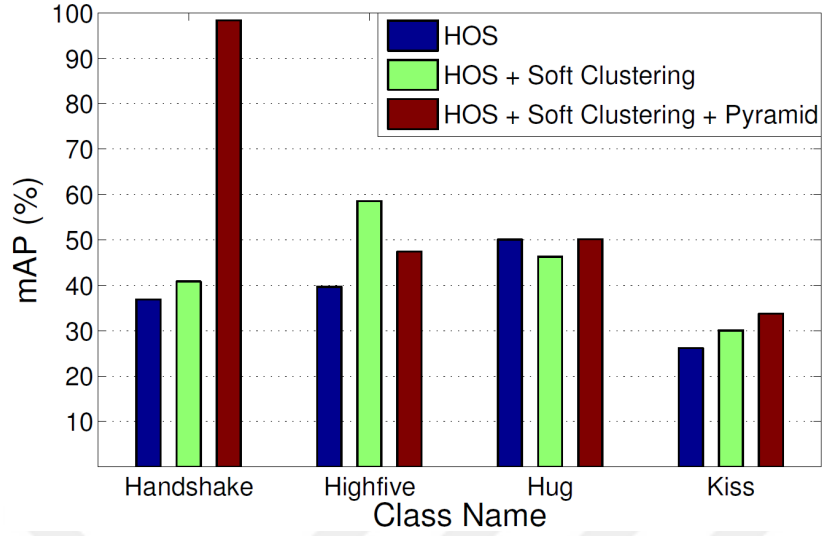


FIGURE 5.8: Classwise mAP performance of the proposed HoS representations, together with its extensions on the TV Human Interactions dataset.

5.4.3 Test with Hierarchical HoS with Deep Features

We have tested seven different descriptors based on the spatial and temporal deep features on the TV Human Interaction Dataset, which are:

1. deep spatial features,
2. deep temporal features,
3. deep spatial features extracted with spatial sliding window,
4. deep spatial features extracted from multiple layers of network,
5. deep temporal features combined with deep spatial features,
6. deep temporal features combined with deep spatial features which are extracted with spatial sliding window.
7. deep temporal features combined with deep spatial features extracted from multiple layers of network,

Similar to the experiments with the local features, we have conducted the same pipeline. The choice for the number of candidate and selected sequential patterns is determined based on the cross validation results on the training set. While performing cross validation, we again

applied 5-fold cross validation so that at each fold the train dataset is divided into two parts where %80 of the data is used in training and %20 of the data is used for validation.

When the temporal features are combined with the deep spatial features (including the scenario the deep spatial features are extracted with the sliding window method), the best number of selected sequential patterns for both type features are used according the results in the following sections, so that there is no parameter selection is needed.

5.4.3.1 Parameter Selection for Deep Spatial Features

The Figure 5.9 shows the effect of the number of the class specific candidate patterns on the performance when the deep spatial features are used. The performance is provided with no selection of the features.

The best performance is achieved when the number of candidate patterns is 700 which means that the initial descriptor length is $700 * 5 = 3500$ (where 5 is the number of classes) for each level.

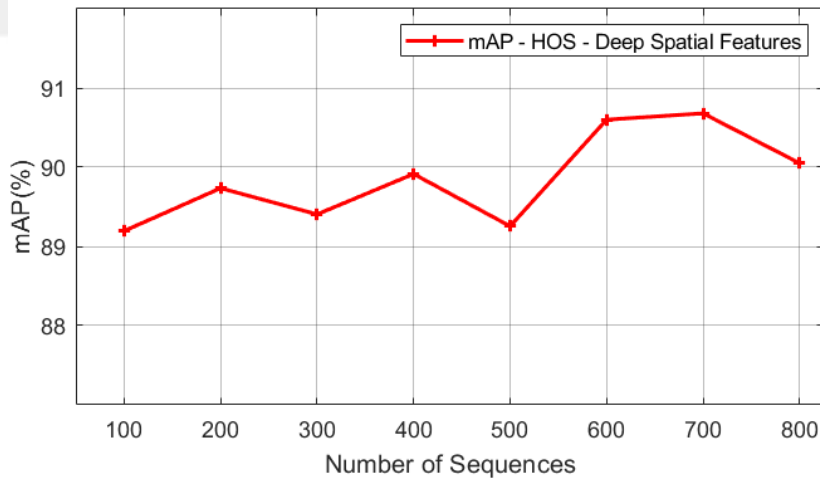


FIGURE 5.9: Deep Spatial Features - The effect of number of candidate sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class.

The Figure 5.10 shows the effect of the number of the class specific selected patterns on the performance when the deep spatial features are used. The selection is performed according to Section 4.7 by using the 700 candidate sequential patterns. The best performance is achieved when the number of selected patterns is 125 which means that the final descriptor length is $125 * 5 = 625$ (where 5 is the number of classes) for each level.

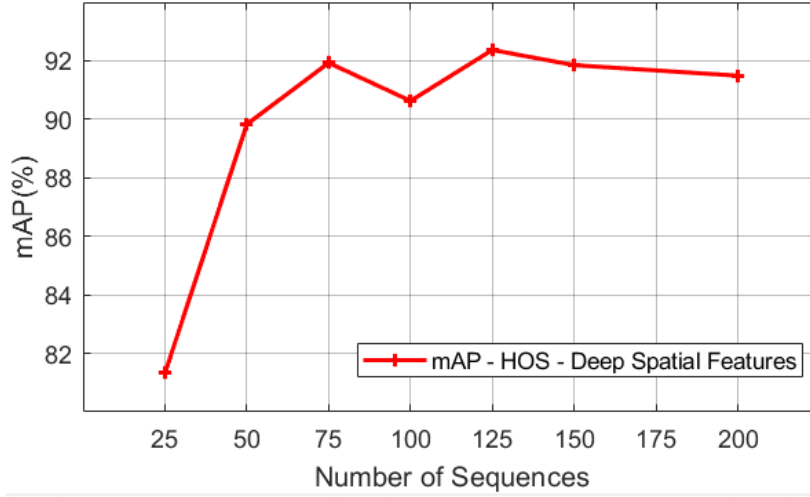


FIGURE 5.10: Deep Spatial Features - The effect of number of selected sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class.

5.4.3.2 Parameter Selection for Deep Spatial Features Extracted via Sliding Window

We have tested the performance of the deep spatial features when we extracted multiple descriptors at each frame. Section 4.2 describes how the deep features are extracted. The Figure 5.11 shows the effect of the number of the class specific candidate patterns on the performance when the deep spatial features are used. The performance is provided with no selection in the features.

The best performance is achieved when the number of candidate patterns is 600 which means that the initial descriptor length is $600 * 5 = 3000$ (where 5 is the number of classes) for each level.

The Figure 5.12 shows the effect of the number of the class specific selected patterns on the performance when the deep spatial features are extracted using sliding windows. The selection is performed according to Section 4.7 by using the 600 candidate sequential patterns. The best performance is achieved when the number of selected patterns is 300 which means that the final descriptor length is $300 * 5 = 1500$ (where 5 is the number of classes) for each level.

When we look at the performance difference of using sliding windows while extracting features (Figure 5.10 vs Figure 5.12), we see that using sliding windows to extract deep spatial features has a negative effect on the cross validation performance. This difference may the

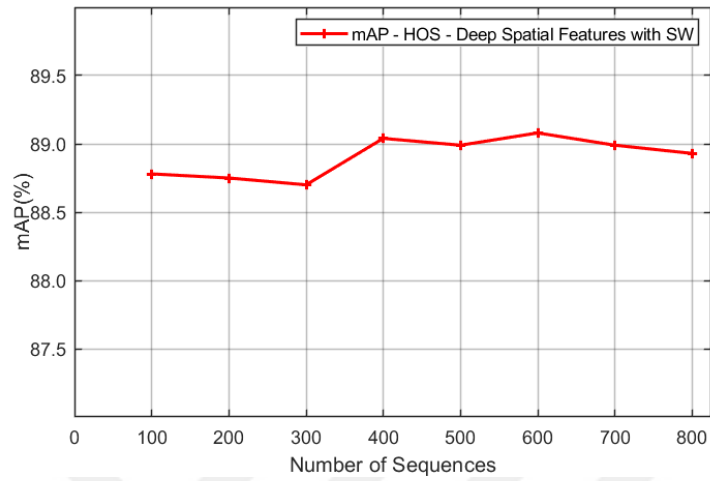


FIGURE 5.11: Deep Spatial Features Extracted using Sliding Windows - The effect of number of candidate sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class.

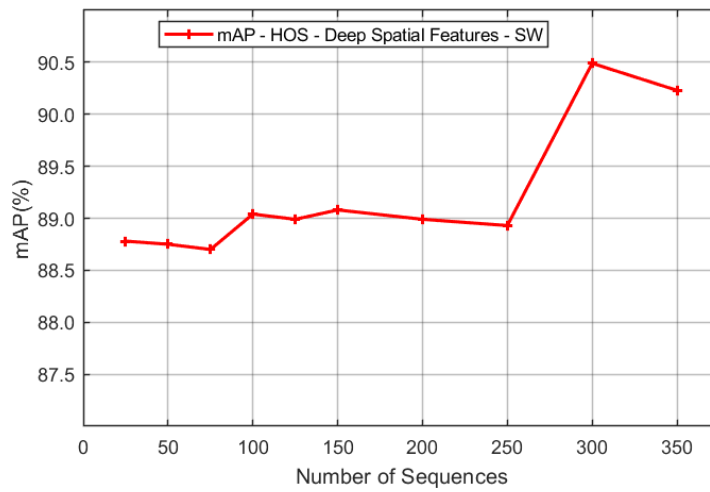


FIGURE 5.12: Deep Spatial Features Extracted using Sliding Windows - The effect of number of selected sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class.

be result of having frequent features those are not spatially related to the action itself. Having more features that are not extracted around the actions may lead to learning patterns that models background of the videos rather than the action.

When we compare the local features with the deep spatial features (Figure 5.10 vs Figure 5.6), we see that local features have better cross validation performance than the deep local features. The reason behind this difference may be the locality of the local features. When we look at spatial position of the local features detected, we see that most of the interest points are around the actions.

5.4.3.3 Parameter Selection for Deep Spatial Features Extracted from Multiple Layers of Network

In addition to the deep spatial features extracted via sliding window, we have tested the performance of the deep spatial features when we extracted multiple descriptors at each frame. Section 4.10.1 describes how the deep features are extracted from multiple layers. Mining with the features extracted from multiple layers of network, resulted hundreds of candidate features with the support values greater than 90%. And we skip the analysis for the effect of the number of candidate patterns on the performance. The average number of candidate patterns for each class at each level was 450.

The Figure 5.13 shows the effect of the number of the class specific selected patterns on the performance when the deep spatial features are extracted from multiple layers of the network. The selection is performed according to Section 4.7 by using the 450 candidate sequential patterns. The best performance is achieved when the number of selected patterns is 125 which means that the final descriptor length is $125 * 5 = 625$ (where 5 is the number of classes) for each level.

When we look at the performance difference of using deep features from multiple layers (Figure 5.10 vs Figure 5.13), we see that using the deep features from multiple layers has a negative effect on the cross validation performance. When we compare the local features with the deep spatial features (Figure 5.10 vs Figure 5.6), we see that local features have better cross validation performance than the deep local features. The reason behind this difference may be the locality of the local features. When we look at the spatial position of the local features detected, we see that most of the interest points are around the actions.

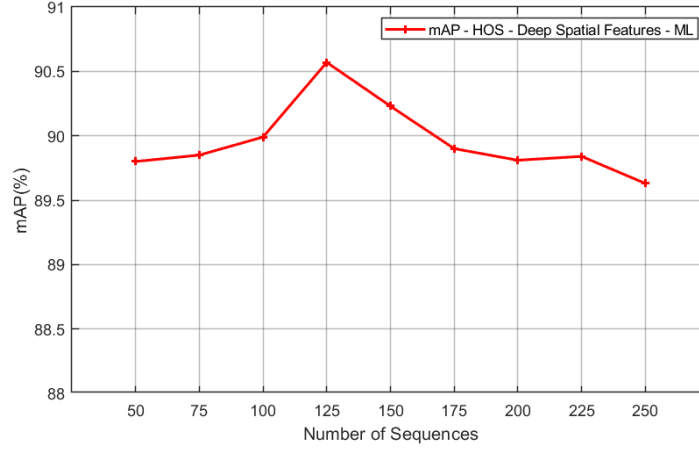


FIGURE 5.13: Deep Spatial Features Extracted from Multiple Layers of Network - The effect of number of selected sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class.

5.4.3.4 Parameter Selection for Deep Temporal Features

The Figure 5.14 shows the effect of the number of the class specific candidate patterns on the performance when the deep temporal features are used. The performance is provided with no selection in the features.

The best performance is achieved when the number of candidate patterns is 500 which means that the initial descriptor length is $500 * 5 = 2500$ (where 5 is the number of classes) for each level.

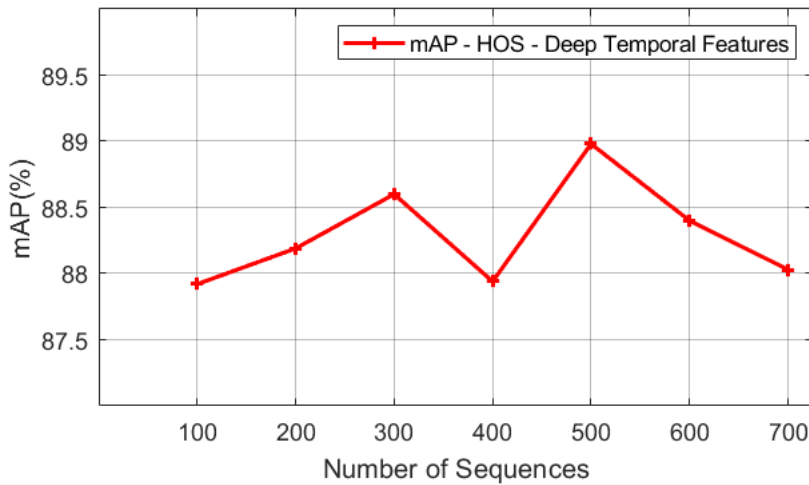


FIGURE 5.14: Deep Temporal Features - The effect of number of candidate sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class.

The Figure 5.15 shows the effect of the number of the class specific selected patterns on the performance when the deep spatial features are used. The selection is performed according to Section 4.7 by using the 500 candidate sequential patterns. The best performance is achieved when the number of selected patterns is 75 which means that the final descriptor length is $75 * 5 = 375$ (where 5 is the number of classes) for each level.

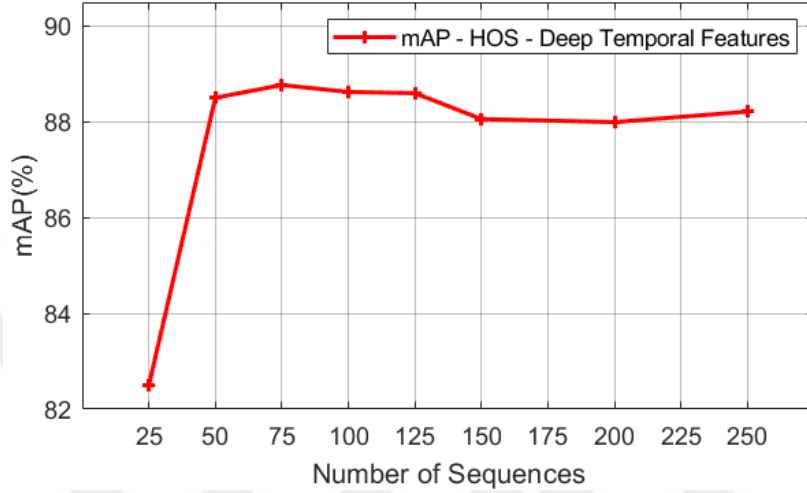


FIGURE 5.15: Deep Temporal Features - The effect of number of selected sequential patterns for TV Human Interactions dataset at each level of the pyramid for each class.

5.4.3.5 Performance of the HoS Method with Deep Features on Test Set

The performance of the HoS method with the deep features are presented in Tables 5.5 and 5.6. The best performing deep feature is the standard spatial features. When the features are extracted using a sliding window, the performance decreases as we see similar result at the parameter selection step in Section 5.4.3.2. This difference may be the result of having frequent features those are not spatially related to the action itself. Having more features that are not extracted around the actions may lead to learning patterns that models background of the videos rather than the action. When the features are extracted from the different layers of the network, the performance is better than the features are extracted using a sliding window, but worse than the pure deep spatial features.

The deep temporal features have worse performance than the deep spatial features and again we see similar results at the parameter selection step in Section 5.4.3.4. This may be the result of having up to 10 frames as a single input to temporal network and losing low-level action specific data.

Method	Annotation	mAP
HoS (SoftClust + TempPyrd + Deep Spatial)	Not Required	%53.7
HoS (SoftClust + TempPyrd + Deep Spatial SW)	Not Required	%46.2
Multi-Layer HoS	Not Required	%49.2
HoS (SoftClust + TempPyrd + Deep Temporal)	Not Required	%47.8
HoS (SoftClust + TempPyrd + Deep Temporal + Deep Spatial)	Not Required	%49.2
HoS (SoftClust + TempPyrd + Deep Temporal + Deep Spatial SW)	Not Required	%46.7
Multi-Layer HoS (+ Deep Temporal)	Not Required	%51.2
HoS (SoftClust + TempPyrd)	Not Required	%57.4

TABLE 5.5: Performance of the HoS method using the deep features on the TV Human Interactions dataset when the negative videos are included in the retrieval task. "SW" refers sliding window. "Multi-Layer HoS" refers the method features from multiple layers of deep network is used with Soft Clustering and Temporal Pyramid structure.

The combination of the deep temporal features together with the deep spatial features has better performance than the deep temporal features, but worse performance than the deep spatial features.

Also, the combination of the deep temporal features together with the deep spatial features extracted using sliding windows, has worse performance than the deep temporal features. This behavior is different when the deep temporal features are combined with the Multi-Layer HoS. The performance is increased in comparison to deep temporal features in this setup.

When the negative videos are included in the retrieval task, the combination of Multi-Layer deep spatial features with deep temporal features outperforms the combination of deep temporal features with deep spatial features extracted using sliding windows.

We observe that the recognition performance of the deep features is lower, compared to using local interest points (%3.7 below when the negative videos are included and %9.4 below when the negative videos are excluded). This is mostly due to the non-local nature of the deep features, since they operate over the whole frame. When working with features extracted from the whole frame, the proposed HoS method couldn't take advantage of finding co-occurring local patterns in a single video frame or finding multiple local patterns within multiple frames.

The Figure 5.16 shows the recognition performance (mAP) of HoS method with the best deep features (which is deep spatial features) for each class. Similar to the results with local features (Figure 5.8), we observe that the best performing class is *Handshake* class. And

Method	Annotation	mAP
HoS (SoftClust + TempPyrd + Deep Spatial)	Not Required	%59.8
HoS (SoftClust + TempPyrd + Deep Spatial SW)	Not Required	%52.2
Multi-Layer HoS	Not Required	%57.8
HoS (SoftClust + TempPyrd + Deep Temporal)	Not Required	%53.9
HoS (SoftClust + TempPyrd + Deep Temporal + Deep Spatial)	Not Required	%57.3
HoS (SoftClust + TempPyrd + Deep Temporal + Deep Spatial SW)	Not Required	%52.4
Multi-Layer HoS (+ Deep Temporal)	Not Required	%56.4
HoS (SoftClust + TempPyrd)	Not Required	%69.2

TABLE 5.6: Performance of the HoS method using the deep features on the TV Human Interactions dataset when the negative videos are excluded. "SW" refers sliding window. "Multi-Layer HoS" refers the method features from multiple layers of deep network is used with Soft Clustering and Temporal Pyramid structure.

also, the deep spatial features are better than the local features for *Kiss* class. The *Kiss* class doesn't contain significant movement in the execution of this interaction and the deep features seem better in modeling spatial data than the local features.

5.4.4 Comparison of the Best HoS Method with the State of the Art

We compare our best method (which is HoS (SoftClust + TempPyrd)) using local features with the several methods as shown in Table 5.7 and Table 5.8.

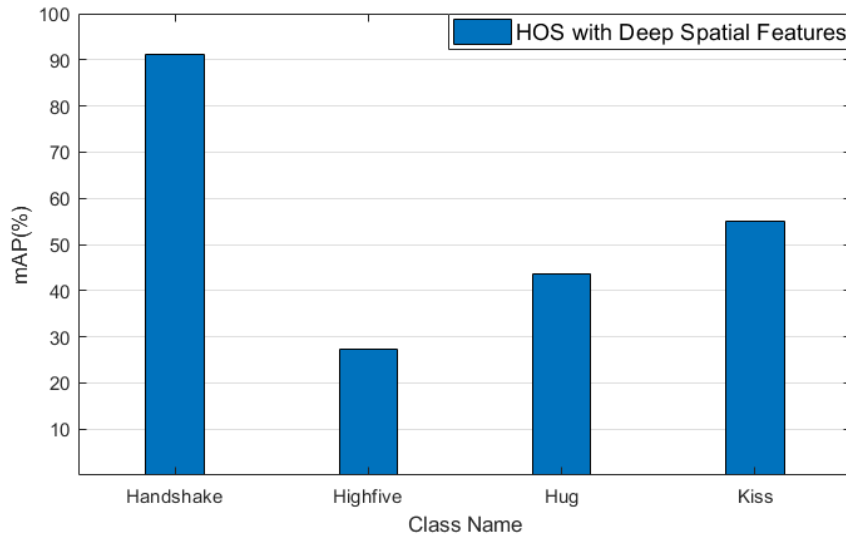


FIGURE 5.16: Classwise mAP performance of the HoS method with deep spatial features on the TV Human Interactions dataset

Method	Annotation	mAP
Wang et al. [20]	Not Required	%53.4
Yu et al. [67]	Not Required	%56.0
HoS (SoftClust + TempPyrd)	Not Required	%57.4
Marín-Jiménez et al. [11]	Automatic	%39.2
Patron-Perez et al. [21] (SL)	Automatic	%42.4
Patron-Perez et al. [21] (SL)	Manual	%54.8
Hoai et al. [22]	Manual	%56.3
Binlong et al. [66]	Manual	%68.0

TABLE 5.7: Performance comparison on the TV Human Interactions dataset when the negative videos are included in the retrieval task.

In the method of Patron-Perez et al. [21], head orientation features are included in the final descriptors, to enrich their representation. Marín-Jiménez et al. [11] apply standard BOW + STIP method and discard the STIPs outside the person region in the training phase by using person bounding boxes provided within the database.

We report the best results of Patron-Perez et al. [21] when including the negative videos as part of the retrieval task, which have settings using manual or automatic annotations and structured learning (SL). Hoai et al. [22] uses upper body annotations to train the upper body detector.

Similar to our experimental setting, Yu et al. [67] and Gaidon et al. [23] didn't use any annotations. As can be seen, in the presence of negative videos, our method outperforms other methods that do not use any annotations. When the negative videos are included, our method is %+1.4 better than the Yu et al. [67]. Note that, the method of Binlong et al. [66] has the best results in this dataset, but their method requires manual annotation.

When the negative videos are not included (Table 5.8) in the evaluation, our method is %+3.0 better than Yu et al. [67] and almost same as Khokher et al. [16]. Khokher et al.'s method [16] uses a short window video stabilization step to gain robustness against the camera motion. In our framework, we do not apply such a video stabilization and therefore, we can say that the features are affected by the camera motion. We believe that our framework could also make use of such an addition of a video stabilization step.

Method	Annotation	mAP
Gaidon et al. et al. [23]	Not Required	%62.4
Ma et al. [24]	Not Required	%64.4
Yu et al. [67]	Not Required	%66.2
Khokher et al. [16]	Not Required	%69.1
HoS (SoftClust + TempPyrd)	Not Required	%69.2

TABLE 5.8: Performance comparison on the TV Human Interactions dataset when the negative videos are excluded.

5.5 Results on UT Human Interactions Dataset

We also test our best method (HoS (SoftClust + TempPyrd)) on another benchmark human interactions dataset, UT Interactions. We compare our method with bag of words baseline and the state-of-the-art methods. The results are presented in Table 5.9. Burghouts and Schutte [65], Raptis and Sigal [29] and Ryoo [85] tested their method only on the set#1.

Method	Set #1 Acc	Set #2 Acc	AVG Acc
HoS + SoftClust + TempPyrd	%95.0	%95.0	%95.0
Y. Kong and Y. Fu [64]	%95.0	%88.0 ¹	%95.0
Burghouts and Schutte [65]	%93.3	-	%93.3
Raptis and Sigal [29]	%93.3	-	%93.3
Zhang et al. [13]	%95.0	%90.0	%92.5
Yu et al. [67]	%93.3	%91.7	%92.5
Liang J. et al. [25]	-	-	%92.3
Sefidgar at al. [30]	%93.3	%90.0	%91.7
Vahdat et al. [68]	%93.0	%90.0	%91.5
HoS + SoftClust	%88.3	%93.3	%90.8
Marín-Jiménez et al. [11]	%86.0	%88.0	%87.0
HoS	%88.3	%85.0	%86.7
Ryoo [85]	%85.0	-	%85.0
Patron-Perez et al. [21]	%84.0	%86.0	%85.0
Zhang et al. [63]	%76.0	%78.0	%77.0
BOW [13]	%75.0	-	%75.0
Slimani et al. [15]	%40.6	%66.7	%53.6

TABLE 5.9: Comparison of the proposed method with the state of the art methods on the UT Interaction dataset.

According to results provided in Table 5.9, our method is significantly better than the standard bag-of-words baseline [13] and it achieves the state-of-the-art performance. In this dataset, we observe that set#2 is more challenging than the set#1, because set#2 has camera

	Hshake	Hug	Kick	Point	Punch	Push
Hshake	19	0	0	1	0	0
Hug	2	18	0	0	0	0
Kick	0	0	20	0	0	0
Point	0	0	0	20	0	0
Punch	0	0	1	0	19	0
Push	0	0	1	0	1	18

TABLE 5.10: Confusion table of UT Interactions dataset for HoS representation is used with soft clustering and Temporal Pyramid.

jitters and moving background. The contribution of the soft clustering to HoS representation is higher in set#2. Our method with soft clustering and hierarchical temporal pyramid gives the best results in the set#2.

In set#1, the method of Zhang et al. [13], Kong and Fu [64] and our method have the same performance. Note that, in these results, Marín-Jiménez et al. [11] and Patron-Perez et al. [21] did not include the class *point* in their experiments. We should also note that the method of Kong and Fu [64] requires bounding boxes or a person detector, while our method does not require such additional annotations and/or auxiliary tools.

Table 5.10 shows the confusion matrix when HoS representation is used with soft-clustering and hierarchical temporal pyramid. Most of the confusion is between *hug* and *handshake* classes. In the UT Interactions dataset, the performance gain on accuracy of the hierarchical temporal pyramid and soft clustering is nearly +4%.

5.6 Time and Space Complexity of the Proposed Method

In terms of time complexity, the most complex step of our algorithm is the sequence mining step. The running time of the algorithm is mostly dependent on the number of patterns in the search space. When there are many similar frequent patterns in a sequence database, the number of the extracted sequential patterns may increase exponentially.

In our experiments, we have searched for the $min_{support}$ that generates required number of patterns by starting min_sup from the %99 and decreasing with a linear weight until the

¹On Set#2, Y. Kong and Y. Fu [64] did not report the numerical accuracy value, therefore we didn't include the accuracy on Set#2. However, in Figure 6(b) of [64], it seems that the accuracy is reported around 88% in the fully observed case for Set#2, whereas our method yields an accuracy of 95% in this set (Note that, this is a chart reading and it may differ by 1-2 points. Nonetheless, it is clearly observable that the accuracy is less than 90% from the chart.)

required number of patterns are found. Since the frequent sequential patterns are mined for each class and for each level of the pyramid, we run the multiple mining functions in parallel to take advantage of multi-core processing power. As an example, for the TV Human Interaction Dataset, there can be 35 parallel mining operations can be run in parallel since there are five classes and seven temporal levels. We have run the mining algorithm in parallel, just at the class level such that the patterns for each class is mined in parallel. We didn't run parallel miner for each level of the pyramid because of the maximum number of processing cores available in our test environment. With this method, the total running time of the mining algorithm with local features for UT Interactions was less than 25 minutes for each fold on a multi-core computer, and for TV Human Interactions it takes less than 10 minutes.

When the deep features are used, mining sequential patterns takes much less time since the number of features is less than the local features with the exception of using features extracted multiple levels of the network. The space complexity of the algorithm is similar to the time complexity, the mining algorithm uses less than 4Gb memory for both of the datasets. When working with features from multiple levels of the deep network, the mining duration has increased to almost 10 times more, find-out many sequential patterns with greater support values.

5.6.1 Runtime Environment

The exact specification of the running environment is shown below. Note that, this is the specification of the server in our computer vision laboratory. Most of the time, 2 or 3 cores and up to 16GBs of primary memory is used for the experiments in this thesis.

- Fedora 26 Linux operating system,
- Intel® Xeon® CPU E5-2640 v2, @ 2.00GHz, 6 cores, 20 MB cache memory,
- 46GB of primary memory,
- 2TB of secondary memory for storage of videos and intermediate files,
- Nvidia® Tesla® K40c graphics card having 12 GB of graphics memory and 2880 Cuda® cores.

5.6.2 List of References to Available Software Implementations Used in This Thesis

The following list provides references to the publicly available implementations of the algorithms used in this thesis:

- In Section 4.5, sequence mining algorithm implementation with time labels: <https://www.philippe-fournier-viger.com/spmf/HirateYamana.php>
- In Section 4.1.1, motion boundary histogram descriptor implementation: http://lear.inrialpes.fr/people/wang/download/improved_trajectory_release.tar.gz
- In Section 4.1, space-time interest point detector implementation: <https://www.di.ens.fr/~laptev/download.html#stip>

6 DISCUSSION

In this thesis, we propose a novel representation for human interaction recognition in videos. The idea is to build histogram of local feature sequences to model the relations between local spatio-temporal feature points. To this end, we incorporate sequence mining methods from data mining research. We show that extracting temporal relations between local feature points via frequent sequence mining and representing them via histogram of sequences framework is a successful method for the recognition of the human interactions. The evaluation on two benchmark datasets shows the effectiveness of the proposed approach.

By using sequence mining algorithms and sequence selection step, the experiments show that our final set of sequences not only frequent but also discriminative.

According to the experimental results, the hierarchical temporal pyramid structure and the soft clustering improves the recognition performance in all cases. With the help of the hierarchical temporal pyramid for enhancing the histograms, the sequential patterns cover the whole video at multiple temporal scales and this hierarchical structure makes the HoS more robust and comprehensive.

Our method can be used with different local feature representations as desired and the local features which are robust to camera movements may improve the performance. While being better or on par with the state-of-the-art methods for human interaction recognition, our framework has several advantages, like requiring less supervision and manual annotations, less auxiliary tools, offering an easily extendible framework to work with other types of features.

6.1 Future Work

This work provides a new representation that is capable of using different type of local information and there are different directions that can improve the performance of the method. Our work can be extended in the following directions:

- The most exciting one is the using different type of local features such as body parts, motion segments as an input to the histogram sequences representation. A multi-granular representation that combines sequence of local interest points, sequence of body parts and sequence of motion segments may have additional improvement on

the performance since it encodes both low level and high level information into a single representation. Use of multi-person pose features (such as [86]) together with the histogram of sequences may also be beneficial since the temporal pose models of the human actions tend to encode action specific information.

- The local feature descriptors are highly affected by the camera motion, either using video stabilization methods or using different type descriptors that are robust to camera motion is likely to improve the performance of the system.
- Another promising thing that is very likely to improve the performance is using dense local interest points together with the parallel sequence mining methods. With the advance of GPU technology, there is an open area to run a sequence mining algorithm in parallel. To the best of our knowledge, currently there are multiple methods [87–89] to run the sequence mining algorithms in parallel. It can discover many different interesting sequential patterns.

REFERENCES

- [1] J.K. Aggarwal and M.S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys*, **2011**.
- [2] Juan Carlos Niebles, Chih-Wei Chen, and Li Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *Proceedings of the 11th European Conference on Computer Vision: Part II, ECCV'10*. **2010**.
- [3] Michalis Vrigkas, Christophoros Nikou, and Ioannis Kakadiaris. A review of human activity recognition methods. *Frontiers in Robotics and AI*, 2, **2015**.
- [4] Daniel Weinland, Remi Ronfard, and Edmond Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision Image Understanding*, **2011**.
- [5] Limin Wang, Yu Qiao, and Xiaoou Tang. Mining motion atoms and phrases for complex action recognition. In *Computer Vision (ICCV), 2013 IEEE International Conference on*. **2013**.
- [6] Heng Wang and Cordelia Schmid. Action Recognition with Improved Trajectories. In *IEEE International Conference on Computer Vision (ICCV)*. **2013**.
- [7] Heng Wang, Muhammad Ullah, Alexander Kläser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *Proceedings of the British Machine Vision Conference*. **2009**.
- [8] Jaesik Choi, Ziyu Wang, Sang-Chul Lee, and Won J. Jeon. A spatio-temporal pyramid matching for video retrieval. *Computer Vision and Image Understanding*, **2013**.
- [9] Ivan Laptev. On space-time interest points. *International Journal of Computer Vision*, **2005**.
- [10] Marcin Marszałek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *Conference on Computer Vision & Pattern Recognition*. **2009**.

- [11] Manuel J. Marín-Jiménez, Enrique Yeguas, and Nicolás Pérez De La Blanca. Exploring stip-based models for recognizing human interactions in tv videos. *Pattern Recognition Letters*, **2013**.
- [12] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part II*, ECCV'06. **2006**.
- [13] Yimeng Zhang, Xiaoming Liu, Ming-Ching Chang, Weina Ge, and Tsuhan Chen. Spatio-temporal phrases for activity recognition. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part III*, ECCV'12. **2012**.
- [14] Silvio Savarese, Andrey DelPozo, Juan Carlos Niebles, and Li Fei-Fei. Spatial-temporal correlatons for unsupervised action classification. In *Proceedings of the 2008 IEEE Workshop on Motion and Video Computing*, WMVC '08. **2008**.
- [15] K. N. e. H. Slimani, Y. Benezeth, and F. Souami. Human interaction recognition based on the co-occurrence of visual words. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 461–466. **2014**.
- [16] Phung S. L. Khokher M. R., Bouzerdoum A. Human interaction recognition using low-rank matrix approximation and super descriptor tensor decomposition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1847–1851. **2017**.
- [17] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury. A "string of feature graphs" model for recognition of complex activities in natural videos. In *2011 International Conference on Computer Vision*, pages 2595–2602. **2011**.
- [18] M.S. Ryoo and J.K. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *Computer Vision, 2009 IEEE 12th International Conference on*. **2009**.

- [19] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004*. **2004**.
- [20] Heng Wang, Alexander Klaser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, **2011**.
- [21] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman. Structured learning of human interactions in tv shows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **2012**.
- [22] M. Hoai and A. Zisserman. Talking heads: Detecting humans and recognizing their interactions. In *IEEE Conference on Computer Vision and Pattern Recognition*. **2014**.
- [23] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Activity representation with motion hierarchies. *International Journal of Computer Vision*, **2014**.
- [24] Shugao Ma, Leonid Sigal, and Stan Sclaroff. Space-time tree ensemble for action recognition. *Computer Vision and Pattern Recognition (CVPR)*, **2015**.
- [25] Liang J., Xu C., Feng Z., and Ma Z. Affective interaction recognition using spatio-temporal features and context. *Computer Vision and Image Understanding*, **2016**.
- [26] Nazli Ikizler and David A. Forsyth. Searching for complex human activities with no visual examples. *International Journal of Computer Vision*, 80:337–357, **2008**.
- [27] Stefan Carlsson and Josephine Sullivan. Action recognition by shape matching to key frames. In *IEEE Computer Society Workshop on Models versus Exemplars in Computer Vision*. **2001**.

- [28] Li Liu, Ling Shao, and Peter Rockett. Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition. *Pattern Recognition*, **2013**.
- [29] Michalis Raptis and Leonid Sigal. Poselet key-framing: A model for human activity recognition. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13. **2013**.
- [30] Yasaman S. Sefidgar, Arash Vahdat, Stephen Se, and Greg Mori. Discriminative key-component models for interaction detection and recognition. *Computer Vision and Image Understanding*, pages 16–30, **2015**.
- [31] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. C3D: generic features for video analysis. *CoRR*, **2014**.
- [32] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14. IEEE Computer Society, **2014**.
- [33] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. *Computer Vision and Pattern Recognition*, **2016**.
- [34] Qing Li, Zhaofan Qiu, Ting Yao, Tao Mei, Yong Rui, and Jiebo Luo. Action recognition by learning deep multi-granular spatio-temporal video representation. *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, **2016**.
- [35] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Proceedings of the 27th International Conference on Neural Information Processing Systems*, **2014**.

- [36] T.Brox and J.Malik. Object segmentation by long term analysis of point trajectories. In *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science. **2010**.
- [37] Yu Hirate and Hayato Yamana. Generalized sequential pattern mining with item intervals. *Journal of Computers*, 1(3), **2006**.
- [38] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *Transactions on Intelligent Systems and Technology*, **2011**.
- [39] Alonso Patron, Marcin Marszalek, Andrew Zisserman, and Ian Reid. High five: Recognising human interactions in tv shows. In *Proceedings of the British Machine Vision Conference*. **2010**.
- [40] M. S. Ryoo and J. K. Aggarwal. UT-Interaction Dataset, ICPR contest on Semantic Description of Human Activities (SDHA), **2010**.
- [41] Cavent A. and Ikizler-Cinbis N. Histograms of sequences: a novel representation for human interaction recognition. *IET Computer Vision*, 12:844–854(10), **2018**. ISSN 1751-9632.
- [42] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, pages 525–531 vol.1. **2001**.
- [43] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12*, pages 1097–1105. USA, **2012**.
- [44] Pierre Sermanet, David Eigen, Xiang Zhang, Michael Mathieu, Rob Fergus, and Yann Lecun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International Conference on Learning Representations (ICLR)*. **2013**.

- [45] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, pages 818–833. Springer International Publishing, **2014**.
- [46] Andrew G. Howard. Some improvements on deep convolutional neural network based image classification. *CoRR*, abs/1312.5402, **2013**.
- [47] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, pages 147–151. **1988**.
- [48] H. Lee, A. Battle, R. Raina, and A. Ng. Efficient sparse coding algorithms. In *Advances in Neural Information Processing Systems*. **2007**.
- [49] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, T. Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. **2010**.
- [50] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. **2005**.
- [51] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. **1992**.
- [52] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *IEEE Conference on Computer Vision and Pattern Recognition*, **2014**.
- [53] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **1998**.
- [54] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. **1986**.

- [55] Rakesh Agrawal and Ramakrishnan Srikant. Mining sequential patterns. In Philip S. Yu and Arbee L. P. Chen, editors, *Proceedings of the Eleventh International Conference on Data Engineering*. IEEE Computer Society, **1995**.
- [56] Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499. **1994**.
- [57] Jian Pei, Jiawei Han, B. Mortazavi-Asl, H. Pinto, Qiming Chen, U. Dayal, and Mei-Chun Hsu. Prefixspan,: mining sequential patterns efficiently by prefix-projected pattern growth. In *Proceedings 17th International Conference on Data Engineering*. **2001**.
- [58] Jiawei Han Jianyong Wang and Chun Li. Frequent closed sequence mining without candidate maintenance. *IEEE Transactions on Knowledge and Data Engineering*, **2007**.
- [59] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, **1998**.
- [60] Jiebo Luo Jingen Liu and Mubarak Shah. Recognizing realistic actions from videos ”in the wild”. *IEEE International Conference on Computer Vision and Pattern Recognition*, **2009**.
- [61] Meinard Müller. *Information Retrieval for Music and Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, **2007**. ISBN 3540740473.
- [62] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, **1993**.
- [63] B. Zhang, P. Rota, N. Conci, and F. G. B. De Natale. Human interaction recognition in the wild: Analyzing trajectory clustering from multiple-instance-learning perspective. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*. **2015**.

- [64] Y. Kong and Y. Fu. Max-margin action prediction machine. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **2016**.
- [65] G.J. Burghouts and K. Schutte. Spatio-temporal layout of human actions for improved bag-of-words action detection. *Pattern Recognition Letters*, **2013**. Smart Approaches for Human Action Recognition.
- [66] Binlong Li, M. Ayazoglu, T. Mao, O. I. Camps, and M. Sznai. Activity recognition using dynamic subspace angles. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, **2011**.
- [67] Gang Yu, Junsong Yuan, and Zicheng Liu. Propagative hough voting for human activity recognition. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part III, ECCV'12*. **2012**.
- [68] Arash Vahdat, Bo Gao, Mani Ranjbar, and Greg Mori. A discriminative key pose sequence model for recognizing human interactions. In *International Conference on Computer Vision Workshops*. **2011**.
- [69] Nazli Ikizler-Cinbis Shugao Ma, Jianming Zhang and Stan Sclaroff. Action recognition and localization by hierarchical space-time segments. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*. **2013**.
- [70] Mikel D. Rodriguez, Javed Ahmed, and Mubarak Shah. Action mach: a spatio-temporal maximum average correlation height filter for action recognition. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*. **2008**.
- [71] Yao Li, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. Mining mid-level visual patterns with deep cnn activations. *International Journal of Computer Vision*, **2017**.
- [72] Vito Monteleone, Liliana Lo Presti, and Marco La Cascia. Hop: Histogram of patterns for human action representation. In *International Conference on Image Analysis and Processing*. **2017**.

- [73] Nizar R. Mabroukeh and C. I. Ezeife. A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys*, **2010**.
- [74] Ramakrishnan Srikant and Rakesh Agrawal. Mining sequential patterns: Generalizations and performance improvements. In *Advances in Database Technology*. **1996**.
- [75] Jay Ayres, Jason Flannick, Johannes Gehrke, and Tomi Yiu. Sequential pattern mining using a bitmap representation. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. **2002**.
- [76] Zhenglu Yang, Yitong Wang, and Masaru Kitsuregawa. Lapin: Effective sequential pattern mining algorithms by last position induction for dense databases. In *Advances in Databases: Concepts, Systems and Applications*. **2007**.
- [77] Mohammed J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine Learning*, **2001**.
- [78] C. I. Ezeife, Yi Lu, and Yi Liu. Plwap sequential mining: Open source code. In *Proceedings of the 1st International Workshop on Open Source Data Mining: Frequent Pattern Mining Implementations*. **2005**.
- [79] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense point trajectories by gpu-accelerated large displacement optical flow. In *Computer Vision – ECCV 2010*, pages 438–451. Springer Berlin Heidelberg, Berlin, Heidelberg, **2010**.
- [80] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Conference on Computer Vision & Pattern Recognition*. **2008**.
- [81] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, **2012**.

- [82] Gunnar Farneback. Two-frame motion estimation based on polynomial expansion. In *Image Analysis*, pages 363–370. **2003**.
- [83] Jose M. Chaquet, Enrique J. Carmona, and Antonio Fernández-Caballero. A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding*, 117(6), **2013**.
- [84] Andrea Vedaldi and Brian Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 1469–1472. **2010**.
- [85] M. S. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Proceedings of the 2011 International Conference on Computer Vision*. **2011**.
- [86] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1302–1310. **2017**.
- [87] Dongjin Yu, Wei Wu, Suhang Zheng, and Zhixiang Zhu. Bide-based parallel mining of frequent closed sequences with mapreduce. In *Proceedings of the 12th International Conference on Algorithms and Architectures for Parallel Processing - Volume Part II*, pages 177–186. **2012**.
- [88] Ke Wang, Elaheh Sadredini, and Kevin Skadron. Sequential pattern mining with the micron automata processor. In *Proceedings of the ACM International Conference on Computing Frontiers*. **2016**.
- [89] Chun-Chieh Chen, Hong-Han Shuai, and Ming-Syan Chen. Distributed and scalable sequential pattern mining through stream processing. *Journal of Knowledge and Information Systems*, 53(2), **2017**.

CURRICULUM VITAE

Credentials

Name,Surname : Aytaç ÇAVENT
Place of Birth : ANKARA
Marital Status : Married
E-mail : aytaccavent@cs.hacettepe.edu.tr
Address : Computer Engineering Department of Hacettepe University
Beytepe, Ankara

Education

BSc. : Computer Engineering, Hacettepe University
Ankara, TURKEY, 2009
MSc. : -
PhD. : Computer Engineering, Hacettepe University
Ankara, TURKEY, 2018, (Integrated PhD.)

Foreign Languages

English

Work Experience

Avionics Software Engineer, ASELSAN, 2009-2015
Avionics Systems Engineer, ASELSAN, 2015-On Going

Areas of Experiences

Image Processing, Computer Vision, Data Mining, Avionics Systems Engineering

Projects and Budgets

-

Publications

"Histograms of Sequences: A novel representation for Human Interaction Recognition",

IET Computer Vision, November 2018, 12:844–854(10), ISSN 1751-9632,
Aytac CAVENT, Nazli IKIZLER-CINBIS

”Recognition of complex human activities by using sequential pyramid matching”,
(”Karmasik Insan Hareketlerinin Sirasal Piramit Eslesmesi ile Taninmasi”)(in Turkish),
Signal Processing and Communications Applications Conference (SIU), April 2012,
Aytac CAVENT, Nazli IKIZLER-CINBIS

Oral and Poster Presentations





HACETTEPE UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING
THESIS/DISSERTATION ORIGINALITY REPORT

HACETTEPE UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING
TO THE DEPARTMENT OF COMPUTER ENGINEERING

Date: 04/10/2018

Thesis Title / Topic: Recognition Of Human Interactions Using Histogram Of Sequences

According to the originality report obtained by ~~myself~~/my thesis advisor by using the *Turnitin* plagiarism detection software and by applying the filtering options stated below on 04/10/2018 for the total of 93 pages including the a) Title Page, b) Introduction, c) Main Chapters, d) Conclusion sections of my thesis entitled as above, the similarity index of my thesis is 11 %.

Filtering options applied:

1. Bibliography/Works Cited excluded
2. Quotes ~~excluded~~ / included
3. Match size up to 5 words excluded

I declare that I have carefully read Hacettepe University Graduate School of Science and Engineering Guidelines for Obtaining and Using Thesis Originality Reports; that according to the maximum similarity index values specified in the Guidelines, my thesis does not include any form of plagiarism; that in any future detection of possible infringement of the regulations I accept all legal responsibility; and that all the information I have provided is correct to the best of my knowledge.

I respectfully submit this for approval.

Date and Signature

04/10/2018

Name Surname: Aytaç ÇAVENT

Student No: N10163027

Department: Computer Engineering

Program: Computer Engineering

Status: ☐ Masters ☐ Ph.D. ☒ Integrated Ph.D.

ADVISOR APPROVAL

APPROVED.

Doç. Dr. Nazlı İKİZLER CİNBİŞ

(Title, Name Surname, Signature)