DISSERTATION

SUBMITTED
TO THE
COMBINED FACULTIES FOR THE NATURAL SCIENCES AND FOR MATHEMATICS
OF THE
RUPERTO-CAROLA UNIVERSITY OF HEIDELBERG, GERMANY
FOR THE DEGREE OF
DOCTOR OF NATURAL SCIENCES

Put forward by

M.Sc., Dipl.-Ing. Borislav Antić
Born in: Novi Sad, Serbia
Oral examination:

LATENT STRUCTURED MODELS FOR VIDEO UNDERSTANDING

Advisor: Prof. Dr. Björn Ommer

Abstract

The proliferation of videos in recent years has spurred a surge of interest in developing efficient techniques for automatic video interpretation. The thesis improves the understanding of videos by building structured models that use latent information to detect and recognize instances of actions or abnormalities in videos. The thesis also proposes efficient algorithms for inference in and learning of the proposed latent structured models that are appropriate for learning with weak supervision.

An important class of latent variable models is the multiple instance learning where the training labels are provided only for bags of instances, but not for instances themselves. As inference of latent instance labels is performed jointly with training of a classifier on the same data, multiple-instance learning is very susceptible to overfitting. To increase the robustness of popular methods for multiple instance learning, the thesis introduces a novel concept of superbags (ensemble of bags of bags) that allows for decoupling of classifier training and latent label inference steps.

In the thesis, a novel latent structured representation is proposed to discover instances of action classes in videos and jointly train an action classifier on them. Action class instances typically occupy only a part of the whole video that is not annotated in weakly labeled training videos. Therefore, multiple instance learning is proposed to find these latent action instances in training videos and jointly train the action classifier. The thesis proposes a sequential method to multiple instance learning to increase the robustness of the training.

For the interpretation of crowded scenes, it is important to detect all irregular objects or actions in a video. However, the abnormality detection is hindered by the fact that the training set does not contain any abnormal sample, thus it is necessary to find abnormalities in a test video without actually knowing what they are. To address this problem, the thesis proposes a probabilistic graphical model for video parsing that searches for latent object hypotheses to jointly explain all the foreground pixels, which are, at the same time, well matched to the normal training samples. By inferring all latent normal hypotheses in a video, the model indirectly finds abnormalities as those hypotheses that are not supported by normal samples but still need to be used to explain the foreground. Video parsing is applied sequentially on individual video frames, where hypotheses are jointly inferred by a local search in a graphical model. The thesis then proposes a spatio-temporal extension of the video parsing, where an efficient inference method based on convex optimization is developed to find abnormal/normal spatio-temporal hypotheses in the video.

Zusammenfassung

Die enorme Verbreitung von Videos innerhalb der letzten Jahre hat zu einem gesteigerten Interesse an der Entwicklung von effizienten Techniken für die Interpretation von Videos geführt. Die Hauptmotivation dieser Dissertation ist es, das Verständnis von Videos zu verbessern, indem strukturierte Modelle konstruiert werden, welche latente Informationen nutzen, um wichtige versteckte Aspekte der sichtbaren Welt zu kodieren. Diese Dissertation präsentiert Algorithmen für effiziente Inferenz in sowie das Erlernen der vorgeschlagenen Modelle, die anwendbar sind auf schwach überwachte Trainingszenarios.

Um die latenten strukturierten Modelle zu trainieren, inferieren wir latente Variablen und trainieren Instanzklassifikatoren gemeinsam. Die Trainingszielfunktion ist typischerweise sehr anfällig für Überanpassung, wenn das Modell nicht richtig initialisiert wurde. Eine wichtige Klasse von latent strukturierten Modellen mit einem großen Potential für die Anwendung in der Bilderkennung ist das Multiple Instance Learning, bei dem die Problemstruktur Multimengen von Instanzen beinhaltet, sogenannte Bags. Um die Robustheit der Lernmethoden des Multiple Instance Learning zu verbessern, wird die Problemstruktur um sogenannte Superbags erweitert. Dies erlaubt die Entkoppelung von Parameterschätzung und der Inferenz von latenter Information.

In dieser Dissertation werden neuartige latente strukturierte Modelle vorgestellt, um die semantische Interpretation eines Videos zu verbessern. Um ein Video verstehen zu können, ist es notwendig die Handlungen der Objekte in einer Szene zu kategorisieren. Typischerweise finden die Handlungen nur in einem Teil des Videos statt. Diese Information ist jedoch nicht verfügbar während des Trainings oder des Testvorgangs. Daher wird in dieser Dissertation ein strukturiertes Modell vorgestellt, welches Multiple Instance Learning nutzt, um den versteckten Handlugsteil des Videos zu erkennen. Um die Robustheit beim Trainieren der Modelle zu erhöhen, schlägt diese Dissertation einen sequentiellen Ansatz für das Multiple Instance Learning vor, der auf dem Video Trimming beruht.

Um überfüllte Szenen zu interpretieren, ist es besonders wichtig alle unregelmäßigen Objekte, beziehungsweise Handlungen im Video zu erkennen. Allerdings wird die Erkennung von Unregelmäßigkeiten durch die Tatsache beeinträchtigt, dass die Trainingsmenge keine entsprechenden Beispiele enthält. Daher ist es notwendig Unregelmäßigkeiten in einem Test Video zu finden ohne zu wissen was sie sind. Um dieses Problem zu addressieren, schlägt diese Dissertation ein probabilistisches Modell mit latenten Variablen für das Video Parsing vor, das nach Objekt-Hypothesen sucht,

um gemeinsam alle Vordergründ-pixel zu erklären, welche zur gleichen Zeit, gut zu den normalen Trainingsbeispielen passen. Bei der Inferenz der latent normalen Struktur eines Videos, erkennt das Model auf indirekter Weise Unregelmäßigkeiten als Hypothesen, die nicht durch normale Beispiele unterstützt werden, aber dennoch benutzt werden müssen, um den Vordergrund zu beschreiben. Video Parsing wird sequentiell auf einzelne Video Bilder angewandt. Dafür wird eine Lokal-Inferenze-Technik in einem graphischen Modell genutzt. Diese Dissertation stellt daraufhin eine umfangreiche Erweiterung des Video Parsing Modells in der räumlich-zeitlichen Domäne vor, wo neuartige effiziente Inferenzverfahren auf konvexer Relaxation basierend entwickelt werden, um die latenten räumlich-zeitlichen Hypothesen im Video zu erkennen.

Acknowledgements

I would like to express my deepest gratitude and appreciation to my advisor Prof. Dr. Björn Ommer. His priceless advice, insightful remarks, and creativity have guided me steadily throughout my entire PhD studies. I will always gladly remember our long engaging discussions and his persuasive spirit of enthusiasm which he passes on to his students.

I would also like to thank Prof. Dr. Christoph Schnörr for the support and an impeccable leadership and supervision of the whole research training group. Furthermore I gratefully acknowledge the financial support I received from the German Research Foundation (DFG) within the program "Spatio-/Temporal Graphical Models and Applications in Image Analysis", grant GRK 1653. I especially thank our assistant Evelyn for her great work and help with every administrative detail. Thanks also go to all my colleagues in the research training group, computer vision group and HCI for spending nice time together and sharing memorable moments.

Last but not least, I would like to thank my wife Dajana, my sturdy pillar of strength I could always rely on, my father Radosav and mother Dragica and the rest of my family for their love and encouragement. Without them I wouldn't be here where I am today.

Contents

Chapter 1

Introduction

1.1 Visual Understanding

The world is extraordinarily complex and to be able to survive in it, humans use sophisticated cognitive functions that help them to interpret the world and formulate their actions accordingly. Humans perceive the world by their senses that are physical measurements of the processes that exist in the world. By processing sensory signals in the brain, a meaningful representation of the world is created. Vision has a prominent role in human interpretation of the world. Psychological research has indicated that a large part of the brain known as visual cortex is responsible for processing visual signals that come from the eyes.

Computer vision aims at processing signals captured by digital cameras to extract the high-level knowledge that corresponds to the semantic interpretation of the visual world. Computer vision is one of the core disciplines of artificial intelligence. Many higher level artificial intelligence functions rely on the world interpretation provided by computer vision. To comprehend the visual scene, computer vision has to answer the following questions: who (recognition of objects in the scene), what (recognition of actions that objects perform), where (recognition of type of the scene), and how (recognition of attributes of the objects and actions).

Object recognition, which lies at the core of computer vision, has the goal to learn visual object categories and identify new object instances that belong to these categories. Although recognition also involves the problem of identifying a particular object instance in a scene, in this thesis we will discuss the broader problem of identifying object instances that belong to the same semantic class - e.g. pedestrians, cars, motorcycles etc. There are three standard formulations of the visual object recognition: image classification, object detection and semantic segmentation (see Fig. 1.1). In image classification, the goal is to decide whether an instance of a certain object category is present in an image. More informative description of a scene is provided by object detection, which aims at identifying all object instances of a certain class in the scene and simultaneously finds their locations usually given as bounding boxes around the object instances. Se-

(a) Image Classification (b) Object detection (c) Semantic segmentation

Figure 1.1: The illustration of three main object recognition tasks: image classification, object detection and semantic segmentation.

mantic segmentation provides a more insightful view into the structure of a scene than image classification. More than just tagging an image with a segmentation label as in image classification, semantic segmentation associates the whole segmentation mask to each semantic category in an image.

Object recognition was first applied in the late 1960s for optical character recognition in the post offices of the United States. Subsequent successful use of object recognition methods in industrial applications such as visual inspection of semiconductor components, paved the way for more advanced applications in the years that followed. Object recognition quickly spread to other domains, e.g. biometric recognition (such as finger-print recognition), visual surveillance (e.g. vehicle detection on the street), and medical imaging (e.g. detection of anatomical structures). Despite a great success of object recognition techniques in applications with controlled environment, the general problem of object recognition in unconstrained environments is still elusive, due to high intra-class variability, sensitivity to occlusion, variable illumination and background clutter.

In recent years there is a trend of creating many more video clips than images. For example, more than a billion users visit the YouTube web site each month, and more than 100 hours of video material is uploaded to YouTube every minute [web14]. Video captures not only the static aspects of the scene (appearance, color and texture), but also the motion which is important for understanding actions that objects perform in the scene. For example, an image of a person in Fig. 1.2 does not tell us if the person is standing up or sitting down. Only by watching the whole video clip we can see based on the motion of the person that he is standing up. With respect to the temporal structure, actions can be divided into atomic (such as getting in the car or handshaking), and repetitive (such as jogging or eating). Atomic actions consist only of a single phase of the action, whereas in repetitive actions there are several phases included in the video clip.

Recognition of atomic actions is also of interest for understanding of complex activities performed by one or more persons in the scene, because they can be decomposed into a sequence of atomic actions. For example, a triple jump activity is nothing but a sequence of three atomic actions: hop, step and jump. Collective activities of a group

Figure 1.2: Static images cannot answer to some elementary questions about the scene, for example weather the person in this image is sitting down or standing up?

of people can also be decomposed into atomic actions that persons perform in a coordinated way (such as waiting in a queue where people are all lining up). Action recognition is also important for the problem of crowd analysis, where a large number of individual human actions helps in understanding the meaning of a crowd video. As in the case of object recognition, action recognition is often handicapped by the large intra-class variation, severe occlusions, clutter, and change of viewpoint between training and test videos.

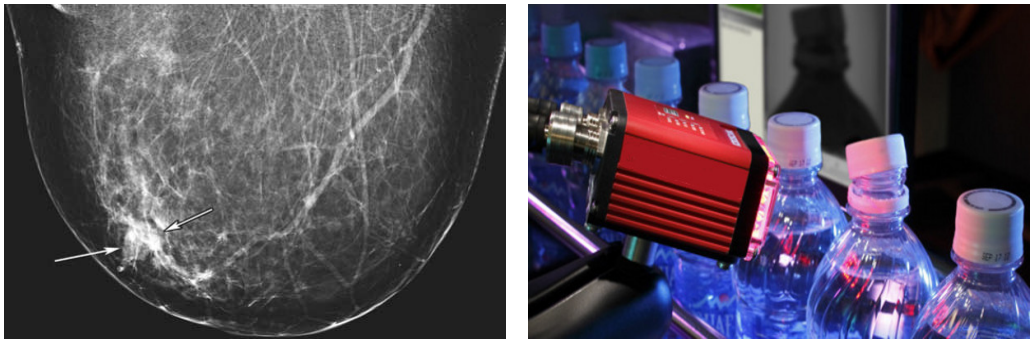
The recognition problem in video is often compound with the question: are all objects or events in a video regular or are there some abnormal instances? Abnormal instances play a prominent role in interpretation of a video, because abnormality detection is typically followed by certain measures that aim to minimize the consequences of the detected abnormality. In many fields we are faced with a need to detect abnormal instances, such as finding suspicious objects or unusual behavior in surveillance video, lumps in mammographic images, or defect products in a factory (Fig. 1.3). Therefore, it is important for a computer vision system to be able to find abnormal instances in a video.

1.2 Structured Models

The goal of computer vision, as we have already seen, is to automatically find the semantic interpretation of a scene from images or videos. The observed visual data are related to a higher-level information by the virtue of a visual model. In a mathematical sense, visual model is defined by specifying a relationship f that exists between input data x and output information y,

$$y = f(x). \tag{1.1}$$

For example, in image classification problem, the bag-of-features (BoF) model uses as input x a histogram of local features to represent image data, and output is the semantic



(a) Lump in mammographic image (b) defect product in factory

(c) irregular object in surveillance video (d) unusual behavior in surveillance video

Figure 1.3: Illustration of visual abnormality detection.

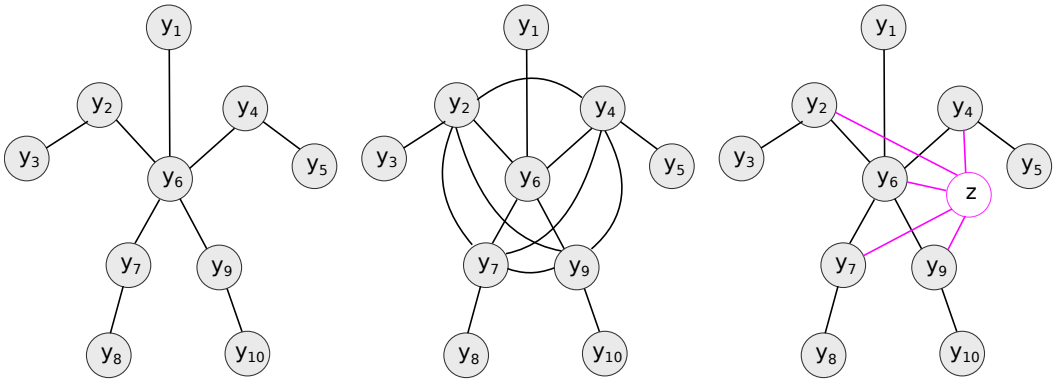
label y associated with an image. The model then specifies a relationship that exists between the input x and output y as discriminative classification rule. A detailed explanation of the BoF model is deferred to Sect. 3.2.

Structured models in general involve a large number of output variables with their relations and previously mentioned input-output relations. A structured model defines a mapping $f : \mathcal{X} \longrightarrow \mathcal{Y}$ between an input domain $\mathcal{X}$ and a structured output domain $\mathcal{Y}$. The mapping f is given in an implicit form, as a result of maximization of an evaluation function $g(x, y)$ for a given input $x \in \mathcal{X}$ over all possible structured output configurations $\mathcal{Y}$,

$$y = f(x) = \underset{y'}{\operatorname{argmax}}\, g(x, y'). \tag{1.2}$$

In probabilistic graphical models, where input x and output y are random variables, this corresponds to the criterion of maximum a posteriori probability (MAP), $g(x, y) = p(y|x)$. In structured SVM model, evaluation function is a linear function $g(x, y) = w^\top \Phi(x, y)$ of a feature vector $\Phi(x, y)$, that involves relations between input x and output y variables. The parameter vector w is estimated by learning on the training set $(x_1, y_1), \ldots, (x_n, y_n)$.

Structured models are very useful in computer vision. In image segmentation problem, the goal is to label each pixel in an image as foreground/background. The Markov random field (MRF) model takes the input image x, and predicts the labels $y \in \{0, 1\}^n$



(a) Tree-structured model (b) Higher-order cliques (c) Latent variable

Figure 1.4: Different structured models for the *pedestrian* class. (a) Tree structured model which can be applied using efficient inference algorithms. (b) Non-tree structured model that involves higher-order cliques that increase representational power of the model at the cost of making inference harder than in the tree model. (c) Introduction of a latent variable z makes the structure of the model simpler than in (b), however without sacrificing the representational power of the model in (b).

of all pixels jointly. It uses as evaluation function $g(x, y)$ a linear combination of pixel-wise data terms $g_i(x, y_i)$ and pair-wise interaction terms $g_{i,j}(y_i, y_j)$ that encourage pixels i and j to have the same label if they are neighbors ($\mathcal{N}$ denotes the set of all neighboring pairs of pixels),

$$y = f(x) = \underset{y' \in \{0,1\}^n}{\operatorname{argmax}} \sum_{i=1}^{n} g_i(x, y'_i) + \sum_{(i,j) \in \mathcal{N}} g_{i,j}(y'_i, y'_j). \tag{1.3}$$

Markov random field models are also useful in other image processing domains, such as image denoising [PPLA01], stereo estimation [YHMU12], and optical flow [RB07], to name a few.

Structured prediction has also been applied in higher-level computer vision tasks, such as object recognition. The main idea here is to represent an object as a set of local parts, whose appearance and spatial dependencies are described by a structured model. This idea dates back to the 1970s when Fischler and Elschlager [FE73] proposed pictorial structures for object detection in images. In year 2005, Felzenszwalb and Huttenlocher proposed a modification of the original pictorial structure scheme, where spatial dependencies between parts are defined on a tree graph, because there are efficient inference methods developed for tree graphs that use dynamic programming.

A tree is a minimal connected graph in terms of number of edges (Fig. 1.4(a)). In a tree with n nodes there are $n - 1$ edges, and one cannot find a connected graph with n nodes that uses less than $n-1$ edges. However, in case of articulated object recognition, this number of edges in the graph is not sufficient to represent all structural dependencies that exist between parts of an articulated object. Adding more edges to the graph increases the flexibility of the model, but this comes at the expense of losing computa-

tional tractability in inference due to emergence of large cliques (1.4(b)). Better results can be obtained by designing graphical models with latent variables that efficiently use the graph structure for computationally feasible inference (Fig. 1.4(c)). As the latent information is not provided in the training set, it needs to be inferred in the training. A recent successful example of latent modeling in computer vision is the Felzenszwalb's Deformable Part Model [FGMR10]. This model uses latent HOG parts that are spatially connected as a star graph that has in the center the provided bounding box as a reference frame. However, individual HOG parts are not provided, but they have to be inferred during training. Felzenszwalb et al. propose a discriminative training procedure called Latent SVM to infer the missing labeling information and find HOG-based classifiers. The score of data x for class y is found by maximizing a linear cost function overall all latent configurations z in the latent space $\mathcal{Z}$,

$$g(x, y) = \max_{z \in \mathcal{Z}} w_y^\top \Phi_y(x, z). \tag{1.4}$$

In a standard supervised learning, a classifier is trained on the set of training instances where each instance is given as a pair of input data x_i and output label y_i. The recognition phase then uses the classifier to predict labels y of new instances based on their input data x. However, there are applications in which instance labeling is too costly or simply unavailable. The Multiple-instance Learning (MIL) is a flexible learning scheme that has been proposed to deal with learning from ambiguous instance labels. MIL is based on the following paradigm: instead of providing labels for the training instances, labels are provided for the bags (i.e. sets) of instances. A bag is assigned a certain label if at least one of its instances is labeled with that label. If none of the instances in a bag has some label, then the label is not assigned to the bag. Therefore, labels of instances in bags can be considered as latent variables, because they are not given during training, but they have to be inferred. An illustration of the MIL problem is given in Fig. 1.5.

The MIL idea was proposed in the machine learning community by Dietterich et al. [DL97] who worked on the problem of drug discovery. The problem consisted of predicting properties of molecules based on their shape statistical features. As one molecule can take multiple distinct shape configurations, and it is unknown which shape configuration gave rise to the certain property of a molecule, MIL is used to represent a molecule as a bag of shape instances. Beside the prediction of drug activity, MIL has also been applied in other disciplines, such as computer vision, e.g. for image classification [CBW06], object tracking [BYB11], or object detection [MO12].

In a binary case, the MIL training set consists of bags $\{B_1, \ldots, B_n\}$ with their labels $\{Y_1, \ldots, Y_n\}$, $Y_i \in \{-1, +1\}$. Each bag is a set of instances $B_i = \{x_{i1}, \ldots, x_{im}\}$, $x_{ij} \in \mathcal{X}$. Typically, all instances are assumed to have a label that is not provided during training, but has to be inferred while the classifier is trained. For a positive bag, there is at least one positive instance in the bag, which is typically called a positive *witness*,

$$Y_i = \begin{cases} +1, & \text{if } \exists j \text{ s.t. } y_{ij} = +1. \\ -1, & \text{otherwise.} \end{cases} \tag{1.5}$$

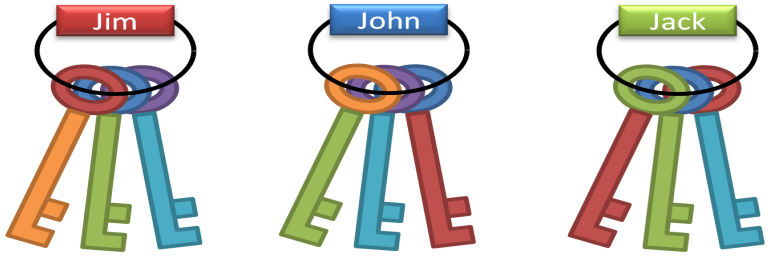


Figure 1.5: An illustration of the multiple-instance learning (MIL). Three persons are responsible for the maintenance of a building. Each person has a key ring with keys that unlock some of the doors in the building. Jim and John can unlock the emergency door, while Jack cannot do it. The goal is to predict which of the keys unlocks the emergency door. Jim's and John's key rings are, in MIL terminology, the positive bags and it is thus necessary to find which key appears on both of these key rings, and, at the same time, does not appear on the Jack's key ring (negative bag).

MIL's latent structure is defined by a relationship between latent instance labels y_{ij} and a bag label Y_i,

$$Y_i = \max_j y_{ij}. \tag{1.6}$$

There are two levels on which a classifier can be trained in MIL. An instance classifier assigns labels to individual instances based on their value, $f : \mathcal{X} \longrightarrow \mathcal{Y}$, while a bag classifier assigns a label to the whole bag, $F : \mathcal{X}^m \longrightarrow \mathcal{Y}$. However, because of the relationship that connects bag and instance labels, it is necessary to train only the instance classifier, as the bag label can be predicted from the instance predictions, $F(B_i) = \max_j f(x_{ij})$.

1.3 Objective

The main objective of this thesis is to improve video understanding by building structured models that represent latent action instances and abnormal/normal object instances in a visual scene. As the thesis uses a weakly supervised training setup, latent instances need to be inferred automatically from training videos. For all latent variable models proposed in this thesis, efficient inference and learning algorithms need to be developed.

Video understanding is a very broad topic that underpins many problems in computer vision. However, there are several tasks that are of crucial importance for building a working video understanding system, such as the object detection and classification, action recognition or abnormality detection. The goal of this thesis is to advance the state-of-the-art in the disciplines of action recognition and abnormality detection in videos. In action recognition, the goal is to enhance the recognition performance by inferring a part of a video that corresponds to the latent action instance. The goal is

to develop a robust learning procedure that can find the latent action parts in training videos and to simultaneously train the action classifier on them.

The thesis also aims at improving abnormality detection, which has a prominent role in explaining the content of a video. The goal of the thesis is to avoid independent detections of abnormalities in individual image patches or regions, but to build a structural model that uses latent object hypotheses that jointly explain the foreground information in a video. Besides sequentially applying the model on successive video frames, the thesis also investigates extensions to the spatio-temporal domain where more intricate interactions between object hypotheses are possible thus allowing for a better discerning of abnormal and normal object instances.

With respect to inference and learning, the thesis aims at improving multiple-instance learning as a structured representation that deals with ambiguous labeling in the training set. The thesis investigates the process of simultaneous inference of the latent instance labels and training the instance classifier to make it more robust by introducing additional structured concepts in the model. Additionally, efficient techniques for inference and learning in the proposed graphical models for abnormality detection are sought after in the thesis.

1.4 Current Challenges in Video-based Recognition

Computer vision continues to be one of the hardest areas of artificial intelligence, due to the complexity and fuzziness of the visual world that needs to be transformed into a semantic representation. Of crucial importance is thus to model the structural properties of the visual world, and to devise efficient algorithms for learning and inference in such models. We state here some of the most important challenges that are met when building a working system for video interpretation.

Intra-class variability in the action class: The set of all instances of an action that participates in the action class is very diverse. People perform the same action in many different ways. For example, a person has a characteristic walking style that can be used for identification of that person. Beside differences in motion, appearance is also very diverse among instances of an action class. For example, a running person can be a businessman hurrying up for the next business meeting, or it can be a contestant in an athletic race. The length of action can also vary a lot in different action instances.

Variety of camera conditions: Camera introduces many degrees of freedom in video understanding process. The position of the camera with respect to an actor induces view-point changes, that hamper action learning, because there are typically not enough training samples for all possible camera views. Another artifact is the camera motion, that introduces a motion bias. Also the low video resolution or compression artifacts can be a significant hurdle in action recognition or abnormality detection.

Video editing artifacts: We also might not see the complete action because video has been edited by cropping or cutting. For example, we might see only the face of a person sitting down, but not the whole vertical motion of the body, in which case it is really

hard to recognize the action. Another problem is the lack of continuity in recorded video of an action. For example, a video of a person getting our of car first shows the person sitting in the car, and then the person already standing nearby the car so the intermediate moments that capture the action are not recorded.

Occlusion: Sometimes the action is not fully visible because parts of the actions are occluded by other objects in the scene. In case of actions that consist of coordinated motion of two actors, one might not see both actors all the time, but only one of them because the other one is occluded (such as in hugging action). Human motion is also susceptible to self-occlusion, when body parts occlude each other in the course of action (e.g. running).

Clutter: Video understanding process is hampered if the action instances are buried amid less relevant ingredients of a video. The presence of clutter in video can severely degrade the recognition accuracy because it introduces undesirable features into video representation that mask the relevant action features. The clutter in video can exist in both spatial and temporal domain.

1.5 Contributions of the Thesis

This thesis makes the following contributions:

- Action recognition and abnormality detection in videos are improved by introducing novel latent structured models, that represent unknown action instances or abnormal/normal object instances in weakly labeled videos as latent hypotheses that are jointly inferred within the model.
- The robustness of multiple-instance learning (MIL) is increased by introducing the concept of superbags (ensemble of bags of bags). To avoid overfitting that occurs because the classifier is trained on the same data that are used for inference, superbags decouple the processes of latent instance label inference and classifier training by performing them on separate superbags. The variance of instance labels that are inferred by multiple-instance learning is reduced by averaging label predictions cast by multiple superbag classifiers. The concept of superbags is easily integrated with a number of popular MIL methods, that yields an improved performance on MIL benchmark datasets and in various computer vision applications.
- The method for discovery of latent action class instances (action subsequences) in weakly labeled videos is developed in the thesis. Classification of video into one of the action categories is improved by combining the predictions of the full sequence classifier and action subsequence classifier. Action subsequence classifier can also be used as an action detector.
- Sequential multiple-instance learning is proposed to train the action subsequence classifier on weakly annotated videos. The proposed MIL formulation is used to

jointly find latent action subsequences in training videos and train the action subsequence classifier on them. The proposed training procedure increases the robustness of the action classifier training. A larger number of action subsequence hypotheses are generated in each video to increase the recall rate of discovered action instances and their number and length are gradually decreased. The proposed concept is successfully evaluated on the state-of-the-art dataset for action recognition and showed good results.

- A method for sequential parsing of individual video frames is proposed to detect abnormalities in videos. The parsing model is capable of discovering abnormalities in test videos, although the model has not seen any abnormal sample in training videos. The method avoids a direct search for individual abnormal image patches or regions.

- The sequential video parsing is designed as a Bayesian network where latent object hypotheses are mutually competing for explanation of the foreground pixels in video, and, simultaneously, hypotheses are explained by the normal object exemplars from the training videos. Hypotheses are jointly selected based on the statistical inference technique of explaining away, where only hypotheses that are indispensable for explaining of foreground pixels are retained. Abnormalities are discovered indirectly as those hypotheses which are needed for covering the foreground but which cannot be explained by normal samples. The method achieves state-of-the-art performance both in abnormality detection and localization, as is evidenced by testing on the most challenging benchmark set for abnormality detection.

- Spatio-temporal video parsing is proposed to jointly parse frames in a video, as opposed to sequential video parsing that processed frames individually. Spatio-temporal video parsing allows to resolve both spatial and temporal dependencies between objects in a scene. The inference process aggregates object evidences from different video frames and decides about objects hypotheses in the scene that leads to improved state-of-the-art results in abnormality detection and localization in videos.

- The inference in spatio-temporal video parsing is posed as a convex optimization problem. Consequently, the inference for video parsing does not depend on the model initialization and the global optimum can be efficiently found by projected gradient method.

- Video parsing uses spatio-temporal normal shape prototypes, thus abnormalities can also be inferred by observing how the shape evolves in time. The set of normal spatio-temporal shape prototypes is learned by alternating between the parsing of training videos and estimation of the shape prototypes using the results of parsing. Both the video parsing and estimation of shape prototypes can be formulated as convex optimization problems, and thus can be efficiently solved.

1.6 Organization of the Thesis

Chapter 2 first reviews the general problem of multiple-instance learning (MIL), that has seen many applications in computer vision. Since the MIL problem is prone to initialization error and overfitting, we explain our solution that introduces suberbags (ensembles of bags of bags) as a new concept in MIL. Using superbags, the tasks of label inference and classifier training can be decoupled that leads to a more robust training. We then show how to integrate superbags with some of the most prominent methods for MIL. We evaluate the proposed concept on several standard benchmark datasets for MIL, including also two computer vision datasets and demonstrate the benefit of introducing superbag concept to MIL.

Chapter 3 deals with the problem of action recognition in video. We first review prominent approaches to classification of actions in video. We than explain the proposed approach that uses MIL for the discovery of latent action instances in videos and the joint training of the action classifier. Sequential MIL method is proposed for greater robustness of action classifier training. Evaluation is performed on the most demanding dataset for action recognition, where the proposed sequential MIL approach showed good results.

Chapter 4 proposes a novel approach to abnormality detection by sequential video parsing. It first discusses the main aspects of the previous work on abnormality detection in videos, which aim to directly test individual image patches independently of another. After reviewing background subtraction and probabilistic graphical models that our video parsing method builds upon, the method for sequential video parsing is presented. First, the procedure for finding a shortlist of object hypotheses is described (initialization stage), and then the graphical model for sequential video parsing is explained. After that, details about inference in a graphical model for video parsing are given. Finally, results of video parsing on the state-of-the-result dataset for abnormality detection are shown.

Chapter 5 describes an improved model for video parsing, where the whole spatio-temporal volume is parsed jointly, as opposed to Chapter 4 that parses each frame separately. The method for building a shortlist of spatio-temporal hypotheses is explained, and then the probabilistic graphical model for spatio-temporal parsing. Inference is posed as a convex optimization problem which is explained in a separate section. Learning of normal spatio-temporal prototypes for full video parsing is described afterwards. Finally, extensive results on state-of-the-art datasets for video parsing are presented in the experimental section. Results are extensively compared to all recent state-of-the-art methods for abnormality detection (including the sequential video parsing), both concerning the abnormality detection and pixel-wise localization.

Chapter 6 concludes the thesis with the final discussion.

Chapter 2

Robust Multiple-Instance Learning using Superbags

For automatic interpretation of a video, it is necessary to find semantic categories that the video belongs to. Visual scene can be decomposed into many instances, but not all of them belong to the semantic category of interest. Thus, the task of video understanding is not only to predict semantic category for the whole video, but also to find instances in the video that represent a certain semantic category. However, in a weakly supervised setting that prevails in practice, there is no information about instances of training videos that represent semantic classes that training videos belong to. Therefore, semantic instance labels in training videos are considered as latent variables. Training a classifier on a dataset with latent instance labels can be conveniently formulated as multiple-instance learning (MIL). In contrast to standard supervised learning where all training instances are provided with their class labels, MIL discovers latent instance labels during training of the classifier. MIL setup comes with training instances grouped into bags, with labels provided only at the bag level. So far, instance labels have been inferred on the same data that the classifier is trained on, which leads to overfitting. To resolve this critical issue of multiple-instance learning we introduce an ensemble of bags of bags, i.e., superbags. This concept decouples classifier training and label inference which are performed now on different superbags.

2.1 Outline of the Approach

The framework of MIL has been applied in many practical applications because it provides a powerful mechanism for dealing with latent instance labels that appear in weakly supervised setups. After initially applying MIL to the drug activity prediction [DL97], the concept has quickly spread to many other disciplines such as text-categorization [ATH03] and computer vision. Many authors used MIL for image retrieval [ZG01, VG08], image categorization [CBW06] or object detection [MO12]. Object tracking has also had a great benefit from the MIL setup [BYB11, LSB10], which

manages to seamlessly pick among many candidate patches one that best corresponds to the tracked object and uses it for updating the appearance model.

To train a classifier from instances with unknown labels, most methods for MIL use an alternating optimization approach which iteratively trains the classifier and finds the latent instance labels. However, these two steps are performed on the same data, which renders learning of classifier prone to overfitting and it also increases the variance of inferred labels.

So how can we resolve these issues and increase the robustness of MIL? First, we avoid predicting the same instance labels that the classifier is trained upon. Second, we decrease the uncertainty of inferred instance labels by averaging multiple predictions from separate classifiers. These goals can be achieved by establishing an ensemble of bags of bags, that is referred to as *superbags*. Classifier training and label inference are decoupled by training a classifier on a superbag, predicting instance labels from other superbags, and averaging all these predictions. The superbag concept is easy to integrate with existing MIL approaches, and its results show a consistent gain over baseline methods that use the same data for training the classifier and predicting the labels.

2.2 Max-Margin Learning

The most popular MIL approaches are based on the max-margin discriminative learning concept. Therefore, we give here a brief overview of underlying concepts of the max-margin learning. The concept of max-margin learning is based on a solid theoretical ground known as *statistical learning* theory [Vap95]. In the context of max-margin learning, we give the basic exposition of the support vector machine (SVM) classifier, that and show how it can be generalized with *kernel* functions.

Support Vector Machine

Suppose that we want to train a classifier on the training samples $(\mathbf{x}_1, y_1)$, ..., $(\mathbf{x}_n, y_n)$, where $\mathbf{x}_i \in \mathcal{X}$ is the feature vector of training sample i, and $y_i \in \{-1, +1\}$ is a binary class label of the same training sample. Feature vectors reside in a high-dimensional feature space $\mathcal{X}$. Suppose that class samples are linearly separable in the feature space, so that there are infinitely many hyperplanes that can separate the samples of the two classes (Fig. 2.1). An important question in machine learning is whether some hyperplanes generalize better on the test set than others? Vapnik and Chernovenkis [Vap95] answered this question in their statistical learning theory. According to them, the best generalization properties has a hyperplane with the largest distance between the closest training samples of both classes and the hyperplane that is denoted as *margin* m. The hyperplane is defined in a feature space $\mathcal{X}$ by a linear equation $\mathbf{w}^\top \mathbf{x} + b = 0$. The training samples of both classes that are closest to the hyperplane satisfy the equation $\mathbf{w}^\top \mathbf{x} + b = \pm 1$. The distance between these hyperplanes corresponds to the margin m, that can be calculated as $m = \frac{2}{\|\mathbf{w}\|}$. Training samples with label $y_i = +1$ are correctly

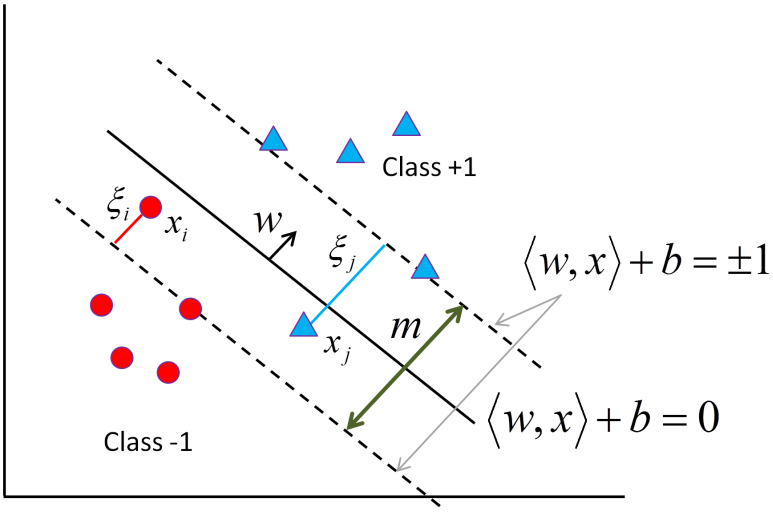


Figure 2.1: Maximum margin classifier finds the separating hyperplane $\mathbf{w}$ that maximizes the distance between the closest samples of both classes and the hyperplane.

classified if they satisfy $\mathbf{w}^\top \mathbf{x}_i + b \geq 1$. For training samples with label $y_i = -1$, the correct classification occurs when $\mathbf{w}^\top \mathbf{x}_i + b \leq -1$. Therefore, training samples of both classes are correctly classified if they satisfy

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \forall i. \tag{2.1}$$

Now we can formulate the optimization problem whose solution is the max-margin classifier,

$$\begin{aligned} &\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \forall i. \end{aligned} \tag{2.2}$$

This is a constrained convex optimization problem in which we try to minimize a quadratic function subject to a set of linear constraints. The constrained optimization problem can be turned into an unconstrained problem using the method of Lagrangian multipliers. The optimization problem from Eq. 2.2 becomes

$$\mathcal{L} = \frac{1}{2}\mathbf{w}^\top \mathbf{w} + \sum_{i=1}^{n} \alpha_i (1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b)), \tag{2.3}$$

with nonnegative Lagrangian multipliers $\alpha_i \geq 0, \forall i$. To eliminate variables $\mathbf{w}$ and b, we set the gradient of the Lagrangian with respect to these variables to zero,

$$\nabla_{\mathbf{w}} \mathcal{L} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \tag{2.4}$$

$$\nabla_b \mathcal{L} = \sum_{i=1}^{n} \alpha_i y_i = 0. \tag{2.5}$$

If we substitute analytical expression for $\mathbf{w}$ into the Lagrangian, after some algebraic transformation, we obtain a dual form of $\mathcal{L}$ whose global optimum can be found by the quadratic programming (QP),

$$\max_{\alpha} \mathcal{L}(\alpha) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j + \sum_{i=1}^{n}\alpha_i \tag{2.6}$$

$$\text{s.t. } \alpha_i \geq 0, \sum_{i=1}^{n}\alpha_i y_i = 0.$$

Once the dual variables α_i are found, the hyperplane $\mathbf{w}$ of the linear classifier is found as a linear combination of the training samples $\mathbf{x}_i$,

$$\mathbf{w} = \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i. \tag{2.7}$$

Dual variables α_i that are equal to zero do not contribute to Eq. 2.7, so the hyperplane $\mathbf{w}$ can then be represented as a linear combination of training samples with nonzero dual value, that are known as *support vectors* (SV). Therefore, SVM classifier allows for a sparse representation of the decision boundary $\mathbf{w}$ determined only by support vectors. Karush-Kuhn-Tucker (KKT) conditions require that

$$\alpha_i \geq 0 \tag{2.8}$$

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 \geq 0 \tag{2.9}$$

$$\alpha_i(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1) = 0 \tag{2.10}$$

The last condition, known as complementary slackness, implies that if the dual variable is strictly positive, $\alpha_i > 0$, then the training sample lies exactly on the maximum margin hyperplane, $y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$. This explains why such a training sample is denoted as support vector.

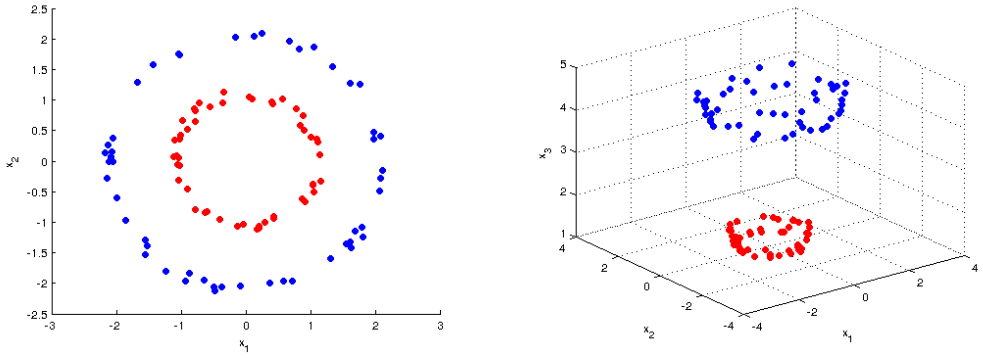
A new test data $\mathbf{x}$ is classified by evaluating the sign of SVM decision function

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b = \sum_{i=1}^{n}\alpha_i y_i(\mathbf{x}_i^\top \mathbf{x}) + b. \tag{2.11}$$

We note that in order to compute SVM score $f(\mathbf{x})$ it is not necessary to know the value of hyperplane normal vector $\mathbf{w}$, but the score can be calculated from scalar products $\mathbf{x}_i^\top \mathbf{x}$ of test instance $\mathbf{x}$ with all support vectors $\mathbf{x}_i$. This idea is used in the development of kernel methods.

Soft Margin

The condition that training samples need to be linearly separable is sometimes not satisfied by the training data. The classification rule, $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$, can be violated by some training samples. The nonnegative slack variable, $\xi_i \geq 0$, is used to relax the classification constraint, $y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i$. If slack equals zero, $\xi_i = 0$, the margin



(a) Before kernel mapping (b) After kernel mapping

Figure 2.2: Illustration of the kernel mapping. In this case, the mapping is done explicitely by the function $\phi : (\mathbf{x}_1, \mathbf{x}_2) \longrightarrow (\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_1^2 + \mathbf{x}_2^2)$. Before kernel mapping data points are not linearly separable. After kernel mapping data points become linearly separable.

is not violated. For slacks $0 < \xi_i < 1$, the margin violation still does not result in the wrong classification. The miss-classification occurs only when slack is greater than one, $\xi_i > 1$. Thus the sum of slack variables $\sum_{i=1}^{n} \xi_i$ can be used as an upper bound of the number of classification errors. Consequently, the classifier training objective function aims to maximize the margin and at the same time to minimize the number of miss-classifications. This can be expressed as follows

$$\min_{\mathbf{w},b,\xi_i} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \tag{2.12}$$

$$\text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i. \tag{2.13}$$

The parameter C controls the trade-off between margin maximization and training error minimization. It can be shown ([Bis06]) that the dual problem of the soft margin case is similar to that of the hard margin,

$$\max_{\alpha} \mathcal{L}(\alpha) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^\top \mathbf{x}_j + \sum_{i=1}^{n} \alpha_i \tag{2.14}$$

$$\text{s.t. } 0 \leq \alpha_i \leq C, \sum_{i=1}^{n} \alpha_i y_i = 0.$$

The only novelty in the soft margin case is that the dual variables α_i are now upper bounded by the parameter C. As in the hard margin case, we again have a quadratic program that can be efficiently solved by off-the-shelf solvers.

Kernel Methods

Training samples $\mathbf{x}_i \in \mathcal{X}$ are usually not linearly separable in the input space $\mathcal{X}$. After a nonlinear transformation $\phi(\cdot)$, training samples $\phi(\mathbf{x}_i)$ may be linearly separated in a

higher dimensional space (Fig. 2.2). Finding a linear classifier in a higher dimensional feature space is equivalent to finding a nonlinear classifier in the input space of training samples. However, explicit mapping of input points $\mathbf{x}_i$ to high dimensional feature points $\phi(\mathbf{x}_i)$ can be very costly, and in some cases even not possible, because feature space can be infinite dimensional. By inspection of Eq. 2.14, it can be seen that training samples $\mathbf{x}_i$ always appear only as a scalar product $\mathbf{x}_j^\top \mathbf{x}_j$. Therefore, there is no need to explicitly calculate the feature mapping $\phi(\mathbf{x}_i)$, but only to calculate the kernel function

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j). \tag{2.15}$$

The situation where the kernel function $K(\cdot, \cdot)$ is used in training and testing instead of the explicit feature transformation $\phi(\cdot)$ is known in literature as the *kernel trick*.

The kernel function is defined as the scalar product between the points in the feature space, thus it essentially represents a similarity measure between input points. To apply a kernel trick, it is enough to know an analytical form of the kernel function $K(\cdot, \cdot)$, i.e. the knowledge of the transformation function $\phi(\cdot)$ is not needed. Some of the most popular kernel functions are:

- Polynomial function of degree d

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + 1)^d \tag{2.16}$$

- Radial basis function

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2) \tag{2.17}$$

- Sigmoid function

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i^\top \mathbf{x}_j + \theta) \tag{2.18}$$

- Histogram intersection function between histograms of dimensionality d

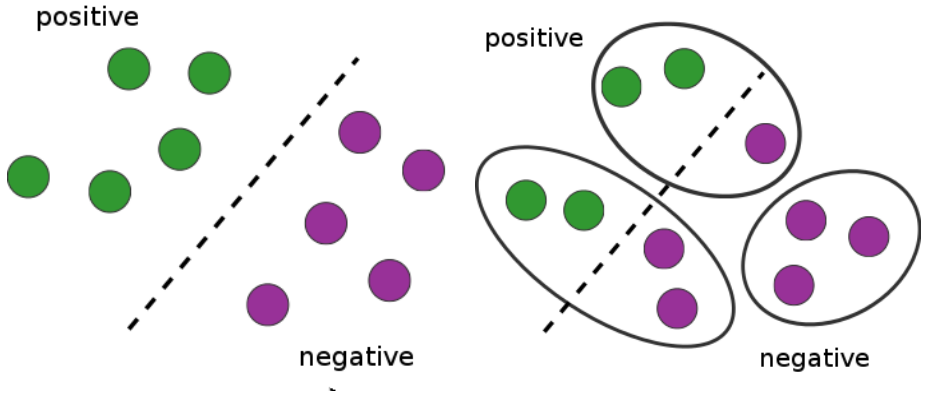
$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{d} \min(x_i(k), x_j(k)) \tag{2.19}$$

- χ^2 kernel function between histograms of dimensionality d

$$K(\mathbf{x}_i, \mathbf{x}_j) = \sum_{k=1}^{d} \frac{2x_i(k)x_j(k)}{x_i(k) + x_j(k)} \tag{2.20}$$

According to Mercer's theorem [SS01], a symmetric function $K(\cdot, \cdot)$ can be expressed as an inner product in some feature space, $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$ if and only if function $K(\cdot, \cdot)$ is positive semidefinite

$$\int K(\mathbf{x}_i, \mathbf{x}_j) h(\mathbf{x}_i) h(\mathbf{x}_j) d\mathbf{x}_i d\mathbf{x}_j \geq 0, \forall h. \tag{2.21}$$



(a) Supervised learning (b) Multiple-instance learning

Figure 2.3: Illustration of the two types of learning.

After applying the kernel trick, the dual formulation of the SVM training can be written as the following quadratic optimization program

$$\max_{\alpha} \mathcal{L}(\alpha) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) + \sum_{i=1}^{n}\alpha_i \tag{2.22}$$
$$\text{s.t. } 0 \leq \alpha_i \leq C, \sum_{i=1}^{n}\alpha_i y_i = 0.$$

Recognition of a novel instance $\mathbf{x}$ by SVM classifier also uses the kernel trick to avoid explicit calculation in the feature space. As the max-margin hyperplane in the feature space is a linear combination of support vectors $\mathbf{x}_i \in \mathcal{S}$,

$$\mathbf{w} = \sum_{\mathbf{x}_i \in \mathcal{S}} \alpha_i y_i \phi(\mathbf{x}_i), \tag{2.23}$$

the classification score is a linear combination of values of the kernel function

$$f(\mathbf{x}) = \mathbf{w}^\top \phi(\mathbf{x}) + b = \sum_{\mathbf{x}_i \in \mathcal{S}} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \tag{2.24}$$

2.3 Common MIL Approaches

In a standard supervised setting, the training set consists of features and their labels $(\mathbf{x}_i, y_i) \in \mathbb{R}^d \times \mathcal{Y}$, and the goal is to learn a classifier $f : \mathbb{R}^d \to \mathcal{Y}$, i.e. a function that maps features to labels. Multiple-instance learning is dealing with instance labels in a weakly supervised way (Fig. 2.3). Labels are provided only for sets of instances

that are called *bags*. Each bag B_I is specified by an index set $I \subseteq \{1, 2, ..., n\}$, i.e. $B_I = \{\mathbf{x}_i : i \in I\}$. Multiple-instance learning is defined on the finite set of bags $\{B_I\}_{I \in \bar{I}}$, where family of index sets $\bar{I} \subseteq 2^{\{1,2,...,n\}}$ is a subset of the power set of set $\{1, ..., n\}$. There are in total m bags in the dataset, i.e. $|\bar{I}| = m$. A label Y_I is associated with each bag B_I, and they are defined in the following way. Instances in the negative bag *all* belong to the negative class, $Y_I = -1 \Rightarrow \forall i \in I : y_i = -1$. On the other hand, a bag with positive label requires that *at least* one of its instances belongs to the positive class, $Y_I = 1 \Rightarrow \exists i \in I : y_i = 1$. The goal of MIL is to simultaneously find the latent instance labels y_i and the instance classifier f. Inferred instance labels have to satisfy MIL constraints that express the relationship between bag labels Y_I and corresponding instance labels y_i, i.e. $Y_I = \max_{i \in I} y_i$.

mi-SVM Method

Andrews et al. [ATH03] proposed mi-SVM and MI-SVM methods for multiple-instance learning that both maximize the margin of the instance classifier. In both mi-SVM and MI-SVM methods, learning problem is formulated as a mixed integer problem. In case of a linear discriminant function $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$, we estimate the weight vector $\mathbf{w} \in \mathbb{R}^d$ and offset $b \in \mathbb{R}$, and find integer values $y_i \in \{+1, -1\}$ that represent latent instance labels of the training bags

$$\min \mathcal{L}(\mathbf{w}, b, \{\xi_i\}, \{y_i\}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{I \in \mathcal{I}} \sum_{i \in I} \xi_i, \tag{2.25}$$

$$\text{s.t. } y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i, \tag{2.26}$$

$$\sum_{i \in I} \frac{y_i + 1}{2} \geq 1, \text{ if } Y_I = +1, \tag{2.27}$$

$$y_i = -1, \text{ if } Y_I = -1. \tag{2.28}$$

Inferring the latent labels and training jointly the instance classifier is a hard mixed integer problem that is standardly solved by alternating optimization. It consists of the two steps: (i) *inferring the labels* - given the discriminant function, find the integer variables y_i that correspond to the unknown instance labels in training bags, (ii) *classifier learning* - given the inferred instance labels from the previous step, find the optimal parameters $(\mathbf{w}, b)$ of the discriminant function. These two steps are performed on the same training bags simultaneously, which means that the same instances are used for both training the discriminant function and imputing the missing labels. Inferring the latent labels with the classifier that is evaluated on the same data it was trained upon makes the MIL procedure very susceptible to overfitting.

MI-SVM Method

MI-SVM method which is also proposed by Andrews et al. [ATH03], aims to find the single most positive instance denoted as *witness* in each positively labeled bag. The witnesses are used with all instances from negative bags to train the instance classifier. However, instances of positive bags that are not selected as witnesses are not used for

training the classifier. MI-SVM method solves the following non-convex optimization problem

$$\min \mathcal{L}(\mathbf{w}, b, \{\xi_I\}, \{\xi_i\}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\big(\sum_{I:Y_I=+1} \xi_I + \sum_{I:Y_I=-1} \sum_{i \in I} \xi_i\big) \tag{2.29}$$

$$\text{s.t. } \max_{i \in I}(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_I, \text{ if } Y_I = +1, \tag{2.30}$$

$$\mathbf{w}^\top \mathbf{x}_i + b \leq -1 + \xi_i, \forall i \in I \text{ if } Y_I = -1, \tag{2.31}$$

$$\xi_I, \xi_i \geq 0, \forall I, i. \tag{2.32}$$

The problem can also be reformulated as a mixed integer program by introducing a witness selector variable $s_I \in I$:

$$\min \mathcal{L}(\mathbf{w}, b, \{\xi_I\}, \{\xi_i\}, \{s_I\}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\big(\sum_{I:Y_I=+1} \xi_I + \sum_{I:Y_I=-1} \sum_{i \in I} \xi_i\big) \tag{2.33}$$

$$\text{s.t. } \mathbf{w}^\top \mathbf{x}_{s_I} + b \geq 1 - \xi_I, \text{ if } Y_I = +1, \tag{2.34}$$

$$\mathbf{w}^\top \mathbf{x}_i + b \leq -1 + \xi_i, \forall i \in I \text{ if } Y_I = -1, \tag{2.35}$$

$$\xi_I, \xi_i \geq 0, \forall I, i. \tag{2.36}$$

Alternating optimization approach is again applied to find parameters $(\mathbf{w}, b)$ of the instance classifier and select witnesses s_I in positive bags.

MICA Method

Selection of a single witness s_I from a positive bag can be too restricting. Mangasarian et al. [MW05] proposed the MICA method that relaxes witness selection to a convex combination over all instances in a positive bag. Parameters ν_i in the convex combination are all nonnegative and sum up to one.

$$\min \mathcal{L}(\mathbf{w}, b, \{\xi_i\}, \{\nu_i\}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\big(\sum_{I:Y_i=+1} \xi_I + \sum_{I:Y_I=-1} \sum_{i \in I} \xi_i\big) \tag{2.37}$$

$$\text{s.t. } \sum_{i \in I} \nu_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_I, \sum_{i \in I} \nu_i = 1, \text{ if } Y_I = +1, \tag{2.38}$$

$$\mathbf{w}^\top \mathbf{x}_i + b \leq -1 + \xi_i, \forall i \in I \text{ if } Y_I = -1, \tag{2.39}$$

$$\xi_I, \xi_i, \nu_i \geq 0, \forall I, i. \tag{2.40}$$

Although the resulting optimization problem is continuous, it is not convex, because of the bilinear function from the margin inequality constraints for positive bags.

Deterministic Annealing

Before we discuss multiple-instance learning algorithms that use deterministic annealing for optimization [GC07], we briefly introduce here the general concepts of deterministic annealing (DA) [Ros98]. Deterministic annealing has been applied in various

discrete optimization problems such as clustering, classification or compression. It describes a procedure to solve a discrete optimization problem

$$\mathbf{y} = \underset{\mathbf{y}' \in \{0,1\}^n}{\operatorname{argmin}} J(\mathbf{y}'), \tag{2.41}$$

for arbitrary objective function $J(\cdot)$. The main idea of deterministic annealing is to replace discrete variables with binary random variables that belong to the space of discrete probability distributions $\mathcal{P}$. We search for a distribution $p \in \mathcal{P}$, so that the expected value of the objective function $J(\cdot)$ is minimal,

$$p = \underset{p' \in \mathcal{P}}{\operatorname{argmin}} \mathbb{E}_{p'}(J(\mathbf{y})) - T \cdot \mathbb{H}(p'). \tag{2.42}$$

The parameter T is called temperature, and if it is equal to zero, $T = 0$, we obtain the global optimum of the original discrete optimization problem in Eq. 2.41. The objective function is made easier for optimization by adding a convex entropy term $-T \cdot \mathbb{H}(p)$ to the original objective function. The original problem can be solved through a sequence of intermediate optimization problems that are obtained for decreased (annealed) temperature values $T_0 > T_1 > \cdots > T_\infty = 0$, where each problem is initialized with a solution of the previous problem. There are no theoretical guarantees that DA will find the global optimum; however, it usually finds a solution close enough to the global optimum.

AL-SVM Method

Gehler and Chapelle [GC07] proposed a deterministic annealing approach to solve the mi-SVM problem. They regards latent instance labels y_i as binary random variables. Moreover, they assume that the space $\mathcal{P}$ consists only of probability distributions that can be factorized into a product of marginal probabilities $p_i = P(y_i = +1)$. All instances in negative bags have a zero-valued marginal probability $p_i = 0$. As each positive bag B_I contains at least one positive instance, the constraint $\sum_{i \in I} p_i \geq 1$ is introduced:

$$\begin{aligned} \min \mathcal{L}(\mathbf{w}, b, \{p_i\}) = \|\mathbf{w}\|^2 + \sum_{I \in \mathcal{I}} \sum_{i \in I} \Big\{ & C\big(p_i \max(0, 1 - \mathbf{w}^\top \mathbf{x}_i - b) \\ & + (1 - p_i) \max(0, 1 + \mathbf{w}^\top \mathbf{x}_i + b)\big) \\ & + T\big(p_i \log(p_i) + (1 - p_i) \log(1 - p_i)\big) \Big\} \end{aligned}$$

$$\sum_{i \in I} p_i \geq 1, \text{ if } Y_I = +1, \tag{2.43}$$

$$p_i = 0, \forall i \in I, \text{ if } Y_I = -1. \tag{2.44}$$

The non-convex objective function can be minimized in a coordinate descent fashion

$$(\mathbf{w}^{(t+1)}, b^{(t+1)}) = \underset{\mathbf{w}, b}{\operatorname{argmin}} \mathcal{L}(\mathbf{w}, b, p^{(t)}) \tag{2.45}$$

$$p^{(t+1)} = \underset{p}{\operatorname{argmin}} \mathcal{L}(\mathbf{w}^{(t+1)}, b^{(t+1)}, p) \tag{2.46}$$

The optimization problem in Eq. 2.45 is a quadratic program similar to the SVM dual problem. The optimization problem in Eq. 2.46 is a convex optimization program with constraints that can be solved using the Lagrangian multipliers:

$$\min \mathcal{L}'(p,\lambda) = \mathcal{L}(\mathbf{w}^{(t+1)}, b^{(t+1)}, p) + \sum_{I:Y_I=+1} \lambda_I \Big(1 - \sum_{i \in I} p_i\Big), \lambda_I \geq 0.$$

The solution of the dual problem is $p_i = \sigma\Big(-(C/T)\big(\max(0, 1-\mathbf{w}^\top \mathbf{x}_i - b) - \max(0, 1+\mathbf{w}^\top \mathbf{x}_i + b)\big) + (1/T)\lambda_I\Big)$, and the dual variables λ_I are determined by the Karush-Kuhn-Tucker conditions.

AW-SVM

The AW-SVM algorithm [GC07] uses DA to find solution to MI-SVM, which infers only those latent labels that are *witnesses* of the class that the whole bag is assigned to. The AW-SVM method calculates a probability p_i that an instance $\mathbf{x}_i$ is a witness in its bag. The sum of probabilities in every positive bag has to be one, $\sum_{i:i\in I} p_i = 1$. For negative bags, all instances are regarded as witnesses of the negative class, $p_i = 1$, since their labels are fixed to $y_i = -1$.

$$\mathcal{L}(\mathbf{w}, b, p) = \|\mathbf{w}\|^2 + C \sum_{I \in \mathcal{I}} \sum_{i \in I} \Big\{ p_i \max\big(0, 1 - Y_i(\mathbf{w}^\top \mathbf{x}_i + b)\big) + T p_i \log p_i \Big\},$$

$$\sum_{i \in I} p_i = 1, \text{ if } Y_I = +1, \tag{2.47}$$

$$p_i = 1, \forall i \in I, \text{ if } Y_I = -1. \tag{2.48}$$

The solution is found by alternating optimization: first SVM parameters $(\mathbf{w}, b)$ are found in a standard SVM training, then p is calculated using the Lagrangian multipliers and it yields $p_i = \exp\Big(-(C/T) \cdot \max\big(0, 1 - Y_I(\mathbf{w}^\top \mathbf{x}_i + b)\big) - (1/T)\lambda_I\Big)$.

ALP-SVM

The mi-SVM algorithm can be initialized by labeling all instances in a positive bag as positive. As a result, the mi-SVM method overestimates the number of positive instances in a positive bag and the optimization gets easily trapped in a bad local optimum. However, DA does not label too many instances positively, but it suffers from a mild MIL constraint requiring at least one positively labeled instance in a positive bag. Typically, only few instances in a positive bags will be labeled as positive. As a remedy, Gehler and Chapelle [GC07] proposed to add a balancing term to the MIL objective, that penalizes large discrepancy between a number of positively labeled instances in a bag and their expected number, given as the α_I-fraction of the size of a bag I.

$$\begin{aligned}\min \mathcal{L}(\mathbf{w}, b, \{p_i\}) = \|\mathbf{w}\|^2 + \sum_{I\in\mathcal{I}} \sum_{i\in I} \Big\{ & C_1 \big(p_i \max\big(0, 1 - \mathbf{w}^\top \mathbf{x}_i - b\big) \\ & + (1-p_i) \max\big(0, 1 + \mathbf{w}^\top \mathbf{x}_i + b\big) \big) \\ & + T\big(p_i \log(p_i) + (1-p_i) \log(1-p_i) \big) \Big\} \\ & + C_2 \sum_{I\in\mathcal{I}} \big(\sum_{i\in I} p_i - \alpha_I |I| \big)^2 . \end{aligned} \tag{2.49}$$

2.4 Multiple-Instance learning with Superbags

In multiple-instance learning framework training of the classifier and inference of latent instance labels are always performed on the same training samples, which renders MIL algorithms susceptible to overfitting, and increases the variance of predicted instance labels. In this chapter an approach is proposed that improves the robustness of multiple-instance learning by decoupling the training of discriminative classifier and the inference of latent instance labels. Inference and learning steps of MIL algorithm are performed on separate bags and the results thereof are eventually combined for training a final classifier. By using separate sets of bags for classifier training and label inference, the final classifier obtains a lower error in predicting latent instance labels. With respect to the well-known bias-variance decomposition of the mean squared error, integration of superbags into MIL decreases the variance of label predictions without increasing the bias. Superbags slightly prolong the MIL training time, but the testing time stays the same. In the sequel it is explained how label inference and classifier training steps can be separated using superbags that are integrated into MIL.

If the instance classifier $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ is known[1], latent instance labels can be predicted by assigning all instances to positive or negative class based on the sign of the classification score, $y_i = \operatorname{sgn} f(\mathbf{x}_i)$. However, the discriminant function $f(\mathbf{x})$ is not available, so it needs to be learned from the training instances whose labels are also not provided. In multiple-instance learning, instance label predictions y_i, and the classifier parameter $\mathbf{w}$ are jointly updated. Labels $y_i \in \{-1, +1\}$ should agree with predictions that the classifier $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ makes about instances $\mathbf{x}_i$. Optimal instance label y_i is thus obtained by minimizing the classification loss function $\ell(\mathbf{x}_i, y_i, \mathbf{w})$. Predicted labels also have to satisfy the MIL constraints:

$$\sum_{i\in I} \frac{y_i + 1}{2} \geq 1, \text{ if } Y_I = 1, \tag{2.50}$$

$$y_i = -1, \forall i \in I, \text{ if } Y_I = -1. \tag{2.51}$$

The regularization function $\Omega(\mathbf{w})$ is used to prevent excessive values of the classifier's parameter $\mathbf{w}$. Classification errors are measured by the empirical loss func-

[1] Offset b of the discriminant function is included in the weight vector $\mathbf{w}$.

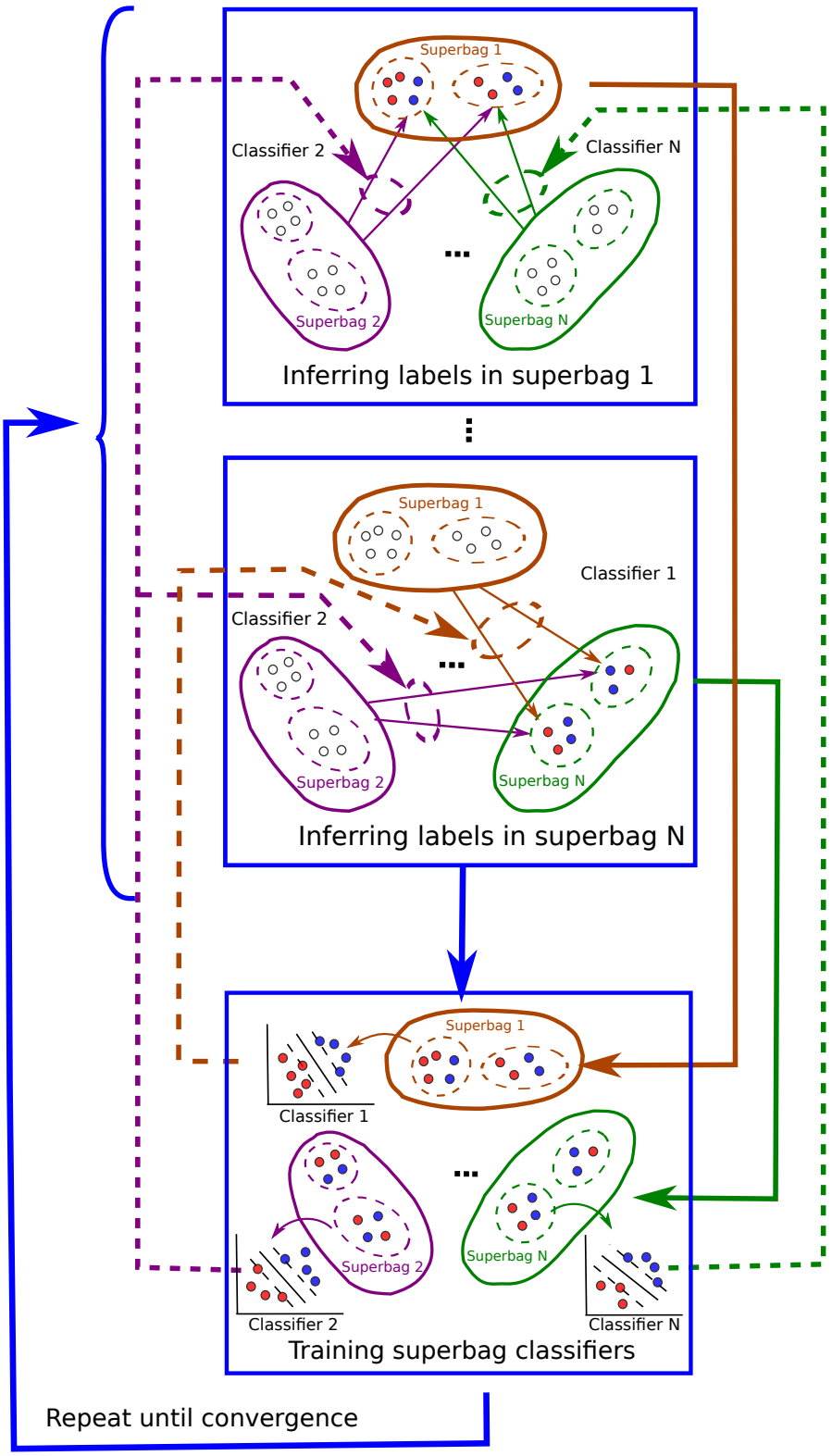


Figure 2.4: Sketch of the processing pipeline for decoupling classifier training and label inference by performing them on different superbags.

tion $R_{emp}(\mathbf{w})$, that is defined as a sum of loss functions for all training instances, $R_{emp}(\mathbf{w}) := \sum_{i=1}^{n} \ell(\mathbf{x}_i, y_i, \mathbf{w})$. The classifier $\mathbf{w}$ is obtained by minimizing the sum of the regularizer $\Omega(\mathbf{w})$ and the empirical loss function $R_{emp}(\mathbf{w})$,

$$\mathbf{w} = \operatorname*{argmin}_{\mathbf{w}'} \Omega(\mathbf{w}') + C \cdot R_{emp}(\mathbf{w}'). \tag{2.52}$$

As both the instance labels and the classifier hyperplane are updated on the same data, the two processes become strongly entangled, which increases the error of label predictions. The mean squared error (MSE) of the label prediction $\hat{y}_i$, given the ground truth label y_i, is defined as $MSE = \mathbb{E}\{(\hat{y}_i - y_i)^2\}$, and it can be expressed as the sum of the squared bias $b^2 = (\mathbb{E}\{\hat{y}_i\} - y_i)^2$ and the variance $\sigma^2 = \mathbb{E}\{(\hat{y}_i - \mathbb{E}\{\hat{y}_i\})^2\}$. The bias is a measure of systematic error in label prediction and it is larger if the model is less flexible. On the other hand, the variance measures the variability of predicted labels around the average prediction. Overly flexible models have a large variance, because they easily overfit to the training samples.

A natural question is whether overfitting can be avoided by decreasing the variance of label predictions that will then lead to a more robust model that better generalizes to new samples. The solution (illustrated in Figure 2.4) is based on two observations: (i) overfitting can be avoided if the labels are predicted by classifiers that are not trained on the same training samples, and, (ii) the variance of label predictions can be decreased by averaging predictions obtained by different instance classifiers. Both of these requirements can be satisfied if we randomly separate all training bags into multiple sets of bags - *superbags*. A classifier is trained on each superbag and used to predict labels of instances that belong to other superbags. That way, the training of the classifier and the inference of latent labels are decoupled. Moreover, for each unknown label predictions from other classifiers of the ensemble are aggregated, and by their averaging the variance of the instance label predictions is reduced. We illustrate this in Sect. 2.6.1 that analyzes the uncertainty of label prediction in MIL.

As mentioned before, superbags are created by randomly sampling the sets of training bags. The result is an ensemble of superbags $\{S_{\mathcal{I}}\}_{\mathcal{I}\in\hat{\mathcal{I}}}$, where a superbag consists of a set of bags, $S_{\mathcal{I}} = \{B_I : I \in \mathcal{I}\}$. The family of superbag indexes $\hat{\mathcal{I}} \subseteq 2^{\bar{I}}$ is a subset of the power set of the bag indexes $\bar{I}$. Let k denote the total number of superbags, $|\hat{\mathcal{I}}| = k$. Superbags are created by sampling bags with replacement, otherwise classifiers would be trained on too few training bags in superbags. Consequently, superbags might overlap, i.e. superbag index sets $\mathcal{I} \in \hat{\mathcal{I}}$ are in general not disjoint. The size of a superbag is set to be the fraction r of the total number of bags in the dataset, i.e. $\forall \mathcal{I} \in \hat{\mathcal{I}} : |S_{\mathcal{I}}| = r \cdot |\bar{I}|$. A separate classifier $f_{\mathcal{I}}$ is trained on each superbag $S_{\mathcal{I}}$, i.e. a hyperplane $\mathbf{w}_{\mathcal{I}}$ is learned only from instances that belong to superbag $S_{\mathcal{I}}$. The hyperplane $\mathbf{w}_{\mathcal{I}}$ is selected from a set of well-behaved functions that are determined by the regularizer $\Omega(\mathbf{w}_{\mathcal{I}})$, and it also has to fit well to the training data in the superbag $S_{\mathcal{I}}$. This is quantified by the superbag empirical loss function,

$$R_{emp}(\mathbf{w}_{\mathcal{I}}) := \sum_{I\in\mathcal{I}} \sum_{i\in I} \ell(\mathbf{x}_i, y_i, \mathbf{w}_{\mathcal{I}}). \tag{2.53}$$

Superbag classifier is found by minimizing the sum of the regularizer and the superbag empirical loss function,

$$\hat{\mathbf{w}}_{\mathcal{I}} := \operatorname*{argmin}_{\mathbf{w}_{\mathcal{I}}} \Omega(\mathbf{w}_{\mathcal{I}}) + C \cdot R_{emp}(\mathbf{w}_{\mathcal{I}}). \tag{2.54}$$

The goal is to decouple the inference of missing instance labels y_i and the training of the ensemble of classifiers $\mathbf{w}_{\mathcal{I}}$. Therefore, ideally label y_i of instance $\mathbf{x}_i$ is predicted only by classifiers that are trained on superbags which do not contain the point $\mathbf{x}_i$. However, in order to provide numerical stability of the iterative procedure, we also include predictions made by classifiers that are trained on $\mathbf{x}_i$, but these predictions are given a smaller weight β, whose value is determined by cross-validation. Consequently, the labels y_i are inferred by the following optimization,

$$\hat{y}_i = \operatorname*{argmin}_{y_i} \sum_{\mathcal{I}:i\notin\mathcal{I}} \ell(\mathbf{x}_i, y_i, \mathbf{w}_{\mathcal{I}}) + \beta \sum_{\mathcal{I}:i\in\mathcal{I}} \ell(\mathbf{x}_i, y_i, \mathbf{w}_{\mathcal{I}}), \tag{2.55}$$

subject to the general MIL constraints defined earlier in Eq. 2.50 and 2.51. Note that standard MIL is a special case of superbag MIL when $k = 1$ and $r = 1$, i.e. when there is only one superbag which contains all the training bags. In that case, only the second term remains in Eq. 2.55.

Difference to Co-training and Bagging

Training a classifier with missing labels also appears in semi-supervised learning, where it is addressed by co-training [BM98]. In co-training, two classifiers are trained on the same labeled set of points, but with different features. The classifiers then predict labels of a large unlabeled set of points. Patterns that are confidently labeled by either of the classifiers are appended to the training set of both classifiers. So the two classifiers are always updated with the same set of training points. Confident label predictions of one classifier are used during co-training to resolve ambiguities about unlabeled patterns of the other classifier. Different from the concept of co-training, the superbag approach trains the classifiers on the *same* features, but using *different* data points. Superbag classifiers are all trained on the same features, since in many applications finding new independent features is not feasible. Usually, it is not possible to change the feature representation of the given data. Training classifiers on the same data, as performed by co-training, makes the classifiers dependent on each other and their training less robust. The superbag approach trains the classifiers on different data points which reduces the variance of inferred labels and increases the robustness of the classifiers. Superbag classifiers are trained only from weakly labeled points, whereas in co-training the classifiers are trained initially on the fully labeled set of points.

Classifying data patterns with multiple classifiers is also part of the bagging method [Bre96]. Bagging makes several training sets by sampling them with replacement from the original set of points. These sets are then used for training the ensemble of classifiers, which later jointly classify new test points. However, bagging can be applied only in the supervised setting, where the labels of training instances are all provided. In contrast, the superbag ensemble is trained on weakly labeled patterns, where finding

the missing labels and training the classifiers are performed simultaneously. Besides, in the superbag approach an ensemble of classifiers is used only during training to robustly resolve the missing labels, whereas in bagging the ensemble of classifiers is used in testing. After inferring the latent labels of training instances in the superbag method, the final classifier is trained to predict labels of novel instances.

2.5 Integrating Superbags into Common MIL Approaches

In this section we show how the concept of superbags can be easily integrated into some of the most popular instance-level classifiers for MIL that are based upon the standard soft-margin SVM formulation [ATH03, GC07]. In particular, we choose the methods AL-SVM, AW-SVM and ALP-SVM proposed by Gehler and Chapelle [GC07], because they generalize the widely employed mi-SVM and MI-SVM methods [ATH03] used in many different applications. Sect. 2.5.1 - 2.5.3 show how to integrate the concept of superbags into the methods of AL-SVM, AW-SVM and ALP-SVM, respectively.

2.5.1 AL-SVM with Superbags

The concept of superbags can be easily integrated into the AL-SVM algorithm. An ensemble of classifiers $\mathbf{w}_\mathcal{I}$ is trained on all patterns from a superbag $S_\mathcal{I}$ by minimizing the regularized empirical loss,

$$\hat{\mathbf{w}}_\mathcal{I} = \operatorname*{argmin}_{\mathbf{w}_\mathcal{I}} \Omega(\mathbf{w}_\mathcal{I}) + C \cdot R_{emp}(\mathbf{w}_\mathcal{I}), \tag{2.56}$$

with the empirical loss function

$$R_{emp}(\mathbf{w}_\mathcal{I}) := \sum_{I \in \mathcal{I}} \sum_{i \in I} \big(p_i \ell(\mathbf{x}_i, y_i = +1, \mathbf{w}_\mathcal{I}) + (1 - p_i)\ell(\mathbf{x}_i, y_i = -1, \mathbf{w}_\mathcal{I})\big). \tag{2.57}$$

In contrast to Eq. 2.53 that operates on deterministic label assignments y_i, Eq. 2.57 contains the expectation of the empirical loss, because latent labels are now treated as binary random variables. In case of the quadratic regularizer $\Omega(\mathbf{w}_\mathcal{I}) = \frac{1}{2}\|\mathbf{w}_\mathcal{I}\|^2$ and the standard hinge-loss function $\ell(\mathbf{x}_i, y_i, \mathbf{w}_\mathcal{I}) = \max\big(0, 1 - y_i \cdot (\mathbf{w}_\mathcal{I}^\top \mathbf{x}_i)\big)$, the optimization problem becomes a quadratic program (QP), that is solved by standard solvers.

Latent instance labels are inferred by minimizing the sum of empirical losses incurred by superbag classifiers,

$$\begin{aligned} \hat{p}_i = \operatorname*{argmin}_{p_i} C\Big(& \sum_{\mathcal{I}: i \notin \mathcal{I}} \big(p_i \ell(\mathbf{x}_i, y_i = +1, \mathbf{w}_\mathcal{I}) + (1 - p_i)\ell(\mathbf{x}_i, y_i = -1, \mathbf{w}_\mathcal{I})\big) + \\ & \beta \sum_{\mathcal{I}: i \in \mathcal{I}} \big(p_i \ell(\mathbf{x}_i, y_i = +1, \mathbf{w}_\mathcal{I}) + (1 - p_i)\ell(\mathbf{x}_i, y_i = -1, \mathbf{w}_\mathcal{I})\big)\Big) + \qquad (2.58) \\ & T \cdot \big(p_i \log(p_i) + (1 - p_i)\log(1 - p_i)\big). \qquad (2.59) \end{aligned}$$

The probabilities p_i can be obtained in a closed form. MIL constraints can be added to the objective function using Lagrangian multipliers as in the baseline AL-SVM method.

2.5.2 AW-SVM with Superbags

AW-SVM [GC07] can be easily integrated with the concept of superbags. A classifier $\mathbf{w}_{\mathcal{I}}$ is trained only on training instances from the superbag $S_{\mathcal{I}}$, which is achieved by minimizing the regularized expected empirical loss function,

$$\hat{\mathbf{w}}_{\mathcal{I}} = \underset{\mathbf{w}_{\mathcal{I}}}{\operatorname{argmin}}\, \Omega(\mathbf{w}_{\mathcal{I}}) + C \cdot R_{emp}(\mathbf{w}_{\mathcal{I}}), \tag{2.60}$$

with the expected empirical loss function

$$R_{emp}(\mathbf{w}_{\mathcal{I}}) := \sum_{I \in \mathcal{I}} \sum_{i \in I} p_i \ell(\mathbf{x}_i, y_i = Y_I, \mathbf{w}_{\mathcal{I}}). \tag{2.61}$$

For the quadratic regularizer and the hinge-loss function, the optimization in Eq. 2.61 is a standard QP that can be solved by standard solvers.

In the inference step, the superbag-based AW-SVM finds the probabilities p_i that instances $\mathbf{x}_i$ is a witness by minimizing the expected empirical loss given the general MIL constraints as before,

$$\hat{p}_i = \underset{p_i}{\operatorname{argmin}}\, C\Big(\sum_{\mathcal{I}: i \notin \mathcal{I}} p_i \ell(\mathbf{x}_i, y_i = Y_I, \mathbf{w}_{\mathcal{I}}) + \beta \sum_{\mathcal{I}: i \in \mathcal{I}} p_i \ell(\mathbf{x}_i, y_i = Y_I, \mathbf{w}_{\mathcal{I}})\Big) + T p_i \log(p_i). \tag{2.62}$$

The solution of the inference problem can be easily obtained in the closed form by solving the Lagrangian dual as in the baseline AW-SVM.

2.5.3 ALP-SVM with Superbags

As the balancing term $\sum_I (\sum_{i \in I} p_i - \alpha_I |I|)^2$ does not depend on the parameters $\mathbf{w}_{\mathcal{I}}$ of the ensemble of superbag classifiers, the classifiers can be trained in the same way as in Eq. 2.56. However, the label inference step is changed by adding the balancing term, and it has now the following form,

$$\hat{p} = \underset{p}{\operatorname{argmin}} \sum_i \Big\{ C_1 \Big(\sum_{\mathcal{I}: i \notin \mathcal{I}} \big(p_i \ell(\mathbf{x}_i, y_i = +1, \mathbf{w}_{\mathcal{I}}) + (1 - p_i) \ell(\mathbf{x}_i, y_i = -1, \mathbf{w}_{\mathcal{I}})\big) + \tag{2.63}$$

$$\beta \sum_{\mathcal{I}: i \in \mathcal{I}} \big(p_i \ell(\mathbf{x}_i, y_i = +1, \mathbf{w}_{\mathcal{I}}) + (1 - p_i) \ell(\mathbf{x}_i, y_i = -1, \mathbf{w}_{\mathcal{I}})\big)\Big) + \tag{2.64}$$

$$T \cdot \big(p_i \log(p_i) + (1 - p_i) \log(1 - p_i)\big)\Big\} + C_2 \sum_I (\sum_{i \in I} p_i - \alpha_I |I|)^2. \tag{2.65}$$

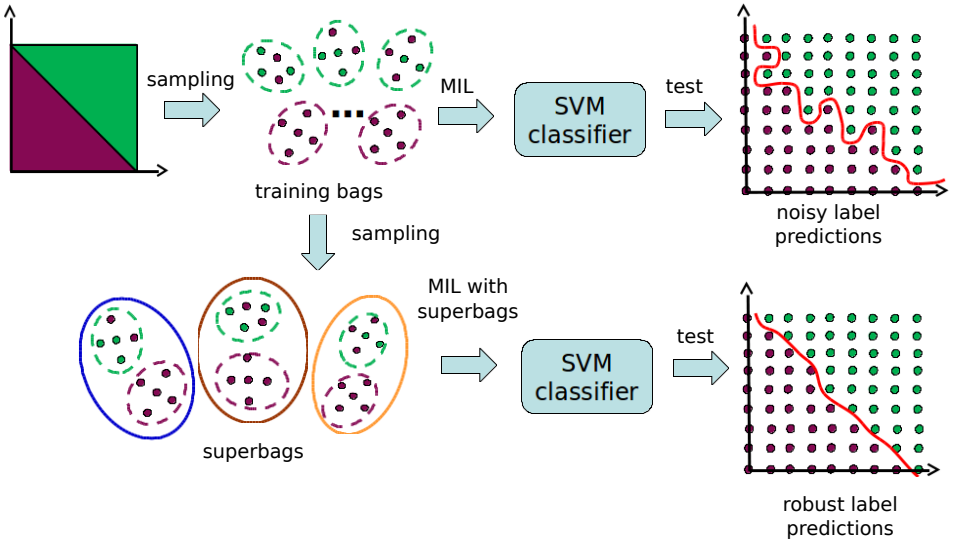


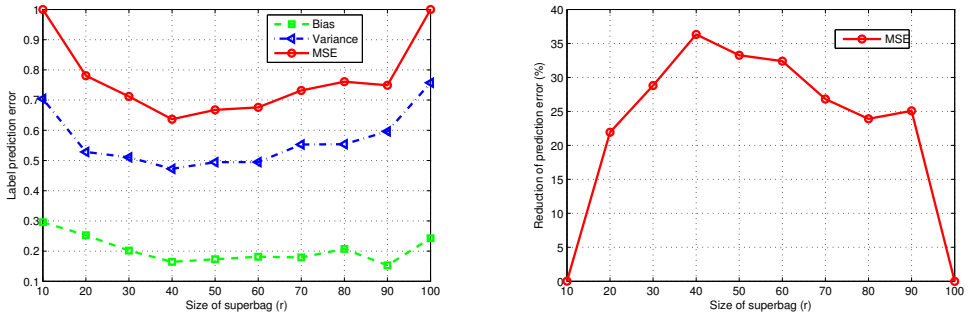
Figure 2.5: The workflow of the synthetic experiment.

2.6 Experimental Evaluation

In the experimental section, we evaluate how integrating superbags improves MIL by decreasing the variance of label predictions and avoiding overfitting. In Sect. 2.6.1 we first analyze the uncertainty of label predictions in a synthetic experiment, and in Sect. 2.6.2 standard MIL benchmark datasets are used to measure the performance gains after integrating the ensemble of superbags into some popular MIL methods that are used as baselines. Finally, in Sect. 2.6.3 we show how performance of the MIL based image re-ranking system can be improved if the concept of superbags is applied to it.

2.6.1 Analyzing the Uncertainty of Label Predictions in MIL

A synthetic experiment (Fig. 2.5) is created in order to analyze the uncertainty of label predictions in MIL. The dataset consists of $m = 100$ bags, where each bag has five points sampled from a unit square in the plane. The diagonal of the square separates the positive and the negative class. 30 bags are sampled strictly from the negative class (negative bags), while other bags are sampled from both classes. Baseline MIL algorithm in this experiment is mi-SVM, which infers the missing labels for all training instances. The mi-SVM algorithm can be obtained from AL-SVM by setting the temperature T to zero. We integrate the idea of superbags into the baseline mi-SVM method. Ten superbags are created by random sampling, with their size being changed from $r = 10\%$ to $r = 100\%$ in an increment of 10%. We note that superbags of the maximal size $r = 100\%$ both train the classifiers and find the missing labels on all the training data at once, which corresponds to the baseline mi-SVM method. We use linear



(a) Label prediction error (b) Reduction of the prediction error

Figure 2.6: Analysis of the label prediction errors in the synthetic experiment. Fig. (a) shows the mean squared error (MSE) and its two components, variance and squared bias, when the size of superbags r is changed. All quantities are normalized with respect to MSE of the baseline mi-SVM method ($r = 1$). Fig. (b) shows in percents the reduction of the mean squared error with respect to the baseline mi-SVM method. The largest decrease of the prediction error (35%) is obtained for superbags of the size $r = 0.4$.

SVM classifier and fix the hyperparameters to $C = 20$ and $\beta = 0.5$.

The averaged results over five hundred independent runs are given in Fig. 2.6. It shows the uncertainty of label predictions measured by the mean squared error (MSE), and its two components, variance and squared bias, that were discussed in Sect. 2.4. All three quantities are normalized with respect to the mean squared error of the baseline mi-SVM method. The baseline method is obtained for $r = 1$. The variance quantifies how much the model is susceptible to overfitting. We see that by decreasing the size of superbags r, the variance of label predictions decreases, because overfitting becomes less prominent. This is a direct consequence of the decoupling of label inference and classifer training when the size of superbags is decreased. In this experiment, the changes of bias are smaller than the changes of variance. This is because the complexity of the original SVM classifier is not changed. The percent of reduction of MSE with respect to the baseline performance is shown on the right of Fig. 2.6. We see that the prediction error is always lower than in the baseline, and the maximal reduction is achieved for the superbag size of $r = 0.4$. The error rate in that case drops by approximately 35%.

2.6.2 Evaluation on Benchmark Datasets

In this section, we evaluate the proposed concept of superbags on the benchmark datasets for MIL. The ensemble of superbags is integrated with the popular AL-SVM, AW-SVM and ALP-SVM algorithms proposed by Gehler and Chapelle [GC07], and the results are compared to their baseline versions that do not use superbags. The benefit of these classifiers is that they can resolve ambiguous instance labels jointly with bag-level classification. Moreover, mi-SVM-like classifiers are quite popular in the MIL literature, and have found a wide use in computer vision. We compare the results also to

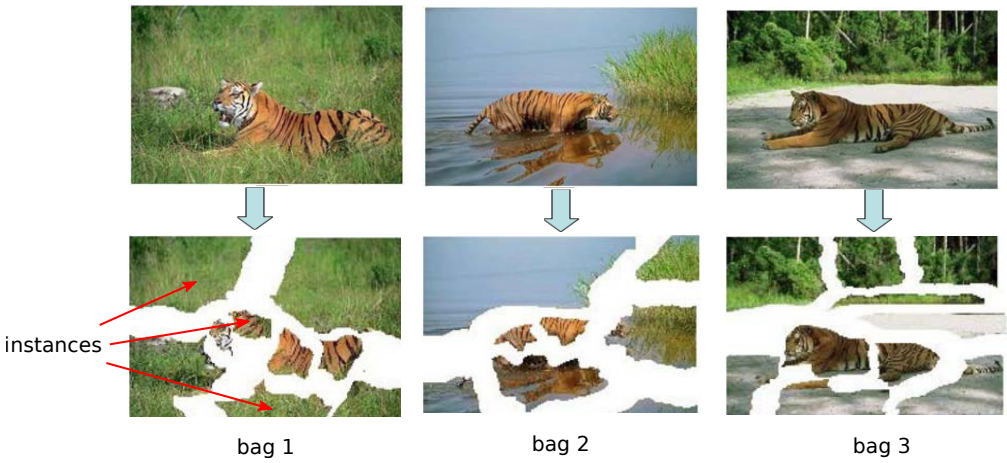


Figure 2.7: The construction of bags and instances in the Corel dataset.

the bag-level classifiers [FRKZ11], which predict only bag labels using rich bag-level features, but do not infer missing instance labels, and are, thus, not applicable for many applications.

We use the well-established benchmark sets, MUSK [DL97] (Musk1 and Musk2) and COREL [ATH03] (Tiger, Elephant and Fox), for the comparison of MIL algorithms (Fig. 2.7). We follow the experimental setup of Gehler and Chapelle [GC07], and use SVM classifier with RBF kernel whose bandwidth σ and parameter C are selected from the sets $\sigma \in \{ 0.5\sigma_0, \sigma_0, 2\sigma_0\}$ and $C \in \{1, 10\}$ by tenfold cross-validation. The value of σ_0 is computed as the median of pairwise distances of all training samples. The balance term α in ALP-SVM is selected by cross-validation from the set $\alpha \in \{0.1, 0.2, ..., 1.0\}$. AL-SVM and AW-SVM are performed without annealing, because, as noted by Gehler and Chapelle [GC07], annealing in their case does not translate into a smaller test error. Consequently, annealing sequence is only applied to ALP-SVM, where it starts from the temperature $T = 10C$ and is decreased at the rate of $2/3$ per round.

We integrate the idea of superbags into three standard MIL algorithms, AL-SVM, AW-SVM and ALP-SVM, and compare the results with their baseline versions that do not use superbags. We sample randomly $k = 10$ superbags. We select superbag size r from the set $r \in \{20\%, 50\%, 80\%\}$ and $\beta \in \{0.5, 1, 2\}$ by cross-validation. Superbag size r is the fraction of the total number of bags in the dataset. The best performance is typically obtained for $\beta = 0.5$, i.e. instance labels are strongly predicted by classifiers trained on different superbags.

The results of testing on MIL benchmark datasets are given in the Tab. 2.1. Classification error of superbag concept integrated into AL-SVM, AW-SVM and ALP-SVM methods is compared with the baseline version that do not use superbags. It is evident that the introduction of superbags to the standard MIL methods consistently improves the their baseline versions. The AL-SVM method achieves the gain of $+5\%$ on the Tiger dataset when superbags are used. The AW-SVM method shows the gain of up to $+3.5\%$ on the Fox dataset. Lastly, the ALP-SVM approach shows an improvement of $+3\%$ on

Table 2.1: The classification error (%) on five different MIL benchmark datasets. We compare the performance of methods AL-SVM, AW-SVM and ALP-SVM when they use superbags to their baseline versions that are without superbags. In all cases, a consistent improvement in performance is achieved.

Dataset	AW - SVM			AL - SVM			ALP - SVM		
	B/L	S-bags	Gain	B/L	S-bags	Gain	B/L	S-bags	Gain
Musk1	14.3	14.2	+0.1	13.3	13.1	+0.2	13.7	12.1	+1.6
Musk2	16.2	13.8	+2.4	17.4	17.4	0	13.8	13.4	+0.4
Tiger	17	14.5	+2.5	21.5	16.5	+5	14	14	0
Elephant	18	17.5	+0.5	20.5	17.5	+3	16.5	16	+0.5
Fox	36.5	33	+3.5	36.5	33	+3.5	34	31	+3

Table 2.2: The comparison of the state-of-the-art methods for MIL classification on the MUSK and Corel benchmark datasets.

Dataset	EMDD [ZG01]	mi-SVM [ATH03]	MI-SVM [ATH03]	MILIS [FRKZ11]	Superbag MIL
Musk1	15.2	12.6	22.1	11.4	12.1
Musk2	15.1	16.4	15.7	8.9	13.4
Tiger	27.9	21.6	16	-	14.0
Elephant	21.7	17.8	18.6	-	16.0
Fox	43.9	41.8	42.2	-	31.0

the Fox dataset.

Tab. 2.2 also provides the results of other MIL methods, namely EMDD [ZG01], MILIS[FRKZ11], MI-SVM [ATH03] and mi-SVM [ATH03]. The integration of the ensemble of superbags into ALP-SVM achieves the best score on all three COREL datasets, Tiger, Elephant and Fox. The results on the Tiger dataset are equal to the baseline ALP-SVM method. The superbags also significantly improve the performance of baseline methods on the MUSK datasets. On the Musk1 dataset, the performance of ALP-SVM after integration with superbags is *on par* ($< 1\%$ difference) with MILIS [FRKZ11], the state-of-the-art method for the MUSK benchmark. Good performance of MILIS on Musk2 dataset is due to the rich bag-level features that MILIS as a bag-level classifier uses. As a consequence, bag-level classifiers are however limited in that they do not infer missing instance labels. By integrating superbags into the instance-level classifiers, their bag-level classification performance approaches the performance of the state-of-the-art bag-level classifiers, which are inferior in terms of the labels they can infer.

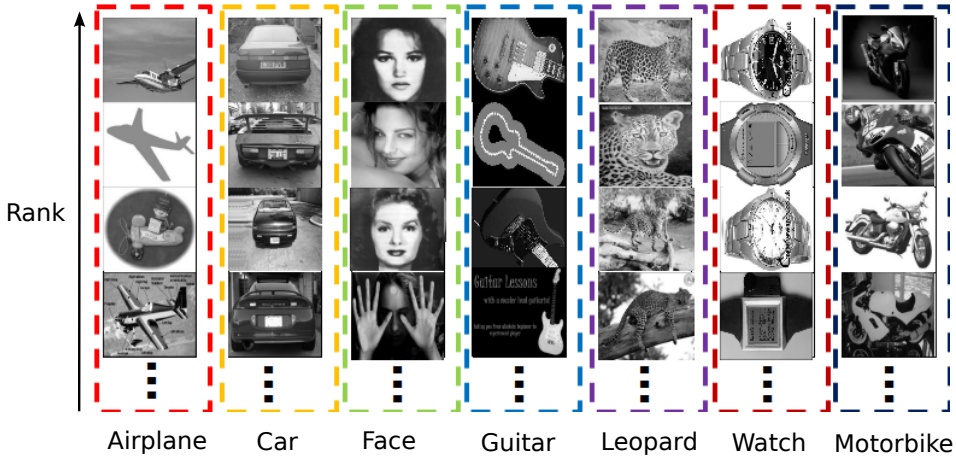


Figure 2.8: Illustration of the Google dataset [FFFPZ05].

2.6.3 Image Re-Ranking Using Superbag MIL

As the last experiment, we apply the superbag enhanced MIL framework to the problem of web image re-ranking. Recently, several groups proposed MIL as a ranking framework [LDXT11, VG08], that is particularly suitable for the re-ranking of web image search results.

The Google dataset (Fig. 2.8) was proposed originally by [FFFPZ05] to enhance the learning of object categories from web image search results. The dataset consists of about 4000 images divided into 7 categories that have on average 600 images. Since images are taken from a text based search, only around 30% of images are with a "good" view of the desired class, 20% are "ok" views, whereas the remaining 50% of images are considered as "junk" images, as they are completely unrelated to the category. In order to apply MIL, the images need to be grouped into multiple bags beforehand. Positive bags are obtained by randomly sampling images that are returned as a search result for given category. If the group is large enough, it can be assumed that at least one image in a bag will be positive. Negative images are obtained by sampling only images from other categories. The seven categories used in the Google dataset are airplane, car (rear), face, leopard, motorbike, guitar and wrist watch.

In order to build a feature representation for images in the Google dataset, we densely sample features around edges at multiple scales. Extracted features are represented by the SIFT descriptor. The method for feature sampling is simpler than that described in [LDXT11, FFFPZ05], where four interest point detectors are used, i.e. Kadir&Brady operator, Harris-Hessian detector, difference of Gaussians and Edge-Laplace detector. SIFT descriptors are quantized into a codebook of 500 visual words. Each image is represented as a bag of words (BoW), i.e. a histogram of visual words in that image. We use mi-SVM as a baseline for re-ranking of the Google images and compare it to the superbag approach that is integrated with the mi-SVM algorithm. In both cases an SVM with RBF kernel is employed and the kernel bandwidth is set to $4/A$, where A is the mean squared distance between images. All bags are of the same size

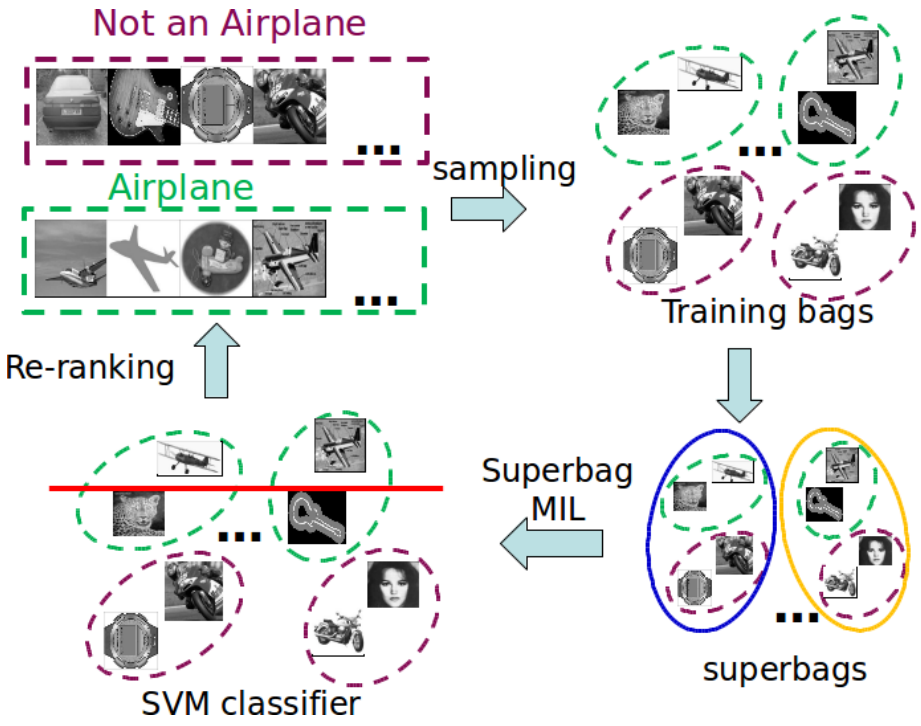


Figure 2.9: The pipeline of the image re-ranking based on superbag MIL.

of 15 images (Fig. 2.9). We use 5 superbags where each superbag consists of 40 bags. Following [LDXT11, VG08] the per-category precision at 15% recall is measured for performance evaluation. Both "good" and "ok" images are treated as positive samples in the experimental evaluation.

The results of the per-category and mean precision at 15% recall are provided in Tab. 2.3. Both, baseline mi-SVM and superbag enhanced mi-SVM consistently improve the results of Google's original web image search over all categories. We also see that superbag enhanced mi-SVM achieves significant improvement over the baseline mi-SVM on four out of seven categories. For airplane category this improvement is highest and equals approximately 25%. The gain of superbag enhanced mi-SVM in the mean precision over the baseline mi-SVM is 6.2%. The rest of the table shows the results of other state-of-the-art methods, i.e. WsMIL [VG08], Schroff et al. [SCZ11] and PMIL-CPB [LDXT11]. The superbag enhanced mi-SVM has the second best result of all compared methods in terms of mean precision, and the gain over Schroff et al.'s method and WsMIL is 11.3% and 5.8%, respectively. Only PMIL-CPB scores better by 5.1% than the superbag based approach. This is a consequence of a stronger constraint for positive bags, which requires that at least a portion of instances in a positive bag is positive, whereas we use a standard MIL constraint with at least one positive instance per positive bag. Besides, the concept of superbags can be also integrated with the PMIL-CPB method to increase its performance.

Table 2.3: Per-category precision and the mean precision (%) at 15% recall over 7 categories of the Google dataset. The abbreviations are for the category names: Airplane (A), Cars-rear (C), Face (F), Guitar (G), Leopard (L), Motorbike (M) and Wrist-watch (W).

	A	C	F	G	L	M	W	Mean
Google [FFFPZ05]	70.0	69.5	43.8	56.6	66.1	72.5	88.9	66.8
mi-SVM [ATH03]	50.7	64.1	86.7	84.3	64.9	83.5	93.0	75.3
WsMIL [VG08]	100	81	57	52	66	79	95	75.7
Schroff [SCZ11]	58.5	-	-	70.0	49.6	74.8	98.1	70.2
PMIL-CPB [LDXT11]	100	75.3	89.9	82.7	86.1	76.6	95.7	86.6
Superbags	76.1	71.9	83.0	82.7	78.7	80.5	97.6	81.5

2.7 Discussion

In this chapter a fundamental issue of widely used multiple-instance learning is addressed. In the underlying optimization algorithm, training of classifier and inference of latent instance labels are iteratively performed on the same training samples. This leads to overfitting and increases the variance of the label predictions. To resolve these issues, the concept of superbags has been introduced, which effectively decouples the inference and learning processes by performing them on different superbags. Experiments on standard MIL datasets show that the method consistently improves several widely used approaches for multiple-instance learning if the superbags are integrated into the optimization routine.

Chapter 3

Action Recognition by Sequential Multiple-instance Learning

Action recognition in videos is an important aspect of video understanding with wide applications ranging from surveillance and security to web-based video sharing (e.g. YouTube and Google videos). However, it is a difficult problem because videos are usually unconstrained and instances of action classes exhibit a large intra-class variability, there is a camera motion, background clutter, occlusion and change of illumination. Besides, action class instances typically correspond only to a subsequence of a video, while the rest of the video is usually the clutter. Detection of actions in videos is hampered by the lack of annotation for the action subsequences in training videos. In this chapter we describe a method based on multiple-instance learning that jointly discovers latent action instances (subsequences) in training videos and trains the action classifier on them. The multiple-instance learning method is extended with a sequential approach that results in a more robust selection of latent action subsequences and classifier training.

3.1 Outline of the Approach

Due to the laborious and costly nature of the video annotation process, video datasets are only weakly annotated. Action label is assigned to a video, if an instance of the respective action class occurs somewhere in video. However, information about the part of the video that corresponds to the action class is not provided. Consequently, action recognition approaches so far trained a classifier on the full video sequence representation to predict an action class of a video. Such methods are typically based on the bag-of-features (BoF) representation [Lap05, WKSCL11], i.e. a histogram of local spatio-temporal descriptors from the whole video sequence is used by discriminative classifier to predict the action label of the whole video.

As instances of an action class typically occur only in parts of videos, training a subsequence classifier that can detect and recognize which part of video corresponds to the

action can improve the understanding of the whole video. The subsequence classifier tests all subsequences of a video and selects the subsequence with the maximal classification score. The classification scores of the subsequence classifier and the full sequence classifier are then fused into a final action classification score. However, training the subsequence classifier *automatically* from the weakly labeled training videos whose subsequence labels are not provided is cumbersome. Multiple-instance learning based method is proposed to solve the following two problems jointly: i) find the action subsequences in training videos, and ii) train the subsequence classifier using the inferred action subsequences. Two tasks are cast in the framework of multiple-instance learning (MIL), so that video sequences become *bags*, and subsequences in a video become *instances* of a bag with latent action labels. MIL updates the subsequence classifier from subsequences detected in the previous round, which is used then to predict action labels for subsequences in the next round of MIL. In baseline MIL, subsequence length is fixed and only one subsequence (witness) is selected per training video. To improve the robustness of the MIL method, a sequential MIL approach is formulated that selects in each training video a number of instances of longer duration, and then sequentially reduces their number and length until only one subsequence remains as a representative of the action class. The subsequence classifier is fused with the full sequence classifier and tested on the difficult Hollywood2 benchmark set where a significant gain over the baseline is observed. Moreover, a favorable performance of the subsequence classifier not only in recognition of actions, but also in their detection in videos is evidenced on two categories of the Hollywood2 dataset.

3.2 Related Work

Local spatio-temporal features. Action recognition in videos has received lots of attention in the computer vision community over the last decade. A large body of literature on action recognition in videos is addressing the question of feature detection and description in videos. Laptev [Lap05] proposed spatiotemporal interest points obtained by the Harris detector that has been extended to the spatiotemporal domain. A novel cornerness measure that combines the Gaussian filter in space and Gabor filter in time is proposed by Dollár *et al.*[DRCB05]. Willems *et al.*[WTG08] proposed to detect spatiotemporal interest points at places where the determinant of the spatiotemporal Hessian matrix is maximal. Recently, Wang *et al.*[WMG09] showed that dense sampling of feature points has better performance than interest point detectors on the challenging video datasets. These state-of-the-art features are described in the following section.

Bag-of-Features representation. Best results on the challenging action recognition datasets such as Hollywood2 have been obtained by building the bag-of-features (BoF) representation of the whole video [SLC04a, WKSCL11]. The video is then classified with the kernel support vector machine (SVM) using normalized histograms of visual words. Recently, the BoF representation is extended in the spatio-temporal domain. Laptev *et al.*[LMSR08] proposed to concatenate BoF representations of the subvolumes

defined on a spatio-temporal grid in video. However, the grid is fixed and does not adapt to the action contained by the video.

Object-centric approaches. A human-centric approach [KMSZ10] detects and localizes human actions in challenging videos using the generic human detector and tracker, and the actions are detected only within the discovered human tracks. However, the method relies on a generic human detector and tracker that is trained on an external dataset which is typically not provided. Ommer *et al.*[OMB09] develop a generic action recognition system that combines compositional object segmentation and tracking.

Structural approaches. Gaidon *et al.*[GHS11] propose a model that uses a sequence of atomic action units, termed actoms, to represents the temporal structure of the action as a sequence of histograms of actom-anchored visual features. The actom model is trained from the actoms-annotated video clips. Niebles *et al.*[NwCFf10] propose to model the complex activities as temporal compositions of motion segments. They train a discriminative model to find a temporal decomposition of a complex activity. Lan *et al.*[LWM11] develop an algorithm for action recognition and localization in videos that uses a figure-centric visual word representation. Their model is trained on videos that are annotated with action labels and bounding boxes around the people performing the action. Hoai and De la Torre [HlT12] propose a max-margin framework for training the early event detector to recognize partial events. Their method is based on Structured SVM, that is extended to accommodate sequential data. However, training of the early event detector is supervised and requires full video annotation.

3.3 Dense Trajectory Features

In this section we explain the extraction of dense trajectories and computation of descriptors related to the trajectories. The exposition is based on the original paper of Wang et al. [WMG09]. Dense sampling of feature points is performed separately for each level of the multi-resolution image pyramid using the grid of points with spacing W pixels. All sampled feature points are then tracked throughout the video (Fig. 3.3). In homogeneous regions of video there is no structure, so feature points that belong to these regions cannot be reliably tracked. Therefore, all feature points residing in these regions need to be removed. The dense optical flow field is computed between each two successive frames I_t and I_{t+1}. A tracked point P_t is smoothed by the median filter, that is more resilient to noise than for example the Gaussian filter. The median filter also gives better estimate of trajectory points in the vicinity of motion edges. Computing optical flow is a costly operation, but once it is done, feature points can be densely tracked at no extra cost. Additional benefit of using a dense optical flow is the smoothness that the optical flow provides, that results in a stable tracking of fast and abrupt motion.

The feature points from subsequent frames are linked by the virtue of displacement vectors, thereby creating the trajectories. However, if there is no feature point in the vicinity of a predicted point location, tracking is discontinued and a new feature point is sampled and tracked. In this way, a well-balanced coverage of the feature points and

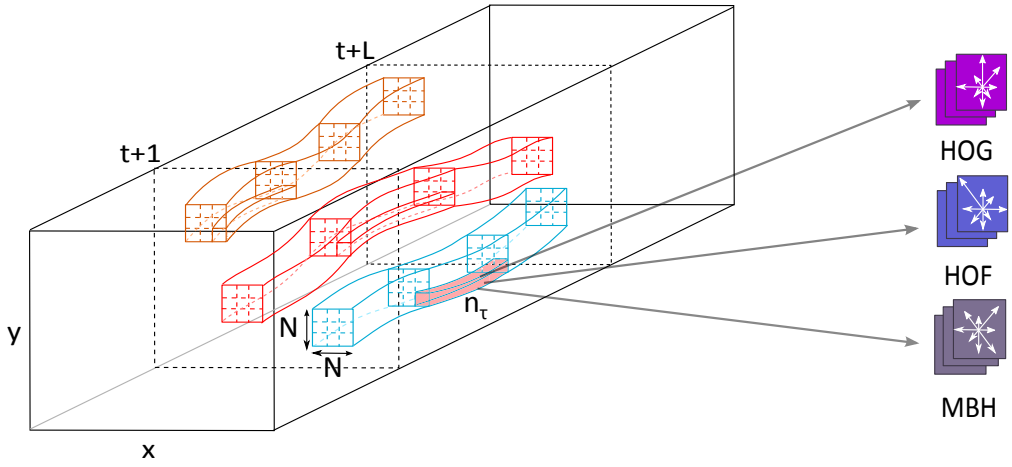
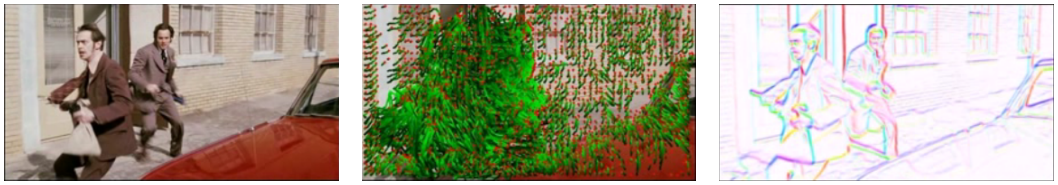


Figure 3.1: Dense trajectories in video are extracted at several scales and they all have a fixed length L. Local spatio-temporal descriptors are computed along the trajectories. The size of the spatio-temporal tube is $N \times N \times L$, and each tube consists of $n_\sigma \times n_\sigma \times n_\tau$ cells. Beside trajectory shape, descriptors also encode the histograms of gradient and flow (HOG and HOF), as well as motion boundary histograms (MBHx and MBHy).

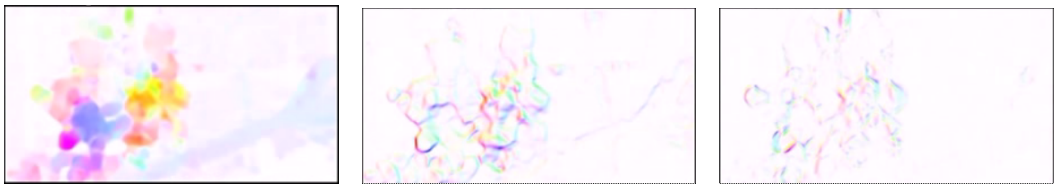
trajectories is ensured. Since static trajectories do not carry relevant motion information, they are removed in the post-processing phase, as well as those trajectories that have a very large displacement between two successive frames.

The shape of each trajectory is described as a sequence of displacement vectors. All trajectories have the same length L, so the shape descriptor ultimately yields a $2L-2$-dimensional descriptor vector. Displacement vectors in the trajectory shape descriptor are then normalized. Each trajectory is also described by other descriptors that are used to represent various aspects of the local spatio-temporal volume around the feature trajectory. These descriptors are: HOG (histograms of oriented gradients), HOF (histograms of optical flow) and MBH (motion boundary histograms). HOG descriptor, which describes the static appearance, and HOF, which focuses on the local motion patterns, both perform well when they are applied separately as features in the BoF model for action recognition. MBH descriptor calculates the derivatives separately for horizontal and vertical optical flow components, which then renders it quite successful at describing the prominent motion details in a video.

To take the dynamics of the video into account, we combine the spatio-temporal component with a trajectory. This results in a space-time trajectory volume, which contains relevant motion information. The trajectory volume size is $N \times N$ pixels and L frames in length. Furthemore, we need to get structure information and we do that by forming a spatiotemporal grid, which divides the trajectory volume into $n_\sigma \times n_\sigma \times n_\tau$ cells. We compute HOG, HOF or MBH descriptors for each cell of the grid, and the resulting descriptor is obtained by concatenating the computed descriptors. Orientations for HOG and HOF are quantized into 8 bins, with an additional bin for HOF to cover pixels with very low optical flow magnitudes. Further on, they are L_2-normalized.



(a) video (b) dense trajectories (c) HOG descriptor



(d) HOF descriptor (e) MBHx descriptor (f) MBHy descriptor

Figure 3.2: Descriptors extracted from dense trajectories. The orientation and magnitude of vectors in HOG, HOF and MBH descriptors are represented by hue and saturation values of pixel's color.

Optical flow carries a lot of relevant information necessary for action recognition, but also contains some spurious information, i.e. background or camera motion. MBH descriptor alleviates some of the deficiencies of optical flow. As opposed to optical flow, which describes the absolute motion between two consecutive frames, MBH represents the relative motion between the frames. It calculates the difference between the foreground and background motion, which suppresses the camera motion, and consequently makes it more suitable for action recognition. Orientation of the MBH descriptor (both vertical MBHy and horizontal MBHx components) is quantized into 8-bin histograms, and both histogram vectors are then L_2-normalized. MBH descriptor significantly outperforms HOF, which comes at no surprise since it efficiently removes the camera motion component from the action representation. Once the optical flow is computed and dense trajectories are extracted, all descriptors can reuse the trajectories which makes the feature extraction process quite efficient.

3.4 Bag of Features Model

An excessive number of local spatio-temporal features in a single video makes a direct comparison of videos based on local features computationally not feasible. Therefore, it is necessary to aggregate information from local spatio-temporal features into an intermediate representation of fixed length so that videos can be efficiently compared. One of the most popular aggregation technique is the Bag of Features (BoF) [CDF+04]. Before local features are aggregated, they first need to be encoded using a codebook that has a fixed number of visual words. Codewords are typically obtained by unsupervised learning in a randomly sampled set of local features from training videos (Fig. 3.4).

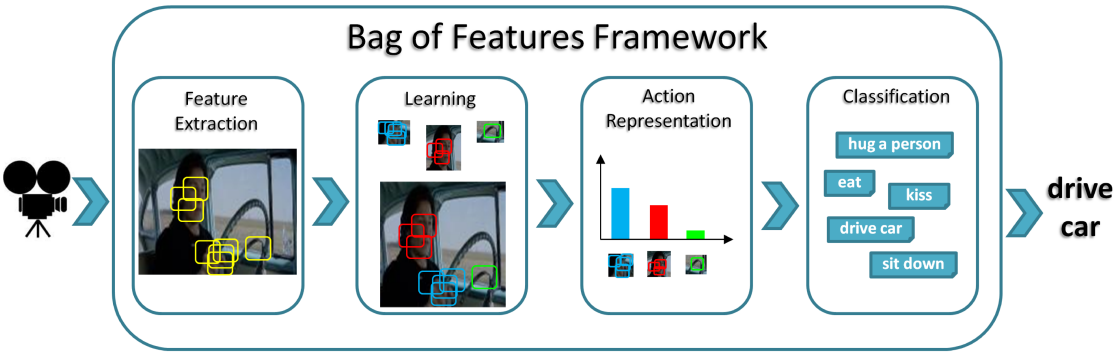


Figure 3.3: Standard Bag-of-Features framework for human action recognition.

The samples training features are clustered into k codewords using k-means algorithm. Local spatio-temporal descriptors are usually assigned to their nearest codewords. BoF representation of a video is obtained by counting how many local features are assigned to each of the k codewords. Final video representation is obtained by concatenating normalized BoF histogram vectors for each local spatio-temporal descriptor (trajectory, HOG, HOF, MBHx and MBHy).

Videos are classified into action categories using SVM classifier that combines different descriptors by a kernel function that couples radial basis function with χ^2-distances between BoF vectors:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sum_k \gamma_k d(\mathbf{x}_{i,k}, \mathbf{x}_{j,k})). \tag{3.1}$$

$d(\cdot,\cdot)$ denotes a χ^2-distance function, whereas $\mathbf{x}_{i,k}$ and $\mathbf{x}_{j,k}$ denote BoF vectors for the k-th descriptor. The normalization constant γ_k is calculated as the inverse of the mean χ^2-distances between all training BoF vectors for the k-th descriptor.

3.5 Sequential MIL for Action Recognition

To recognize an action in video, we first need to find a subsequence that corresponds to the potential action in video and then classify it with a subsequence classifier. However, to train the subsequence classifier we need a set of training subsequences that correspond to the action. As the subsequences in training videos are not annotated, we need to solve two problems jointly: i) training of the subsequence classifier, and ii) detection of subsequences from the training videos that correspond to the action. We use the multiple-instance learning (MIL) to solve these two problems jointly.

In previous chapter we have seen that multiple-instance learning consists of a number of training *bags* B_I, and each bag is associated with a label $Y_I \in \{-1, +1\}$. Bags consist of a number of *instances* $B_I = \{\mathbf{x}_i : i \in I\}$ whose labels $y_i \in \{-1, +1\}$ are not

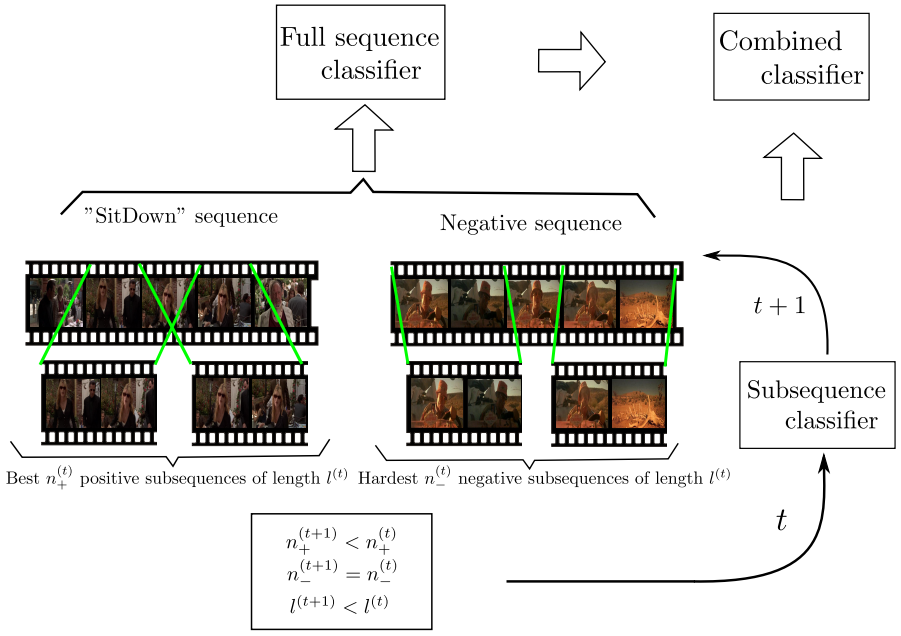


Figure 3.4: An illustration of the proposed joint action detection and classification in videos by sequential MIL. Subsequence classifier is updated in each iteration t using the best $n_+^{(t)}$ subsequences from a positive video and the hardest $n_-^{(t)}$ subsequences from a negative video. The number of positive subsequences and their length decreases during training. The number of negative subsequences $n_-^{(t)}$ is fixed.

provided. Bag B_I has a positive label $Y_I = +1$ if *at least* one of its instances is positive, i.e. $\exists i \in I : y_i = +1$. If all instances in a bag are negative, $\forall i \in I : y_i = -1$, the bag itself is labeled as negative, $Y_I = -1$. The goal of MIL is now to train an instance-level classifier from training instances whose labels are not provided and thus have to be inferred during training.

In the video subsequence classification problem, bag B_I corresponds to the full video sequence, and instances $\mathbf{x}_i, i \in I$ are all subsequences of the training video B_I. For each positive training video $Y_I = +1$, we are looking for a subsequence $s_I \in I$ that is an instance of the action class in that video. This can be formulated as the multiple-instance learning with latent instance selection (MI-SVM approach, Andrews *et al.*[ATH03]) that corresponds to the optimization in Eq. 2.33.

A standard solution to this optimization problem is to perform the following two steps iteratively: i) select the witness $s_I \in I$ in each positive bag B_I that corresponds to an instance $\mathbf{x}_{s_I}$ with the maximal score of the instance classifier, and ii) re-train the instance classifier (w, b) using the instances selected in the previous step. The described MI-SVM approach yields sometimes a bad local optimum of the MIL problem.

We improve the robustness of the MIL training for the task of subsequence classification in two ways. First, we select longer subsequences from positive videos in early rounds of MIL and then slowly decrease their length in later rounds. This allows us to gradually discover the action related part of the sequence when duration of the ac-

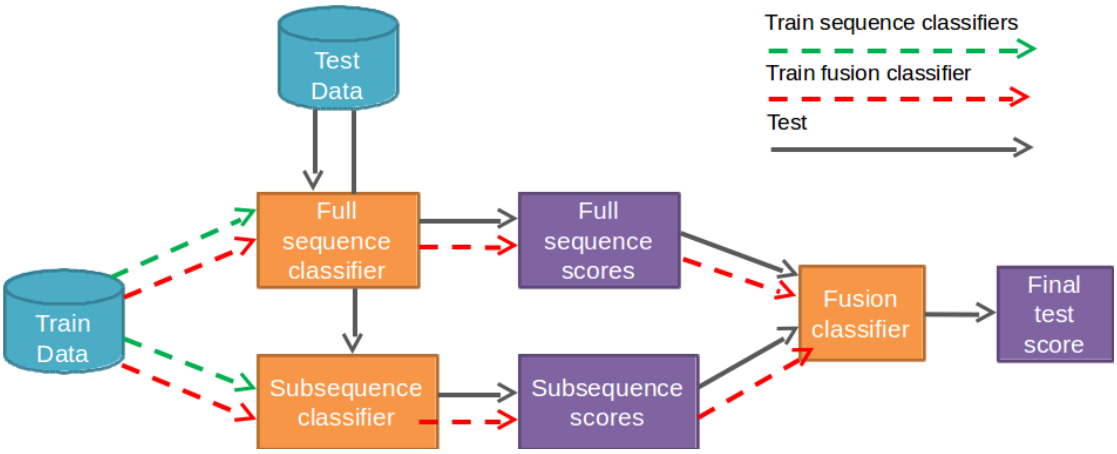


Figure 3.5: Training and testing pipelines of the proposed action detection and recognition system.

tion is short. Subsequences from the negative bags have the same length as the positive subsequences, and thus they are also decreased successively during the MIL training. The length of subsequences used during MIL training is hence given as a monotonically decreasing sequence,

$$l^{(0)} > ... > l^{(t)} > ... > l^{(T)} \neq 0. \tag{3.2}$$

Secondly, we increase the recall rate of the positive instances by selecting initially several instances per positive bag and then slowly reducing their number until only one instance is selected per bag. Thus, the number of positive instances that is selected by the MIL algorithm is a monotonically decreasing sequence,

$$n_+^{(0)} > ... > n_+^{(t)} > ... > n_+^{(T)} \neq 0. \tag{3.3}$$

For negative videos, we can train the SVM model with all subsequences that are extracted from the video. As there is a large number of subsequences in a video, we actually work only with a small number $n_-^{(t)}$ of hard negative subsequences per bag. The hard negative subsequences are those that have the least negative score produced by the subsequence classifier, and thus they define the margin in the max-margin classification setup. The number of hard negative instances per bag is kept fixed throughout the MIL training,

$$n_-^{(t)} = const., \ 0 \leq t \leq T. \tag{3.4}$$

In short, these are the steps of the algorithm for learning the subsequence classifier:

1. Sample randomly a number of long subsequences from positive and negative video sequences and train the initial subsequence classifier from them.

2. Use the subsequence classifier to select the best $n_+^{(t)}$ positive subsequences of length $l^{(t)}$ from each positive video.

Categories	Baseline classifier	Fusion classifier
AnswerPhone	28.76%	31.40%
DriveCar	89.13%	89.15%
Eat	61.78%	66.42%
FightPerson	80.47%	82.89%
GetOutCar	49.72%	53.91%
HandShake	30.65%	30.75%
HugPerson	51.25%	55.90%
Kiss	64.04%	65.72%
Run	82.30%	82.71%
SitDown	63.20%	66.25%
SitUp	20.93%	21.46%
StandUp	66.82%	71.09%
mAP	57.42%	59.80%

Table 3.1: The comparison of the full sequence classifier (baseline) and the combination of the full sequence and subsequence classifiers on all action classes of the Hollywood2 dataset.

3. Use the subsequence classifier to select the hardest $n_{-}^{(t)}$ negative subsequences of length $l^{(t)}$ from each negative video.

4. Re-train the subsequence classifier from the selected positive and negative subsequences.

5. Repeat steps 2. - 4. T times while reducing the length of all subsequences and the number of positive subsequences, and also keeping the number of negative subsequences fixed.

We use the standard bag-of-features representation (BoF) for the classification of videos and their subsequences. The features are obtained as dense trajectories using the method of Wang *et al.*[WKSCL11]. Each feature is encoded using the trajectory, HoG, HoF and MBH descriptors. The non-linear support vector machine is used to predict the action classes based on the BoF representation. We use the histogram intersection kernel with non-linear SVM classifier because of good classification performance and the existence of fast computation method for this kernel (Maji *et al.*[MBM08]).

To classify a novel video, we combine the scores of the full sequence and the subsequence classifiers. The full sequence classification score is computed from the BoF representation of the whole video. The classification score of the subsequence classifier is obtained by applying the subsequence classifier on all subsequences of the same length as the final-round subsequences of the MIL training. We take the maximal subsequence classification score and combine it with the full sequence classification score to yield the final score for the video. The fusion is performed by a nonlinear support vector machine that uses the RBF kernel whose bandwidth is optimized on the validation set. The sketch of the proposed approach is shown in Fig. 3.5.

(a) Get out of car (b) Run (c) Eat

(d) Sit up (e) Kiss (f) Fight person

(g) Drive a car (h) Handshake (i) Hug a person

(j) Answer a phone (k) Stand up (l) Sit down

Figure 3.6: Action categories in the Hollywood2 dataset.

Category	Overlap
SitDown	48.6%
GetOutOfCar	34.4%

Table 3.2: The detection performance of the subsequence classifier on two Hollywood2 categories (SitDown and GetOutOfCar) for which we created the ground truth. The detection performance expressed as the PASCAL's overlap (intersection over union) score.

Methods	Accuracy
Wang *et al.*(2009) [WUK+09]	47.7%
Taylor *et al.*(2010) [TFLB10]	46.6%
Ullah *et al.*(2010) [UPL10]	53.2%
Gilbert *et al.*(2011) [GIB11]	50.9%
Le *et al.*(2011) [LZYN11]	53.3%
Wang *et al.*(2011) [WKSCL11]	58.2%
Wang *et al.*(2013) [WKSL13]	59.9%
Fusion classifier	59.8%

Table 3.3: Comparison of the fusion classifier (full sequence + subsequence classifier) to the state-of-the-art methods on the Hollywood2 dataset.

3.6 Experimental Results

We evaluate the sequential MIL approach on the state-of-the-art Hollywood2 benchmark set for the action recognition (Fig. 3.6). The Hollywood2 dataset consists of 12 action categories: answering the phone, driving a car, eating, fighting, getting out of a car, hand shaking, hugging, kissing, running, sitting down, sitting up, and standing up. The dataset contains 1707 videos that are divided into the training set (823 video clips) and the test set (884 video clips). All video clips are taken from 69 Hollywood movies and they contain realistic, unconstrained human actions with large amount of camera motion. Video sequences vary in length from $5-25$ seconds, and most of the actions are only $1-2$ seconds long. Therefore, the length of the action subsequence is several times shorter than the length of the whole video. This motivates the subsequence classification approach which automatically finds and classifies the part of the video that corresponds to the action. Video sequences in train and test set are taken from different movies. We follow the evaluation protocol of Laptev *et al.*[LMSR08] and train the one-against-all classifier for each action category. Precision-recall (PR) curves are calculated from the classification scores, and per-class performance is based on the average precision (AP) values computed from the corresponding PR plots. Overall performance is reported as the mean average precision over all classes (mAP) and it is used for the comparison with the state-of-the-art.

The full sequence classifier is used as the baseline in our comparison in Table 3.1. We use the same parameters as in the paper of Wang *et al.*[WKSCL11], except for the choice of the kernel function. We use the intersection kernel because there exists a fast method for its computation. Wang *et al.*use χ^2 kernel function that yields 0.8% higher performance, but at the price of slower computation of the kernel matrix (about 3 times longer training and test time).

Table 3.1 shows the comparison of the full sequence classifier (baseline) and the fusion classifier that aggregates the scores of the full sequence and subsequence classifiers. Finding a subsequence in a video that represents the action and classifying it with the subsequence classifier yields an improvement over the baseline of $1-5\%$ in average

Figure 3.7: Detection of the "GetOutOfCar" subsequence for the test video $\#234$. Overlap of detected subsequence and the ground truth is 96.7%. The length of the sequence is 134 frames, and the subsequence is detected from 21th to 80th frame.

Figure 3.8: Detection of the "SitDown" subsequence for the test video $\#682$. Overlap of detected subsequence and the ground truth is to 94.9%. The length of the sequence is 169 frames, and the "GetOutOfCar" subsequence is detected from 81th to 120th frame.

Figure 3.9: Wrong detection of the "SitDown" subsequence in the test video $\#352$. The length of the sequence is 438 frames, and the ground truth covers from 90th to 110th frame.

precision for most of the Hollywood2 classes. Largest improvements are for the classes StandUp, HugPerson and Eat, where the gain is almost 5%. On the other hand, classes such as DriveCar or Run give only a marginal improvement over the baseline. After averaging all the classes, the fusion of the subsequence and full sequence classifiers yields 2.5% better performance than the baseline (59.8% vs. 57.4%).

Table 3.3 shows the comparison of the proposed method and the state-of-the-art methods on Hollywood2 dataset. We see that the fusion of subsequence and full sequence classifier outperforms almost all of the state-of-the-art results, in spite of using slightly inferior kernel function than other methods (intersection kernel vs. χ^2). The most recent result on the Hollywood2 dataset (Wang et al. 2013) combines the BoF representation with spatiotemporal pyramids and yields 59.9% mAP. The performance of the proposed method is only 0.1% weaker than that result, although the proposed method does not use strong, computationally expensive, pyramidal features.

In Table 3.2 we show the results of detection performance for the subsequence classifier. We evaluate this performance using the PASCAL overlap (intersection over union) criterion on two categories of the Hollywood2 dataset for which we have man-

ually labeled the ground truth subsequences. The overlap of detected subsequence and the ground truth for SitDown class is 48.6%, whereas for GetOutOfCar the overlap is 34.4%. We consider these as very good results, because the training of the subsequence detector is automatic.

Finally in Fig. 3.7 - 3.9 we give some qualitative results of subsequence detection for two action classes of Hollywood2. Fig. 3.7 and 3.8 illustrate the successful subsequence detection. In Fig. 3.9 detection failed, most probably because of the large camera zoom on the main actor's face, making his body motion hardly visible while he is performing the "SitDown" action.

3.7 Discussion

In this chapter we described a method that learns an action classifier from automatically discovered subsequences that represent action instances in training videos. As subsequences are not annotated, the proposed method jointly trains the subsequence classifier and labels the action subsequences in training videos. Multiple-instance learning is used to find subsequences that correspond to the action which then allows to train the subsequence classifier. To obtain a robust solution to the MIL problem, a sequential algorithm is proposed that consecutively decreases the number of inferred action subsequences per video and decreases their length until only one action subsequence remains in a video. The fusion of the automatically trained subsequence classifier and the full sequence classifier is evaluated on the challenging Hollywood2 dataset where it yields a significant performance improvement over the baseline classifier. We also examined the action detection performance of the subsequence classifier on two categories of Hollywood2 dataset and notice that it achieves promising results.

Chapter 4

Sequential Video Parsing for Abnormality Detection

We have seen so far that video understanding assumes the ability of a computer vision system to accurately detect and recognize objects and actions in video sequences. The challenge becomes even greater as soon as atypical objects or unusual behavior appear in a scene, that then need to be detected. In this chapter we describe a latent structured model for video parsing, that forgos the standard abnormality detection approach borrowed from object detection field that aims at finding an abnormal instance in a scene independently of other objects. In contrast to that, this chapter proposes a method for sequential video parsing which jointly detects all instances in a single video frame and recognizes abnormalities among them using the context of other object instances in the scene and from training videos.

4.1 Outline of the Appraoch

In many applications, ranging from industrial product inspection to visual surveillance, discovery of abnormal instances in videos is the paramount goal. Although detection and classification of normal object or action categories already pose a great hurdle because of their large within-class variability, in case of abnormality detection there is an additional problem of endlessly many ways in which an object can appear in unusual context (atypical object) or behave abnormally (irregular activity). This renders the task of learning a model for everything that is anomalous practically unfeasible. Moreover, training videos for abnormality detection contain only normal visual patterns, thus rendering a discriminative approach to localization of irregularities in video futile. The following question then arises: how can abnormalities be detected, when it is not known *how they look like*? This fundamental problem has been tackled so far only in the context of independent classification of individual local patches or regions in video frames [BI05, XG05, ZSV04]. However, as we have already explained, these local and independent decisions about abnormalities in videos are ill-posed.

Figure 4.1: The pipeline of the sequential video parsing. To parse a single video frame, a shortlist of object hypotheses is created, such that they are sufficient for explaining a foreground mask. By probabilistic inference, video parsing selects a subset of hypotheses that are necessary for explaining the foreground, and simultaneously matches them to normal training samples. Probability of abnormality is eventually calculated for all selected hypotheses and foreground pixels.

This chapter proposes a sequential video parsing method that aims at jointly detecting and recognizing instances in a single frame based on their mutual context and normal object model learned from the training videos. Lots of videos are nowadays recorded by stationary cameras (e.g. in visual surveillance) in which case efficient background subtraction algorithms [SG99, WGR$^+$09] offer foreground segmentation mask for each video frame. Videos parsing method sets off by constructing a shortlist of object hypotheses that are sufficient for explaining foreground pixels of a video frame. In video parsing object hypotheses need to be laid out all over the frame for the foreground mask to be completely covered, but at the same time protrusion into the background be made as small as possible. To tackle this challenge, all hypotheses are jointly placed within the scene and their spatial configuration determined, so that for each hypothesis the best match in the set of normal training samples is found. Every hypothesis that is needed to explain the foreground but which cannot be explained by a sample from the set of normal training samples found in training videos is considered as abnormal. Consequently, by parsing the scene and *jointly* inferring all necessary object hypotheses, all abnormalities are *indirectly* found in video.

Sequential video parsing is formulated as a two-stage approach: In the first stage a shortlist of object hypotheses in a single frame is computed that has a low false-negative and high false-positive rate, i.e. a superset of all hypotheses is found that might be needed later on for parsing a frame. Background subtraction algorithm is used to discard all hypotheses in the background, and a discriminative background classifier keeps only those hypotheses for which it is highly unlikely that they belong to the background. Based on the shortlist of candidate hypotheses, the problem of parsing a video frame and explaining the foreground mask with object hypotheses becomes a discrete optimization problem. A subset of the initial hypotheses that are required for covering all the foreground needs to be discovered. Hence, for all hypotheses it is necessary to jointly infer their presence and correspondence to the exemplars of normal objects collected in the training videos. Correspondences between hypotheses and normal object exemplars are set up to capture both the object appearance and the motion that represents the object action. Due to our probabilistic approach, not only object hypotheses are labeled as abnormal/normal, but we also infer a per-pixel abnormality probability

which allows to segment abnormal objects in the scene without actually seeing what the abnormal objects look like or how do they behave in the training set. The pipeline of the sequential video parsing is shown in Fig. 4.1.

4.2 Related work

We discuss here the previous work on abnormality detection in videos. The related problem of object detection and tracking in crowded scenes [BC06, ZNW08] aims at recognizing and tracking objects of a *known* class in a scene, whereas our goal is to detect abnormalities, all of them being instances of an *unknown* class. Therefore, object recognition and tracking are beyond the scope of this thesis and the details on these topics can be found in [OMB09].

General scene parsing. Previous approaches related to scene parsing are intended for interpretation of static images. These approaches construct a parametric scene [TCYZ05, AT08] or object models [KY09, FL07, MO12] or a non-parametric exemplar-based representation for objects [LYT09, ME09]. In contrast to these methods, we are not provided with any training samples for the abnormalities we are searching for but we can leverage a foreground/background segregation.

Abnormality Detection using Temporal Dynamics. Kratz and Nishino [KN09] develop a statistical model of local motion patterns in very crowded scenes to find abnormalities as local volumes with large motion variation. Benezeth et al. [BJS11] use low-level motion features to learn the co-occurrence matrix of normal object behavior in the scene, and apply a Markov random field to find deviating behaviors. The approach of Adam et al. [ARSR08] focuses on individual activities occurring only in preselected parts of a scene. Kim and Grauman [KG09] detect abnormalities using a Markov random field that adapts to abnormal activities in videos. Loy et al. [LXG10] use active learning methodology to integrate human feedback into the detection of abnormal events and behaviors. Unsupervised topic models are used for detection of abnormal behaviors in [WMG09, HGX09]. Hospedales et al. [HLGX11] propose a semi-supervised multi-class topic model to classify and localize the subtle behavior in cluttered videos.

Abnormality Detection using Appearance and Temporal Dynamics. A huge body of work on abnormality detection relies on the extraction of semi-local features from video [Low04, SLC04b, JYTK11, JRL13, SSP+11], that are then be used to train a normalcy model. Abnormalities are detected if the normalcy model does not fit the data. Some approaches [DH04, ZSV04] are based on manually specifying constraints that define the condition of normalcy, whereas other methods [XG05, XG08, BGS08, WMG07, ZNRC13, YGC13] learn the normalcy model directly from data in unsupervised way. Cong et al. [CYL11] use sparse reconstruction cost implemented on a normal dictionary of local spatio-temporal patches to detect local and global abnormalities. Saligrama et al. [SC12] propose optimal decision rules for detecting local spatio-temporal abnormalities. An efficient sparse coding method that achieves decent performance in abnormality detection is proposed by Lu et al. [LSJ13]. Mahade-

van et. al [MLBV10] detect unusual objects in crowded scenes by jointly modeling the dynamics and appearance with mixtures of dynamic textures. Li et al. [LMV13] use the mixture of dynamic textures at multiple scales to detect abnormalities in a conditional random field framework.

In this chapter, a sequential model for video parsing is proposed that exploits only spatial interactions between object hypotheses in a single frame. Instead of independently detecting abnormal local patches or regions in a video frame, sequential video parsing discovers abnormalities indirectly by establishing a set of hypotheses that provide a complete explanation of the foreground mask.

4.3 Background Subtraction

Many applications aim at detecting moving objects in video, so that they can later be analyzed and classified into different object categories. If video camera is placed at a fixed location in the scene, foreground is segmented by subtracting the background image. However, to build a robust background model, it is necessary to deal with many difficulties in practice, such as camera shaking and jitter, time of the day and light changes, foreground aperture, background motion, waking/sleeping foreground objects and shadows, to name a few. Methods for background modeling can be pixel-wise and frame-wise. Pixel-wise methods for background modeling assume that pixels of a frame are statistically independent and build a separate statistical model in each of them. The popular pixel-wise background subtraction model of Stauffer and Grimson [SG99] learns a mixture of Gaussian (MoG) model in each pixel and adaptively updates the parameters of the mixture over time. On the other hand, frame-wise models represent background pixels jointly as a vector in a low-dimensional vector space [WGR$^+$09].

4.3.1 Stauffer-Grimson (MoG) Model

MoG model for background subtraction was first proposed by Friedman and Russel [FR97] for modeling background in a traffic surveillance system. Their model consisted of three Gaussian components that represented the values of road, vehicle and shadows. They semantically labeled components of the MoG model using the following rule: darkest component is labeled as shadow, and of the remaining two component, one with larger variance is labeled as vehicle and other as road. Foreground pixels are found by comparing pixel value with each Gaussian component and obtaining the label (vehicle, road or shadow) of the closest Gaussian component. Pixel-wise MoG model proposed by Stauffer and Grimson [SG99] is a generalization of Friedman and Russel's model. A mixture of K Gaussians is fit to each pixel based on the history of pixel's recent values. Typically, each pixel j is represented by the intensity I_j^t in the grayscale or RGB color space. According to the MoG model, the probability of pixel value is given as a

weighted average of K multidimensional Gaussian distributions,

$$P(I_j^t) = \sum_{i=1}^{K} \omega_{i,t} \mathcal{N}(I_j^t; \mu_{i,t}, \Sigma_{i,t}). \tag{4.1}$$

All K Gaussian components are represented with the following parameters: $\omega_{i,t}$ is a weight associated with Gaussian i at time t, whose mean is $\mu_{i,t}$ and covariance $\Sigma_{i,t}$:

$$\mathcal{N}(I_j^t; \mu_{i,t}, \Sigma_{i,t}) = \frac{1}{(2\pi)^{d/2}|\Sigma_{i,t}|^{1/2}} \exp(-\frac{1}{2}(I_j^t - \mu_{i,t})^\top \Sigma_{i,t}^{-1}(I_j^t - \mu_{i,t})). \tag{4.2}$$

In Stauffer and Grimson's approach [SG99], covariance matrices $\Sigma_{i,t}$ in RGB color space are assumed to be diagonal and of the same variance for all color channels:

$$\Sigma_{i,t} = \sigma_{i,t}^2 \mathbf{I}. \tag{4.3}$$

The number K of Gaussian components in a mixture is determined by the multimodality of the background distribution, and is typically set to 3 to 5. Stauffer and Grimson use the online EM algorithm to update the weight, mean and covariance parameters of Gaussian components in the mixture. To find background components in the MoG distribution, Gaussian components are sorted according to the ratio $r_{i,t} = \frac{\omega_{i,t}}{\sigma_{i,t}}$ calculated for each Gaussian component. As background components occur more often than foreground, and their distribution has smaller variance than of the foreground, the ratio $r_{i,t}$ of a background Gaussian component is in general larger than of a foreground component. Therefore, first B Gaussian components whose total weight is greater than a threshold T are labeled as background components,

$$B = \underset{b}{\operatorname{argmin}}(\sum_{i=1}^{b} \omega_{i,t} \geq T). \tag{4.4}$$

When a new frame at time t arrives, pixels in the frame are compared to each component of the MoG distribution. A pixel value I_j^t matches the Gaussian component i of the MoG model if the Mahalanobis distance is less than a threshold:

$$(I_j^t - \mu_{i,t})^\top \Sigma_{i,t}^{-1}(I_j^t - \mu_{i,t}) \leq \text{threshold}. \tag{4.5}$$

If a match is found with one of B background components, pixel is labeled as background, otherwise it is classified as foreground. If there is a match with one of the K components, the weight of the matched Gaussian component is updated according to the learning rate α,

$$\omega_{i,t+1} = (1 - \alpha)\omega_{i,t} + \alpha. \tag{4.6}$$

The remaining parameters of the matched Gaussian (mean and covariance) are updated with a learning rate $\rho = \alpha \mathcal{N}(I_j^t; \mu_{i,t}, \Sigma_{i,t})$ that is proportional to the probability that a pixel value I_j^t matches a Gaussian whose mean is $\mu_{i,t}$ and covariance $\Sigma_{i,t}$,

$$\mu_{i,t+1} = (1 - \rho)\mu_{i,t} + \rho I_j^t, \tag{4.7}$$

$$\Sigma_{i,t+1} = = (1 - \rho)\Sigma_{i,t} + \rho(I_j^t - \mu_{i,t+1})^\top (I_j^t - \mu_{i,t+1}). \tag{4.8}$$

Unmatched Gaussian components in the mixture retain the same mean and covariance values as in the previous time instant, but their weights are dampened as follows,

$$\omega_{i,t+1} = (1 - \alpha)\omega_{i,t}. \tag{4.9}$$

If none of the K Gaussian components is matched to the current pixel value, the least probable Gaussian component is replaced with a new component of low prior weight, and large covariance and the mean equal to the last pixel value.

Despite a huge popularity of the Stauffer-Grimson method for background subtraction, in our experiments the algorithm could not provide a reliable foreground/background segmentation. Quick illumination changes, camera jitter and camera automatic iris correction could not be resolved by a pixel-wise model. Consequently, we decided to use frame-wise models that jointly model all pixels in a frame. A very efficient frame-wise method based on robust principled component analysis[WGR$^+$09] is used.

4.3.2 Robust Principle Component Analysis Model

For videos that are recorded by a static camera, background is constant or changes slowly over time, and thus can be effectively learned from previous frames. The resulting background model can be applied to detect foreground pixels in the video. The final foreground/background segmentation of a video frame is represented as a binary variable $f_j \in \{0, 1\}$ for all pixels j.

Background subtraction assumes that each frame I^t in a video can be decomposed into a background model and a sparse foreground. By stacking successive video frames, image data matrix $I = \left[I^{t-\tau} \cdots I^t\right]$ is obtained. Now, to find foreground pixels in video we need to search for the smallest rank background matrix B that could have produced the image data I with sparse foreground matrix F, $\|F\|_0 \leq k$. The problem can be converted into a Lagrangian dual form:

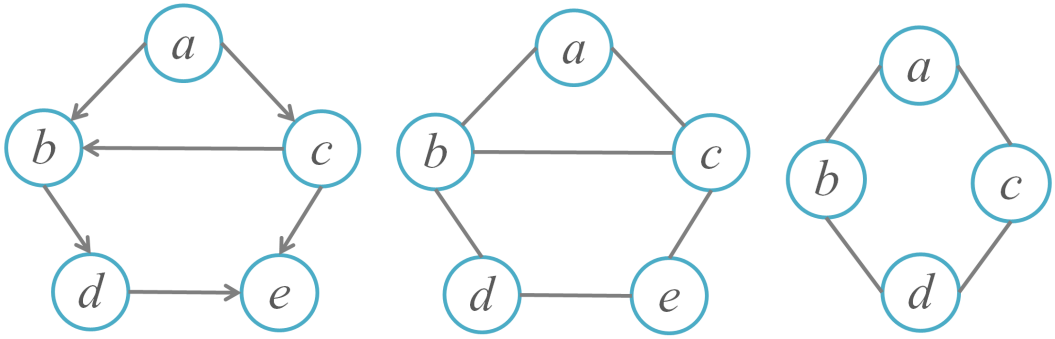
$$\{B, F\} = \underset{\tilde{B},\tilde{F}}{\operatorname{argmin}} \operatorname{rank}(\tilde{B}) + \gamma\|\tilde{F}\|_0, \text{ s.t. } \tilde{B} + \tilde{F} = I. \tag{4.10}$$

However, the problem in Eq. 4.10 is non-convex, and an efficient *exact* solution is not available. Therefore, Wright et al. [WGR$^+$09] proposed to solve the relaxed problem, where the ℓ_0-norm of matrix F is replaced by the ℓ_1-norm, and the rank of matrix B is replaced by the nuclear norm[1] $\|B\|_* = \sum_i \sigma_i(B)$. This yields to the following convex optimization program:

$$\{B, F\} = \underset{\tilde{B},\tilde{F}}{\operatorname{argmin}} \|\tilde{B}\|_* + \gamma\|\tilde{F}\|_1, \text{ s.t. } \tilde{B} + \tilde{F} = I. \tag{4.11}$$

Some recent results [WGR$^+$09] on ℓ_1-norm regularized sparse solution of underdetermined systems of linear equations, i.e. nuclear-norm regularized low-rank solution of matrix equations, show that Eq. 4.11 under some mild conditions leads to a perfect

[1] $\sigma_i(B)$ denotes the i-th singular value of matrix B.



(a) $p(a)p(c|a)p(b|a,c)p(d|b)p(e|c,d)$ (b) $\frac{1}{Z}\psi(a,b,c)\psi(b,d)\psi(c,e)\psi(d,e)$ (c) $\frac{1}{Z}\psi(a,b)\psi(b,d)\psi(a,c)\psi(c,d)$

Figure 4.2: a) A directed acyclic graph and b) an undirected graph. Factorization of distributions that correspond to the graphs are given under each graph. c) An example of an undirected graphical model whose conditional independence properties $b \perp\!\!\!\perp c|a,d$ and $a \perp\!\!\!\perp d|b,c$ cannot be represented by any directed graphical model (Bayesian network).

recovery of the sparse matrix $\bar{F}$ and low-rank matrix $\bar{B}$. Formally, this means that for almost every pair of low-rank matrix $\bar{B}$ and sparse matrix $\bar{F}$, the following holds

$$\{\bar{B},\bar{F}\} = \underset{\tilde{B},\tilde{F}}{\operatorname{argmin}} \|\tilde{B}\|_* + \lambda\|\tilde{F}\|_1, \text{ s.t. } \tilde{B}+\tilde{F} = \bar{B}+\bar{F}. \tag{4.12}$$

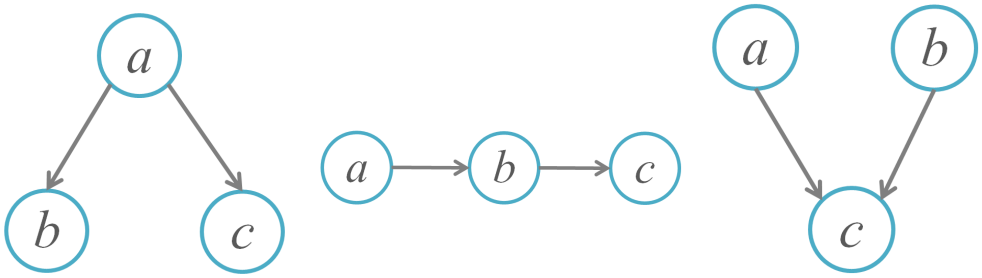
This means that for almost any data matrix $I = \bar{B} + \bar{F}$, its low-rank matrix component $\bar{B}$ and the sparse matrix component $\bar{F}$ can be exactly recovered by solving the convex program of Eq. 4.12.

4.4 Probabilistic Graphical Models

Probabilistic graphical models (PGM) is a modeling paradigm used for specifying properties of multidimensional probability distributions. PGM stands at the intersection of statistics and graph theory. The idea behind PGM is to represent independence properties of a multidimensional probability distribution in a form of a graph. Independence assertions implied by the graphical model are used to simplify the statistical computations performed with high-dimensional probability distributions. In other words, PGM represents a factorization of a probability distribution into a product of low-dimensional probability distributions [MCB10].

A central place in the development of a theory of PGM has the concept of independence of random variables. Random variables X_a and X_b are conditionally independent given random variable X_c, if the following factorization holds,

$$p(x_a, x_b|x_c) = p(x_a|x_c)p(x_b|x_c). \tag{4.13}$$



(a) $b \perp\!\!\!\perp c|a$ (b) $a \perp\!\!\!\perp c|b$ (c) $a \perp\!\!\!\perp b$

Figure 4.3: Three possible Bayesian Networks with three nodes, and their implied independence assertions. The probability distributions the correspond to the Bayesian networks factorize as: a) $p(a)p(b|a)p(c|a)$, b) $p(a)p(b|a)p(c|b)$, c) $p(a)p(b)p(c|a,b)$

Conditional independence property can be compactly written as $X_a \perp\!\!\!\perp X_b|X_c$. Graphical model is a compact way to specify all conditional or unconditional independence properties that hold for a multidimensional probability distribution $p(x)$.

Applications of probabilistic graphical models can be found in many disciplines, such as computer vision, machine learning, bioinformatics or digital communications. In this brief exposition to the theory of probabilistic graphical models we build upon the following two rules of the probability,

- Product rule:
$$p(x_a, x_b) = p(x_a|x_b)p(x_b) \tag{4.14}$$
- Marginalization rule:
$$p(x_a) = \sum_{x_b} p(x_a, x_b) \tag{4.15}$$

4.4.1 Directed Graphical Models

Starting from the product rule, any probability distribution can be factorized in the following way

$$p(x) = \prod_{i=1}^{n} p(x_i|x_{\pi_i}), \tag{4.16}$$

where $\pi_i = \{1, \ldots, i-1\}$ denotes the set of indexes that lexicographically precede the index i. For example, joint probability distribution of four random variables a, b, c and d, can be factorized as

$$p(x_a, x_b, x_c, x_d) = p(x_a)p(x_b|x_a)p(x_c|x_a, x_b)p(x_d|x_a, x_b, x_c). \tag{4.17}$$

Usually a random variable x_i does not depend on all previous random variables x_{π_i}, but only on a subset of them x_{pa_i}, where subset $pa_i \subseteq \pi_i$ is referred to as the *parents* of i.

In that case, we can write the joint probability $p(x)$ as

$$p(x) = \prod_{i=1}^{n} p(x_i|x_{pa_i}). \tag{4.18}$$

Factorization in Eq. 4.18 can be represented as a *directed acyclic graph* (DAG) where each node X_i represents a variable in the original probability distribution X. To indicate that node X_i statistically depends only on its parent nodes $X_j, j \in pa_i$, a directed edge is introduced in the graph that goes from X_j to X_i, for each $j \in pa_i$. An illustrative example of a directed graphical model is given in Fig. 4.2(a).

In the literature [MCB10], directed acyclic graphs (DAG) that represent probability distributions are usually called Bayesian networks. For any Bayesian network it is possible to prove that there is at least one permutation of nodes such that any node appears in the permutation after its parents. Figure 4.3 shows three possible Bayesian networks with three nodes.

d-separation

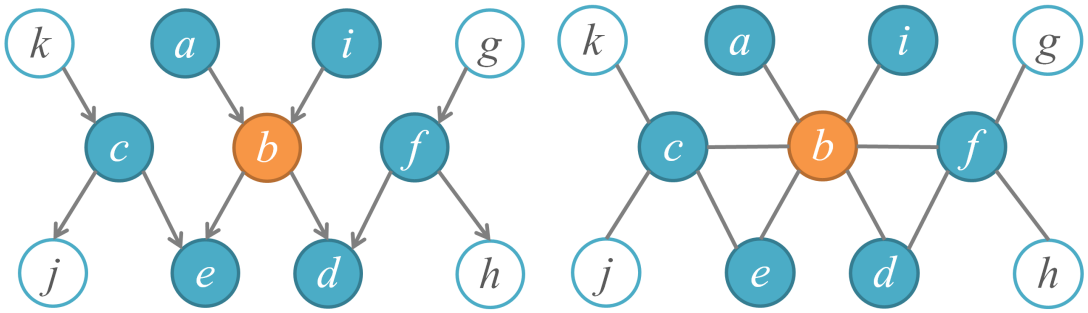
In directed graphical models, d-separation is a generalization of the notion of Markov blanket, and is used to verify statements about conditional independence of sets of nodes. In order to verify that sets of nodes X_A and X_B are conditionally independent given the set of nodes X_C in a Bayesian network, it is necessary to show that each path between X_A and X_B is blocked. There are three rules that help to establish that a certain path in a graph is blocked:

- the path contains a sequence of directed edges $X_i \longrightarrow X_j \longrightarrow X_k$, where node X_j is observed.
- the path contains a sequence of directed edges $X_i \longleftarrow X_j \longrightarrow X_k$, where node X_j is observed.
- the path contains a sequence of directed edges $X_i \longrightarrow X_j \longleftarrow X_k$, where neither node X_j nor any of its descendants is observed.

d-separation is also very useful in a theoretical study of graphical models, because many useful properties of graphical models can be proved by d-separation. For example, it can be proved that a probability distribution $p(x)$ that satisfies all conditional independence assertions implied by d-separations in a DAG, can be factorized according to Eq. 4.18. The converse statement also holds.

4.4.2 Undirected Graphical Models

Some probability distributions have conditional independence properties which cannot be represented by any Bayesian network. Therefore another type of graphical models known as Markov Random Fields (MRF) is used to specify conditional independence properties in the form of undirected graphs. An example of probability distribution that



(a) Markov blanket in DAG (b) Markov blanket in UG

Figure 4.4: Markov blankets in probabilistic graphical models. a) In directed graphs, Markov blanket of a node incorporates its parents, children and other parents of its children. b) In undirected graphs, Markov blanket simply consists of neighbors of a node.

corresponds to an undirected graph, but whose conditional independence properties cannot be represented by any Bayesian network is shown in Fig. 4.2(c).

Before we formally define probability distribution of an undirected graph, we first need to define a clique in a graph. A clique c corresponds to the set of nodes X_c in a graph $\mathcal{G} = (V, E)$, such that there is an edge between any two nodes in the set X_c, i.e. if $X_i, X_j \in X_c$ then $(X_i, X_j) \in E$. In other words, a clique c corresponds to a fully connected subgraph of the original graph. If for a clique c there is no other clique c' such that c is a proper subset of c', then the clique c is called maximal.

Now we can express the probability distribution $p(x)$ represented by a Markov random field as a normalized product of factors $\psi_c(x_c)$ that operate on maximal cliques c,

$$p(x) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \psi_c(x_c). \tag{4.19}$$

In Eq. 4.19, the set of all maximal cliques in the graph is denoted by $\mathcal{C}$, the nonnegative factor of a clique c is denoted by ψ_c, and a normalization constant that provides that all probabilities sum up to one, $\sum_{x \in \mathcal{X}} p(x) = 1$, is denoted by Z. The normalization constant Z is also called the partition function.

A directed graphical model can be converted to an undirected graphical model by applying two simple rules: i) each directed edge is replaced by an undirected edge, ii) for any node X_i that has more than one parent, establish an undirected edge between any two parent nodes X_j and X_k, $j, k \in pa_i$. These rules are used to transform each conditional probability $p(x_i|x_{pa_i})$ of a Bayesian network to a factor $\psi(x_i, x_{pa_i})$ that corresponds to a newly established clique $\{i\} \cup pa_i$. The inclusion of edges between all parents of some node is called *moralization* of a graph.

The transformation from directed to undirected graphical model is not reversible, and some conditional independence properties of a directed graphical model may be lost in the resulting undirected graphical model. For example, the directed graphical model in Fig. 4.3(c) can be converted to a fully connected undirected graphical model

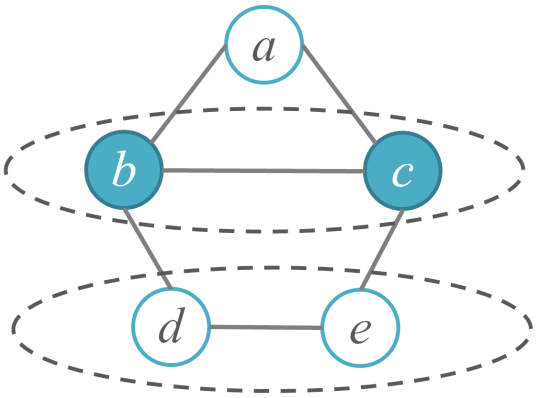


Figure 4.5: An illustration of graph separation in undirected graphical models. Nodes $\{b, c\}$ separate node a from the set of nodes $\{d, e\}$ because any path between a and d, or a and e must pass either through b or c.

with nodes a, b, c, that does not support independence assertion $a \perp\!\!\!\perp b$ that is satisfied by the original directed graphical model.

Graphical models represent the conditional independence assertions of the underlying probability distribution. The assertions can be easily formulated using the *Markov properties*: In case of directed graphical models (Bayesian networks), each node is independent of its non-descendants given its parents. In case of undirected graphical models (Markov random fields), a node is independent of its non-neighbors, give its neighbors.

Another very useful concept in probabilistic graphical models is the Markov blanket. Markov blanket of a certain node X_i is the minimal set of nodes X_{mb_i}, such that node X_i is conditionally independent of any other node X_j that is not in the blanket, given the blanket, i.e. $X_i \perp\!\!\!\perp X_j | X_{mb_i}, \forall j \notin mb_i \cup \{i\}$. In undirected graphical models, Markov blanket of a node consists of node's neighbors. In directed graphical models, Markov blanket of a node includes node's parents, children, and also other parents of its children. An illustration of Markov blankets in undirected and directed graphical models is given in Fig. 4.4.

In undirected graphical models, graph separation (equivalent of d-separations for DAG) is defined by the following simple rule: sets of nodes X_A and X_B are conditionally independent given X_C if any path in the graph that connects X_A and X_B must pass through X_C. For example, in Fig. 4.5 nodes $\{b, c\}$ separate the node a from the nodes $\{d, e\}$. According to graph separation, it follows that $a \perp\!\!\!\perp \{d, e\} | \{b, c\}$.

In undirected graphical models, the famous Hammersley-Clifford theorem connects the graph separation and factorization of the probability distribution: if a strictly positive probability distribution $p(x)$ satisfies all conditional independence assertions implied by graph separation in an undirected graph $\mathcal{G}$, then probability distribution $p(x)$ can be factorized according to Eq. 4.19. The converse statement also holds.

4.4.3 Inference Techniques in Graphical Models

Let (X_A, X_B) be a partitioning of a random variable X into two disjoint sets. Inference solves the following two basic kinds of problems [Bar12]:

- Marginal probabilities:
$$p(x_A) = \sum_{x_B} p(x_A, x_B) \tag{4.20}$$
- Maximum a posteriori (MAP) probability:
$$\max_{x_B} p(x_B|x_A) \tag{4.21}$$

Thus, inference in a graphical model involves the computation of marginal probabilities, or finding values of variables that maximize the posterior probability.

Graphical models allow for efficient computation of marginal probabilities by following the factorization implied by the graphical model and using the distributive law for multiplication and addition. The basic idea is that instead of naively computing $xy + xz$ by two multiplications and one addition, we can use distributive law and calculate $x(y + z)$ that contains one multiplication and one addition, i.e. we can save one multiplication.

Elimination Algorithm

Let us first show on the undirected graphical model of Fig. 4.2(b) how marginal probability $p(d)$ can be efficiently computed by elimination of variables. Naive calculation of marginal probability involves explicit computation of the product of factors for all possible values of marginalized variables,

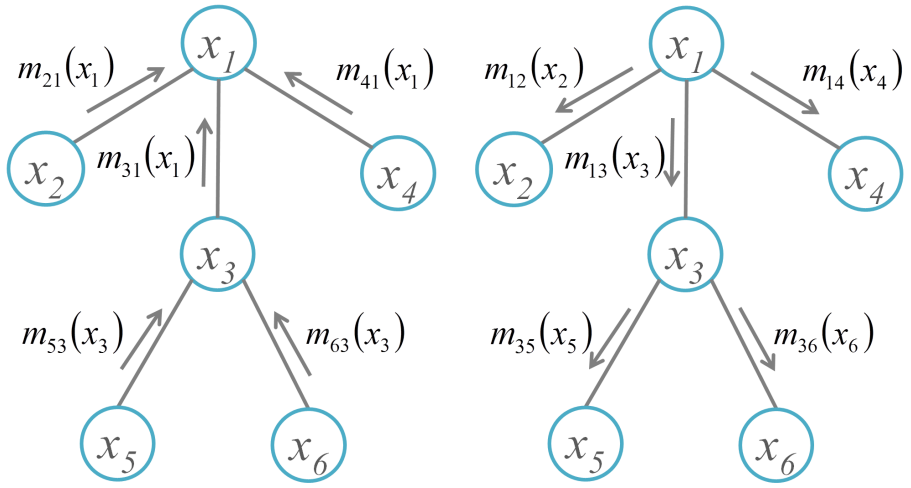
$$\begin{aligned} p(e) &= \sum_{a,b,c,d} p(a, b, c, d, e) \\ &= \frac{1}{Z} \sum_{a,b,c,d} \psi(a, b, c)\psi(b, d)\psi(c, e)\psi(d, e). \end{aligned} \tag{4.22}$$

A significant reduction in computation is obtained if the factors that are constant within a sum are pulled out in front of the sum,

$$\begin{aligned} p(e) &= \frac{1}{Z} \sum_{a,b,c,d} \psi(a, b, c)\psi(b, d)\psi(c, e)\psi(d, e). \\ &= \frac{1}{Z} \sum_{d} \psi(d, e) \sum_{c} \psi(c, e) \sum_{b} \psi(b, d) \sum_{a} \psi(a, b, c). \end{aligned} \tag{4.23}$$

Another example of variable elimination is the computation of the marginal probability $p(x_i)$ in a Markov chain with factorization

$$p(x) = \frac{1}{Z} \prod_{i=1}^{n-1} \psi(x_i, x_{i+1}). \tag{4.24}$$



(a) bottom-up phase (b) top-down phase

Figure 4.6: Illustration of the message passing algorithm in the tree-structured graphical model.

Naive approach to the computation of the marginal probability $p(x_i)$ would be:

$$p(x) = \frac{1}{Z} \sum_{x_1} \cdots \sum_{x_{i-1}} \sum_{x_{i+1}} \cdots \sum_{x_n} \prod_{i=1}^{n-1} \psi(x_i, x_{i+1}). \tag{4.25}$$

The naive procedure would involve $O(\prod_{i=1}^{n-1} |\mathcal{X}_i|)$ multiplication operations. The number of computations can be significantly reduced if factors are pulled out in front of the sums,

$$\begin{aligned} p(x_i) = \frac{1}{Z} \Big[\sum_{x_{i-1}} \psi(x_{i-1}, x_i) \sum_{x_{i-2}} \psi(x_{i-2}, x_{i-1}) \cdots \sum_{x_1} \psi(x_1, x_2) \Big] \\ \cdot \Big[\sum_{x_{i+1}} \psi(x_i, x_{i+1}) \sum_{x_{i+2}} \psi(x_{i+1}, x_{i+2}) \cdots \sum_{x_n} \psi(x_{n-1}, x_n) \Big]. \end{aligned} \tag{4.26}$$

With variable elimination, the number of multiplication operations is reduced to $O(\sum_{i=1}^{n-1} |\mathcal{X}_i||\mathcal{X}_{i+1}|)$. Although variable elimination is efficient for calculating the marginal probability of one variable, it is not efficient for calculating marginal probabilities of all variables in the graph because operations are repeated. Belief propagation algorithm resolves this issue by defining messages that nodes in the graph exchange among each other, which are used for computation of all marginal probabilities in the graph.

Belief Propagation

The aim of belief propagation is to avoid repetition of intermediate results in calculation of marginal probabilities in a tree-structured graphical model. The idea is that intermediate results are exchanged in the form of messages between neighbor nodes in the tree,

so that messages can be reused for computation of all marginal probabilities. The algorithm has many forms, but the most popular is the sum-product algorithm [Mur12].

To explain how belief propagation works, we use a tree graph in Fig. 4.6. One node in the tree is selected as a root, which then induces ordering among other nodes of the tree. Messages are first sent in bottom-up manner from leaf nodes to their parents in the tree, then these nodes send messages further towards the root. The bottom-up phase of message passing ends when all messages arrive at the root. Message sent from node x_i to node x_j is obtained by: i) multiplying all messages that arrived earlier at node x_i with the factor associated with clique (x_i, x_j), and ii) summing the products over x_i creates a message that is a function of only x_j:

$$m_{ij}(x_j) = \sum_{x_i} \psi_{ij}(x_i, x_j) \prod_{k \in \mathcal{N}(i) \setminus j} m_{ki}(x_i). \tag{4.27}$$

After all messages arrive at the root, the top-down phase can begin. The root now sends messages to its children, who than send messages further down the tree until messages finally reach the leaf nodes. The marginal probability at any node of the tree is given by the product of all messages that arrived at that node,

$$p(x_i) \propto \prod_{j \in \mathcal{N}(i)} m_{ji}(x_i). \tag{4.28}$$

Maximum A Posteriori (MAP) Inference

Belief propagation (i.e. the sum-product) algorithm efficiently computes marginal probabilities in a graphical model. In many applications of PGM, it is of interest to compute values of variables with the maximal probability, known as Maximum A Posteriori (MAP) estimation. By following the same ideas that led to the sum-product algorithm, i.e. distributivity of multiplication and addition operations, we note that distributivity also holds for multiplication and max operation: $\max(ab, ac) = a \cdot \max(b, c), a > 0$. Therefore, MAP inference can be achieved by replacing summation in sum-product algorithm with max operation. The latter is known as the max-product algorithm.

4.5 Abnormality Detection by Joint Scene Explanation

After explaining the basics of probabilistic graphical models, we now return to the question of abnormality detection in videos, and propose a sequential method for video parsing that can also be formulated as an inference in a graphical model. In case of a stationary camera the foreground/background segregation becomes feasible due to background subtraction. The foreground mask renders it then possible to turn the abnormality detection problem into a task of video parsing. The goal is thus to explain all the foreground using object hypotheses and to explain each hypothesis using a sample from the set of normal training examples. The underlying statistical inference in a probabilistic graphical model has to be tackled jointly for all hypotheses, since hypotheses

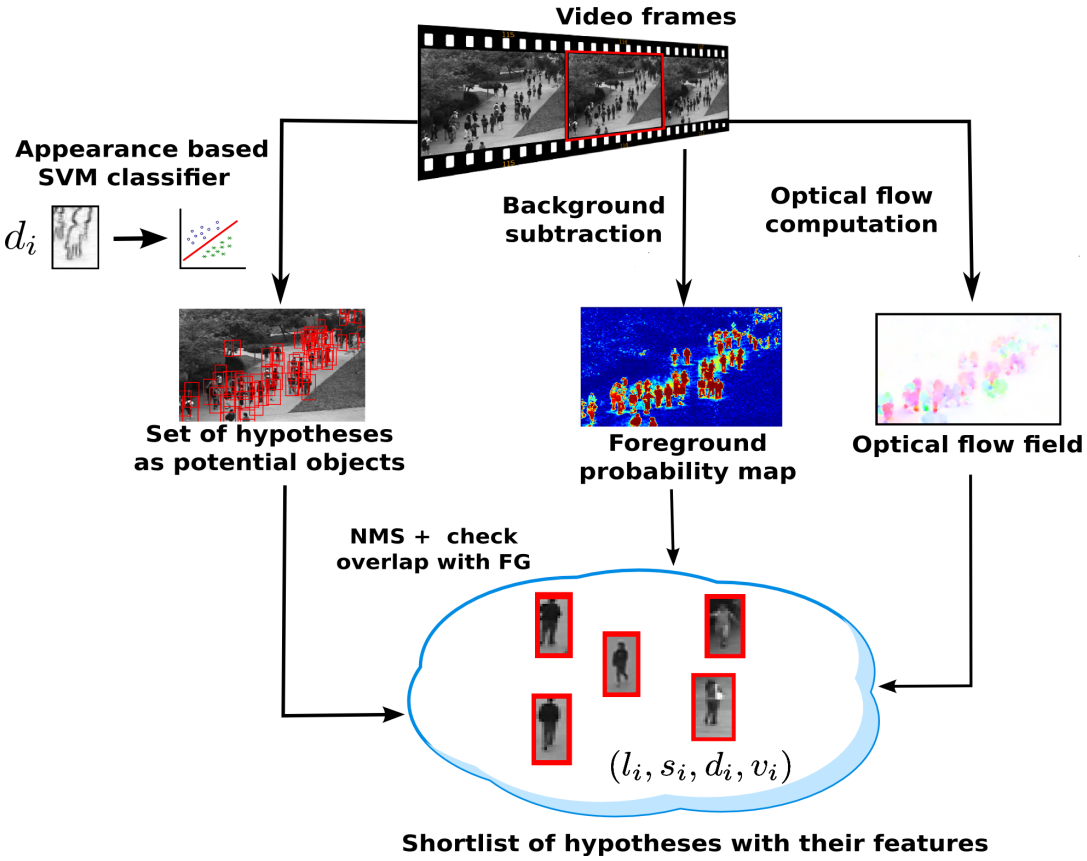


Figure 4.7: Initialization of the video parsing method involves extraction of object hypotheses, segmentation of the foreground pixels using background subtraction and calculation of the optical flow. As a result, a shortlist of object hypotheses is created with their appearance, location, scale and flow features.

can explain each other away. Abnormalities are then those hypotheses that are required to explain the foreground but which themselves cannot be explained by normal training samples.

4.5.1 Initialization

To parse a novel frame in a video, several pieces of information have to be gathered (Fig. 4.7.). In the first place, background subtraction is performed. This is possible because we assume the stationary camera model for our videos.

After background model is calculated in Eq. 4.11, foreground pixels j are found, $f_j = 1$, as pixels that have a large discrepancy between the observation I_j and the background model B_j. The probability that a pixel is foreground $P(f_j = 1)$ is obtained by sigmoid transformation of the difference of pixel's intensity and the background model,

$$P(f_j = 1) = 2\Big(1 + \exp(-\lambda \|I_j - B_j\|)\Big)^{-1} - 1. \tag{4.29}$$

Pixels with foreground probability greater than 0.5 are considered as foreground, $f_j = 1$, and others as background, $f_j = 0$.

Secondly, the optical flow vectors v_j are computed by the method [LFAW08]. Velocity of an object hypothesis h is then calculated as a weighted average of the optical flow vectors over the support $\mathcal{S}_h$ of the hypothesis h

$$v_h = \frac{\sum_{j \in \mathcal{S}_h} P(f_j = 1) \cdot v_j}{\sum_{j \in \mathcal{S}_h} P(f_j = 1)} \tag{4.30}$$

To initialize the subsequent parsing and abnormality detection, a shortlist of candidate object hypotheses is computed. From a large number of object hypotheses that could be established in a video frame most hypotheses are not compatible with the foreground mask as they would be located in the background. Now we can efficiently evaluate an appearance based classifier on candidate object hypotheses in the foreground to obtain a shortlist of relevant hypotheses. Since the training data does not contain abnormal instances but only the background and normal foreground, it is important to note that this is basically an inverted background detector, i.e., a discriminative SVM classifier that is trained to distinguish the background from anything else that deviates from it. A vector of spatiotemporal derivatives

$$d_h = \Big(\frac{\partial I_j}{\partial x}, \frac{\partial I_j}{\partial y}, \frac{\partial I_j}{\partial t} \Big)_{j \in \mathcal{S}_h} \tag{4.31}$$

is used as a feature vector in the SVM classifier. The features capture both the appearance (spatial patterns) and dynamics (temporal patterns) that is crucial for good performance in abnormality detection. The SVM classifier uses a linear kernel and produces a probabilistic output [CL11] as an estimate of the probability of the background class $P(o_h = 0|d_h)$. The classifier is trained in a batch mode on samples from training videos.

The resulting shortlist of object hypotheses is set to have a high recall and low precision. This opportunistic pre-filtering retains a reasonable number of hypotheses (on the order of 10 to 10^2) without losing any relevant ones. However, all of these hypotheses have been found independently of each other. Therefore, there will be spurious hypotheses that can be explained away by others. Moreover, abnormalities can only be discovered once the foreground has been explained by a set of mutually compatible object hypotheses. Abnormal hypotheses are then the ones which cannot be described by the object model that has been learned during training, but which are nevertheless needed to explain the foreground that cannot be explained by other hypotheses.

4.5.2 Model Formulation

Given the initialization, the task of scene parsing is as follows. Select a subset of the initial set of hypotheses that explains all the foreground and explain each object hypothesis using the object model (e.g., which training samples correspond to a particular query hypothesis) that has been learned during training. The activation/deactivation

Figure 4.8: A subset of normal object exemplars obtained from the training data.

of candidate hypotheses and their explanation with the object model have to be solved jointly for all hypotheses since they are mutually competing. The main inference process that parsing is based upon is that of *explaining away* as we will see later. Object hypotheses are necessary for explaining the foreground if they cannot be explained away by others. If such a necessary hypothesis fits to the object model that has been learned from the training videos that only contain normal patterns then this is a normal instance, otherwise we have found an abnormality. Since the model is inherently probabilistic a probability of abnormality is provided. The graphical model of the sequential video parsing approach is shown in Fig. 4.9.

The initialization provides a set of object hypotheses, where each hypothesis has a location $l_h \in \mathbb{R}^2$, a scale $s_h \in \mathbb{R}$, an overall appearance descriptor $d_h \in \mathcal{D}$ that lives in feature space $\mathcal{D}$, and a velocity $v_h \in \mathbb{R}^2$. After the initialization, all object hypotheses are assumed to be required, i.e. the indicator variable $o_h \in \{0, 1\}$ is initialized as $o_h = 1$. Our goal is now to find a subset of all hypotheses that is necessary and sufficient for explaining all pixels of the foreground mask $f_j \in \{0, 1\}$. Moreover, we aim at explaining each hypothesis based on a normal object sample from the training data. Thus, for each hypothesis h the best exemplar $m_h \in \mathcal{M}$ from the training data $\mathcal{M}$ is sought (Fig.4.8). For abnormal objects all exemplars will obviously have high matching costs. Consequently, the probability that sample m_h is matched to the h-th hypothesis in a query frame depends on how similar they are in appearance, $\Delta(d_h, d_{m_h})$. Δ is the distance in the feature space $\mathcal{D}$. Moreover, each visual pattern has a particular probability to occur at a specific location, e.g. cars are more likely to drive on roads than on sidewalks, whereas pedestrians are more likely to walk on sidewalks. The probability that the training sample m_h will be matched to the hypothesis h is given by

$$\begin{aligned} P(m_h|l_h, d_h) &\propto P(m_h|l_h) \cdot P(m_h|d_h) \\ &\propto \frac{\exp\Big(-\beta_l \cdot \|l_h - l_{m_h}\|\Big)}{Z(l_h)} \times \frac{\exp\Big(-\beta_d \cdot \Delta(d_h, d_{m_h})\Big)}{Z(d_h)}, \end{aligned} \tag{4.32}$$

where $Z(\cdot)$ is the partition function. The probability of hypothesis h being an actual object (and not a spurious detection) depends on the observed properties of the hypothesis

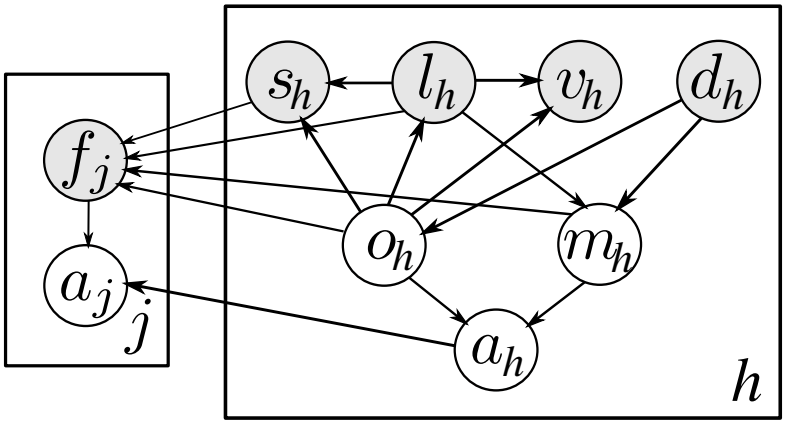


Figure 4.9: Graphical model of the proposed sequential video parsing method for abnormality detection.

(descriptor d_h, location l_h, scale s_h, and velocity v_h) and is given by

$$P(o_h = 1|d_h, l_h, s_h, v_h) \propto P(o_h = 1|d_h) \times p(l_h|o_h = 1) \cdot p(s_h|o_h = 1, l_h) \cdot p(v_h|o_h = 1, l_h) \tag{4.33}$$

Here $P(o_h|d_h)$ is the SVM appearance classifier from Sec. 4.5.1, while $p(l_h|o_h)$, $p(s_h|o_h, l_h)$ and $p(v_h|o_h, l_h)$, for $o_h = 1$, are nonparametric models of location, scale and velocity of normal objects in the image. Otherwise, if $o_h = 0$, location, scale and velocity have uniform distribution.

Finally, we need to estimate the foreground probability of a pixel j. This probability depends on all hypotheses h that cover the pixel. Let $\mathcal{S}_h$ be the support of the h-th hypothesis, i.e., the set of all pixels that are covered by it. Then $\{h : j \in \mathcal{S}_h\}$ is the set of all hypotheses that contain pixel j. We assume that the probability the pixel is background, given all the hypotheses, can be expressed as a product of probabilities of the pixel being background, given each hypothesis alone. We also account for a possibility that the pixel is foreground even if all hypotheses claim it is background. This is modeled by a small probability p_0. The foreground probability is therefore given by

$$P(f_j = 1|\{o_h, m_h, l_h, s_h\}_{h:j\in\mathcal{S}_h}) = 1 - (1 - p_0) \prod_{h:j\in\mathcal{S}_h} \big(1 - P(f_j = 1|o_h, m_h, l_h, s_h)\big) \tag{4.34}$$

To obtain the foreground probability of a pixel based on a training sample m_h, the foreground probability map $P(f^{m_h} = 1)$ of the training sample is pasted into the query frame at the location l_h. Thus we have to shift and scale it from the reference frame of the training sample into that of the current frame and obtain

$$P(f_j = 1|o_h, m_h, l_h, s_h) = o_h \cdot \mathbf{1}(j \in \mathcal{S}_h) \cdot P(f^{m_h}_{s_h^{-1}(l_j - l_h)} = 1) \tag{4.35}$$

Here $\mathbf{1}(\cdot)$ is the indicator function and if $o_h = 0$ or $j \notin \mathcal{S}_h$ then the hypothesis h does not explain the pixel j.

4.5.3 Inference by Foreground Parsing

The goal is now to estimate which of the hypotheses are actually needed to explain the foreground and to find a matching training sample for each hypothesis. For abnormal hypotheses Eq. 4.33 will yield low probabilities. If foreground $f_j = 1$ is observed and asserted by the hypothesis h, and no other hypothesis can be found that could explain the presence of the foreground at that pixel, then the probability of the hypothesis h increases. This statistical inference is also called *explaining away* in the literature, since for an observed variable f_j different hypotheses h that share the same pixel j become statistically dependent so that the absence of one hypothesis can dictate the presence of another.

To infer the unknown variables o_h and m_h, we have to find the joint configuration $\{\hat{o}_h, \hat{m}_h\}_h$ that maximizes the posterior probability

$$\begin{aligned}\{\hat{o}_h, \hat{m}_h\}_h &= \operatorname*{argmax}_{\{o_h, m_h\}_h} P(\{o_h, m_h\}_h | \{d_h, l_h, s_h, v_h\}_h, \{f_j\}_j) \\ &= \operatorname*{argmax}_{\{o_h, m_h\}_h} \prod_h \Big(P(o_h | d_h, l_h, s_h, v_h) \cdot P(m_h | d_h, l_h) \Big) \\ &\quad \times \prod_j P(f_j | \{o_h, m_h, l_h, s_h\}_{h: j \in \mathcal{S}_h}) \end{aligned} \tag{4.36}$$

To solve the given problem we follow an alternating optimization approach. In each iteration we fix all but one hypothesis h and then maximize over its parameters (o_h, m_h). Each iteration is thus a local search in the space $\{0, 1\} \times \mathcal{M}$ where the variables (o_h, m_h) live

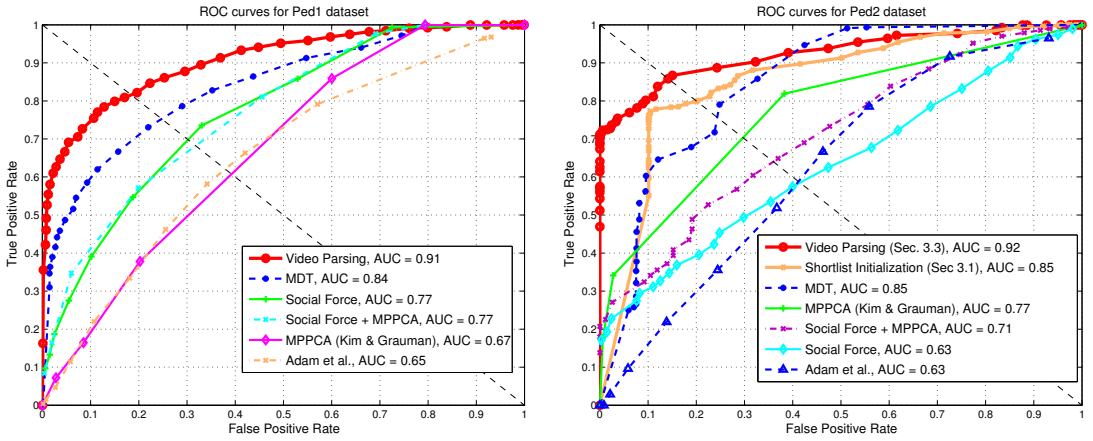
$$\begin{aligned}&\operatorname*{argmax}_{o_h, m_h} P\big(o_h, m_h | \{d_{h\prime}, l_{h\prime}, s_{h\prime}, v_{h\prime}\}_{h\prime}, \{f_j\}_j, \{o_{h\prime}, m_{h\prime}\}_{h\prime \neq h}\big) \\ &= \operatorname*{argmax}_{o_h \in \{0,1\}, m_h \in \mathcal{M}} P(o_h | d_h, l_h, s_h, v_h) \cdot P(m_h | d_h, l_h) \\ &\quad \times \prod_{j \in \mathcal{S}_h} P(f_j | \{o_{h\prime}, m_{h\prime}, l_{h\prime}, s_{h\prime}\}_{h\prime: j \in \mathcal{S}_{h\prime}}). \end{aligned} \tag{4.37}$$

Typically, only few rounds of iterations are needed to converge to a locally optimal solution.

4.5.4 Detecting Abnormalities

Finally, the h-th hypothesis is an abnormality, $a_h = 1$ if this hypothesis is necessary to explain the observed foreground, $o_h = 1$, and if no matching training sample can be found, i.e., the best estimate $\widehat{m}_h$ for a matching sample (obtained from Eq. 4.36) is unlikely to explain this hypothesis,

$$P(a_h = 1 | o_h, m_h) = P(o_h = 1 | d_h, l_h, s_h, v_h) \cdot P(m_h \neq \widehat{m}_h | d_h, l_h, s_h) \tag{4.38}$$



(a) (b)

Figure 4.10: (a) Per-frame abnormality detection results for the *Ped1* dataset. We achieve a 7% gain in AUC over the state-of-the-art. (b) Per-frame abnormality detection results for the *Ped2* dataset. We achieve a 7% gain in AUC over the state-of-the-art. The orange curve illustrates the performance of our approach after the shortlist initialization (Sec. 4.5.1), whereas the red curve depicts the performance after the explaining away procedure (Sec.4.5.3).

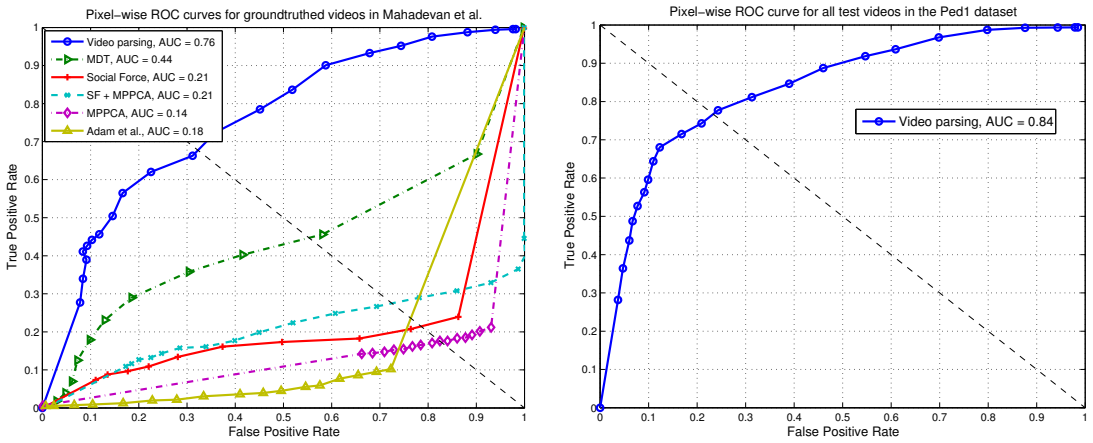
Similarly, pixel j is part of an abnormal object, $a_j = 1$, if it is in the foreground, $f_j = 1$, and if any of the hypotheses that extend over this pixel, $\{h : j \in \mathcal{S}_h\}$, is abnormal,

$$\begin{aligned} & P(a_j = 1 | f_j, \{a_h\}_{h:j \in \mathcal{S}_h}) \\ & = P(f_j = 1 | \{o_h, m_h, l_h, s_h\}_{h:j \in \mathcal{S}_h}) \cdot \max_{h:j \in \mathcal{S}_h} P(a_h = 1 | o_h, m_h) \end{aligned} \tag{4.39}$$

4.6 Experimental Evaluation

4.6.1 Description of the UCSD Anomaly Dataset

We use the challenging UCSD anomaly datasets *ped1* and *ped2*, that were recently proposed by Mahadevan et al. [MLBV10] for measuring the performance of abnormality detection algorithms. Both datasets consist of videos recorded in crowded walkway scenes that also feature lots of challenging abnormal instances which are objects with unusual appearance or behavior. The UCSD *ped1* set contains 34 training and 36 test videos that are all 200 frames long. Due to the low resolution of *ped1* videos, the pedestrians who walk towards and away from the camera are only $10 - 25$ pixels high. In the UCSD *ped2* dataset there are 16 training and 12 test videos that have a variable length (at most 180 frames). Pedestrians in these videos are about 30 pixels high. Videos from



(a) (b)

Figure 4.11: (a) Pixel-wise abnormality detection for the labeling provided by Mahadevan *et al.*. We observe a 32% improvement in the AUC. (b) Pixel-wise abnormality detection for our fully labeled test set.

both benchmark sets are very crowded, so that object heavily occlude one another.

Abnormalities in the UCSD datasets are not staged but occur naturally in the scene and can be grouped into: i) objects that do not fit to the context of the scene, such as a car on a crowded walkway, or ii) objects that look normal but behave in unusual way, such as people that cycle or skateboard across the walkway or walk in the lawn. Abnormalities from the UCSD benchmark sets include also carts and wheelchairs. The training videos consist only of normal objects and actions, so that a model for abnormalities cannot be learned from it.

4.6.2 Evaluation Protocol

We use the standard protocol for evaluating abnormality detection results that was proposed by Mahadevan et al. [MLBV10]. The protocol consists of frame-wise and pixel-wise criteria. The frame-wise criterion labels a frame as abnormal if it contains at least one abnormal object detection. The localization accuracy of detected abnormalities is verified by the pixel-wise criterion that is more rigorous than the frame-wise criterion, since the detected abnormalities are compared to a pixel-level ground-truth mask. The pixel-wise criterion requires that at least 40% of all ground-truth abnormal pixels to be marked as abnormal in order to count a frame as true positive. By calculating the true positive rate (TPR) and false positive rate (FPR) at different detection thresholds we obtain the receiver operating characteristic (ROC).

Frame-wise and pixel-wise criteria use the area under the curve (AUC) as a performance measure calculated directly from the corresponding ROC curve. For the frame-wise criterion we calculate also the equal error rate (EER) as a value obtained when the false positive and false negative rates are equal. For pixel-wise criterion we compute the rate of detection (RD), that is equal to $1-$EER. The pixel-wise criterion is ap-

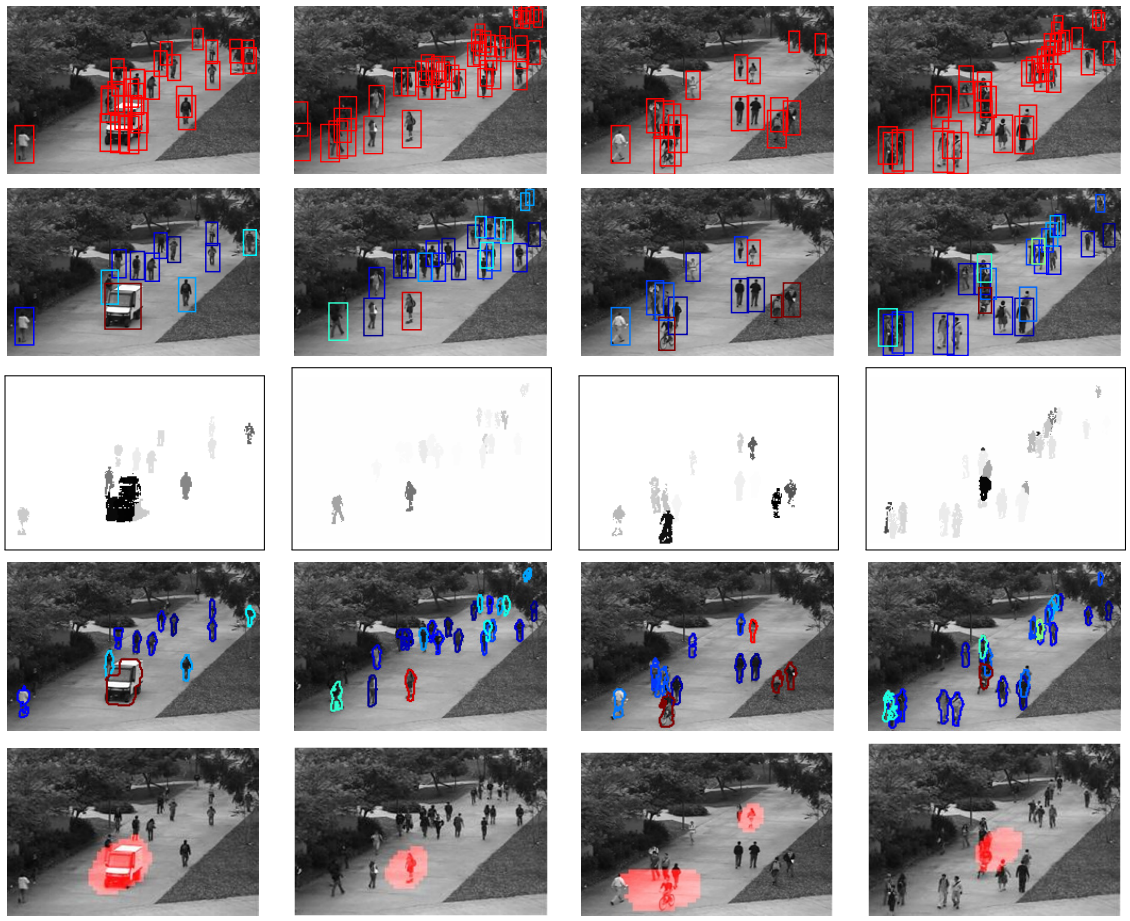


Figure 4.12: Columns show detection results on different frames. Row i) our initial shortlist, row ii) hypotheses and abnormality probability a_h, row iii) per-pixel probability a_j, row iv) best fitting model m_h, row v) result by [MLBV10]. Best viewed in color.

plied on the partially labeled UCSD *ped1* dataset originally provided with the pixel-wise ground-truth annotation. Moreover, in the course of this thesis a pixel-wise ground-truth annotations for the full datasets is completed and included in the evaluation.

4.6.3 Comparing with the State-of-the-Art

We compare the proposed sequential video parsing (SVP) with the state-of-the-art abnormality detection methods on the *Ped1* and *Ped2* benchmark datasets. The methods include the mixture of dynamic textures [MLBV10], the social force model [MOS09], the mixture of optical flow [KG09], the optical flow monitoring method [ARSR08], and a combination of [MOS09] and [KG09] that was investigated in [MLBV10].

In all of the experiments the SVP approach significantly outperforms all the other approaches. The SVP per-frame labeling on the *Ped1* dataset (Fig. 4.10(a)) achieves

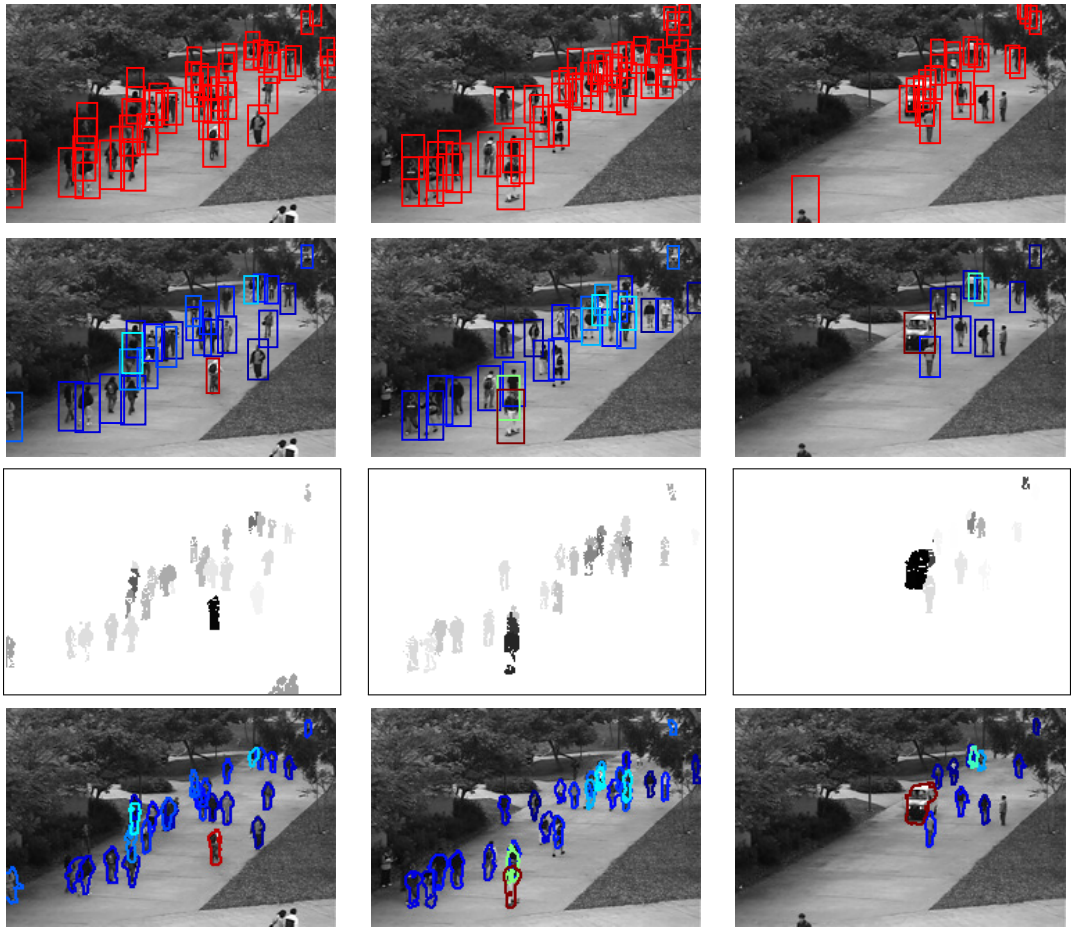


Figure 4.13: Rows show additional detection results on various frames. Column i) our initial shortlist, column ii) hypotheses and abnormality probability a_h, column iii) per-pixel probability a_j, column iv) best fitting model m_h. Best viewed in color.

an EER of 18%, which is an improvement of 7% over [MLBV10], and an improvement of 22% over [KG09]. We also compare the area under the ROC curve (AUC), which is a more robust measure, as it does not depend on only a single spot on the curve. The SVP achieves an AUC of 91% compared to 84% of [MLBV10]. Per-frame labeling on the *Ped2* dataset (Fig. 4.10(b)) yields an EER of 14%, which is an improvement of 11% over [MLBV10], and it also results in an AUC of 92% compared to 85% of [MLBV10]. Nevertheless, current MATLAB implementation of the SVP approach is approximately twice as fast in the prediction phase (5-10 secs per frame) as the approach [MLBV10].

In order to estimate the abnormality of pixels, we follow a direct probabilistic approach where the variables a_j are obtained directly by statistical inference. The abnormality masks are then compared to the pixel-level ground truth masks as in [MLBV10]. The SVP approach improves the AUC in this experiment (Fig. 4.11(a)) by 32% achieving 76% average performance compared to 44% of the approach [MLBV10]. In that pa-

per, the detection performance at the point of equal error was also reported where SVP achieves a 23% gain yielding a detection rate of 68% compared to 45% by [MLBV10]. Note that in the previous experiment [MLBV10] have compared error rates while they are measuring detection rates. This is why we report the standard and more robust AUC in all cases. Fig. 4.11(b) reports the detection performance of the fully labeled *Ped1* test set that we is assembled as part of this thesis.

Fig. 4.12 compares the abnormality localization of the SVP approach with that of [MLBV10]. The columns show results on different frames. i) Row one visualizes the initial shortlist with its spurious detections, ii) row two shows the hypotheses and their abnormality probability a_h (ranging from blue for normal to red for abnormal) after optimization, iii) row three displays the pixel-level abnormality a_j, and iv) row four explains each hypothesis by the best fitting model m_h and for abnormalities all connected abnormal pixels are grouped. The comparison between SVP's localization of abnormalities in row iv) with the approach [MLBV10] in row v) further explains the significant performance gain SVP achieves in Fig. 4.11(a). Further detection results of the sequential video parsing approach are shown in Fig. 4.13.

4.7 Discussion

To avoid the ill-posed problem of directly detecting abnormalities and classifying individual image regions independently from another as 2 abnormal, a sequential video parsing approach is proposed in this chapter. All object hypotheses that are needed to explain the foreground of a single frame in video are jointly inferred. At the same time, each hypothesis seeks to be explained by a normal training example. In the proposed probabilistic graphical model, sets of hypotheses are jointly explaining the foreground while they are also able to explain each other away, simultaneously. Thus hypotheses are not detected individually but their layout is found that jointly describes the scene. Abnormalities are then discovered indirectly as those hypotheses which are needed to explain the scene but which themselves cannot be explained by the normal training samples. The sequential video parsing approach has demonstrated its potential by improving the state-of-the-art performance on a challenging benchmark dataset.

Chapter 5

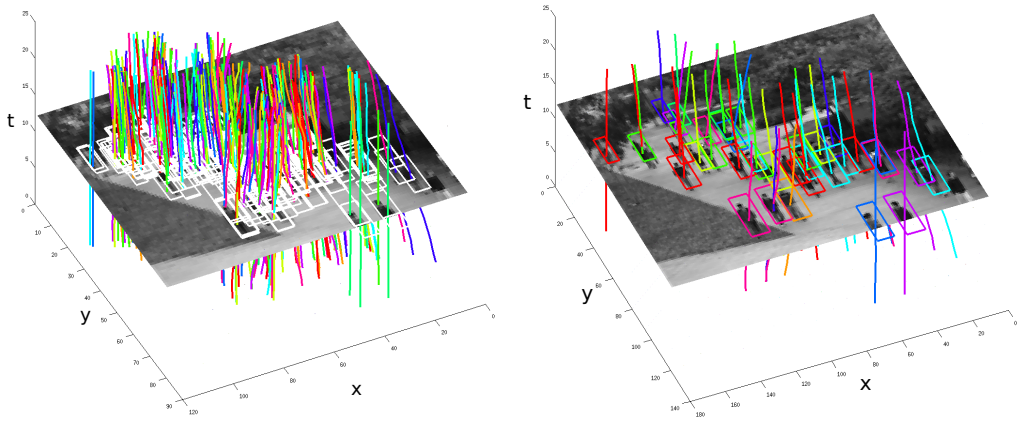
Spatio-temporal Video Parsing for Abnormality Detection

In Chapter 4, a sequential video parsing method for abnormality detection is proposed. That method parses video frames one after another and considers only spatial interactions between object hypotheses in a single frame. In this chapter, video parsing model is extended to the spatio-temporal domain where video frames are jointly parsed. The extension is used to resolve both the spatial and temporal dependencies between object hypotheses in a video and allows for efficient aggregation of evidences from different frames. The new convex formulation of the inference process allows the spatio-temporal video parsing model to find the globally optimal solution.

5.1 Outline of the Approach

As before, we assume that videos are recorded by static cameras (e.g. visual surveillance or industrial inspection). Therefore, robust background subtraction algorithms [WGR+09] can be used to segregate foreground from background in a video. The goal is now to find a set of *spatio-temporal* object hypotheses that jointly explain all foreground pixels in video. This means that normal object hypotheses, which can be learned from the training data, are now spread over the spatio-temporal volume of a video in order to cover foreground pixels, while protruding into the background as little as possible. These hypotheses need to explain the appearance and behavior of the underlying objects in video. As objects are mutually overlapping in crowded scenes, the placement of the object hypotheses in spatio-temporal domain can only be determined jointly. Thus, the aim is to simultaneously select spatio-temporal object hypotheses, which are necessary for explaining the foreground, and identify for each selected hypothesis the best matching instance from the normal object model. Video parsing then *jointly* infers all necessary spatio-temporal hypotheses, so that abnormalities can be *indirectly* discovered in a scene without actually knowing what they look like or how do they behave.

In the first phase of spatio-temporal video parsing, a large number of object can-



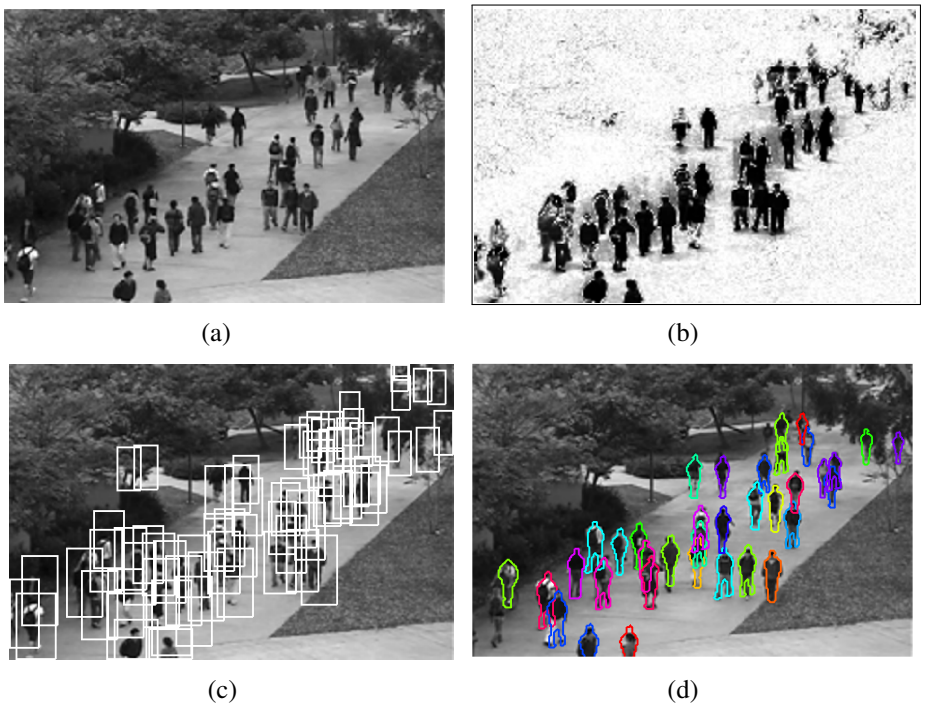
(a) (b)

Figure 5.1: Successive stages of the video parsing: (a) Spatio-temporal object hypotheses found by temporal grouping serve as an input to the video parsing. (b) Subset of spatio-temporal object hypotheses is selected by video parsing to explain the foreground pixels.

didates in each frame is detected and grouped temporally into spatio-temporal object hypotheses. This shortlist of spatio-temporal hypotheses is a superset of all object candidates that might eventually be needed for parsing a video. The object candidates in individual frames are obtained by running a discriminative background classifier and keeping only those patterns that are very unlikely to be background. Subsequently, object candidates in individual frames are linked temporally according to their motion cues so as to establish the shortlist of spatio-temporal object hypotheses. In the second phase of video parsing, the goal is to select hypotheses from the shortlist that can explain the foreground, and to simultaneously find normal object instances that match those hypotheses. This is formulated as an inference problem in a graphical model whose goal is to maximize the probability of the foreground explanation for a video. The inference problem is cast as a convex optimization problem where the unknown variables indicate both, the selection of hypotheses from the shortlist and their corresponding normal object prototypes learned from the training videos. Correspondences between hypotheses and normal object prototypes are based upon their shape, location as well as their appearance and behavior. The probability of abnormality of each hypothesis necessary for explaining the foreground is then calculated using the results of inference.

5.2 Model for Spatio-temporal Video Parsing

In case of a stationary camera, the foreground/background segregation becomes feasible due to background subtraction. The foreground mask renders it then possible to



(a) (b)

(c) (d)

Figure 5.2: Successive stages of the video parsing: (a) Source frame of a video. (b) Foreground probability map that needs to be explained by video parsing. (c) Object candidates found by inverted background detector. (d) Normal object prototypes chosen during video parsing to explain the selected object hypotheses. Best viewed in color.

turn the abnormality detection problem into a task of video parsing. The goal is thus to explain all the foreground of a video using object hypotheses and to explain each hypothesis by an object model learned from the set of normal training videos. The underlying statistical inference problem has to be tackled jointly for all hypotheses, since hypotheses are overlapped and thus can explain each other away. Abnormalities are then those hypotheses that are required to explain the foreground but which themselves cannot be explained by any prototype from the normal object model.

Foreground segmentation After calculating the background model B as in Eq. 4.11, it can be used to find all foreground pixels j, $f_j^t = 1$, as those that have a large discrepancy between the observation I_j^t and the background model B_j^t. The probability that a pixel is foreground $P(f_j^t = 1)$ is obtained by the sigmoid transformation of the difference of pixel's intensity and background model,

$$P(f_j^t = 1) = 2\Big(1 + \exp(-\lambda \|I_j^t - B_j^t\|)\Big)^{-1} - 1. \tag{5.1}$$

Pixels with foreground probability greater than 0.5 are considered as foreground, $f_j^t = 1$, and others as background, $f_j^t = 0$.

Shortlist of Object Hypotheses. For parsing a video, we need to specify a list of spatio-temporal object hypotheses that is sufficient for explaining foreground pixels in

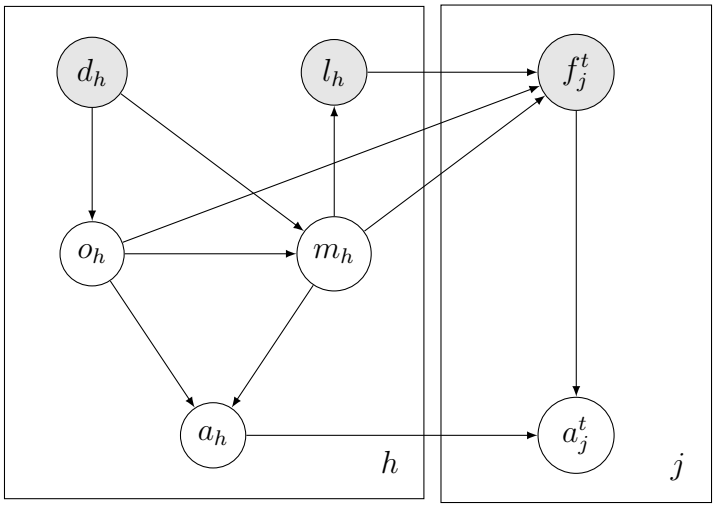


Figure 5.3: Probabilistic graphical model of the spatio-temporal video parsing. The left plate contains all spatio-temporal hypotheses h with their descriptors d_h and locations l_h. The right plate comprises all pixels j with their foreground labels $f_j^t \in \{0, 1\}$. By video parsing, we infer the set of hypotheses, $o_h \in \{0, 1\}$, that are necessary for explaining the foreground, and *jointly* explain the selected hypotheses by the normal object prototypes $m_h \in \{1, \ldots, K\}$. Finally, for each selected hypothesis h we decide if it is abnormal, $a_h \in \{0, 1\}$, and also mark foreground pixels that belong to abnormal objects, $a_j^t \in \{0, 1\}$.

video. An input to the spatio-temporal video parsing algorithm consists of the most suitable *spatio-temporal* object hypotheses for the task of foreground explanation. In Sect. 5.5 we explain the procedure for creating a shortlist of object hypotheses that has a high recall, i.e. where the majority of true-positive object hypotheses is included in the shortlist. However, as the precision rate of the proposed shortlist is low, there will be many superfluous hypotheses that are then explained away by others during video parsing.

We assume that hypotheses from the shortlist span a time window $\{t - \tau, \ldots, t\}$. Each hypothesis h represents a spatio-temporal tube covering locations $l_h := (l_h^{t-\tau} \ \ldots \ l_h^t)$. This is a trajectory of locations $l_h^t = (x_h^t \ y_h^t \ s_h^t)^\top$, which specify the center (x_h^t, y_h^t) and the scale s_h^t of a potential object h at time t. The scale of an object is its size relative to the size (W, H) of the object model. The *support region* of an object hypothesis h at time t is the bounding box of size $(s_h^t W, \ s_h^t H)$, and the set of all pixels j that belong to it is denoted by $\mathcal{S}_h^t$.

The goal of video parsing is then to select a subset from the shortlist of hypotheses that is both necessary and sufficient for explaining the foreground of a test video while, at same time, finding normal object model's prototypes to explain the hypotheses of the subset (Fig. 5.2).

Spatio-temporal object descriptor. A spatio-temporal hypothesis h matches its

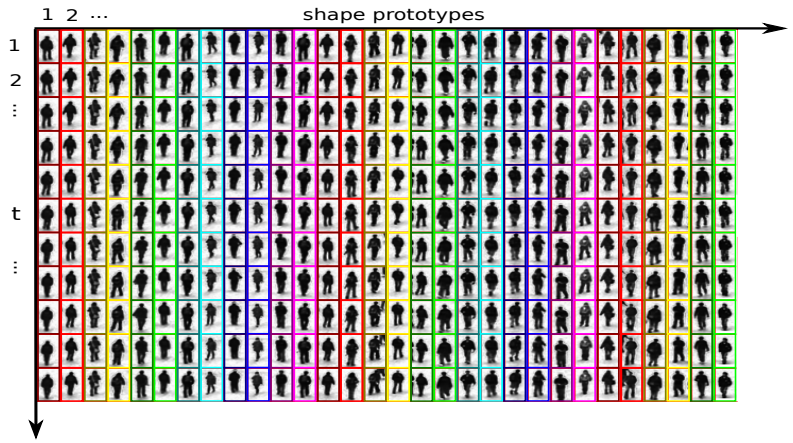
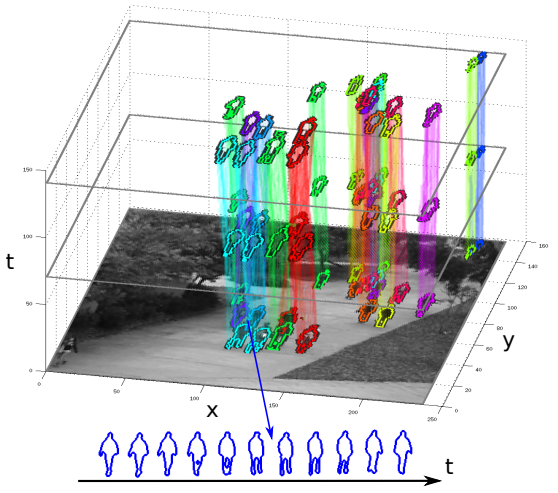


Figure 5.4: The normal object model consist of a set of spatio-temporal shape prototypes, each being a sequence that captures the temporal evolution of a particular shape. Prototypes are accompanied by the appearance and motion descriptors.

corresponding normal object model's prototype both in terms of appearance and motion. Thus, we need a spatio-temporal descriptor d_h to capture the essence of both appearance and motion of hypothesis h. We build descriptor d_h by concatenating per-frame appearance and motion descriptors d_h^t calculated at each time t, $d_h := \left(d_h^{t-\tau} \ \ldots \ d_h^t\right)^\top$. Per-frame object appearance is represented by the spatial derivatives of pixel's intensity in the support region $\mathcal{S}_h^t$ of hypothesis h. Analogously, object motion is represented by the temporal derivatives of pixel's intensity. The appearance and motion representations are combined into a single feature vector as in Eq. 4.31. Since the overall descriptor d_h is long and redundant, PCA is applied to find a compact representation by projecting onto an eigen-basis such that most of the signal is preserved (about 95%).

Activating hypotheses needed for parsing. Not all object hypotheses from the shortlist are needed to explain foreground pixels of video. Video parsing retains only the indispensable hypotheses that cannot be explained away by other hypotheses. Therefore, there is an indicator variable $o_h \in \{0, 1\}$. To initialize parsing, a discriminative classifier is trained to distinguish background spatio-temporal patterns from anything else. This background classifier computes the probability that hypothesis h is background, $P(o_h = 0|d_h)$, which is then inverted to obtain the foreground probability. A hypothesis with high foreground probability can still become inactive if it gets explained away by others during video parsing.

Matching with the object model. Video parsing jointly explains foreground pixels with object hypotheses, and the selected hypotheses $\{h : o_h = 1\}$ with normal object model's prototypes learned from the training data. There are K normal object prototypes that represent a diversity of shape, appearance, and motion of normal objects. Video parsing then determines for each selected hypothesis h which of the K prototypes best explains it. The prototype that video parsing associates with hypothesis h is indicated by the variable $m_h \in \{1, \ldots, K\}$. Sect. 5.4 explains in detail the learning of the normal object model. For the time being, we assume that K normal object prototypes



(a)

(b)

Figure 5.5: (a) Spatio-temporal tubes illustrate the hypotheses selected by video parsing. Normal shape contours that explain the hypotheses are shown overlaid. (b) Superfluous hypotheses are eliminated by the statistical inference of *explaining away*. The idea is the following: Object hypothesis (yellow) is used at the beginning of video parsing to explain the foreground pixel in the middle. Other object hypotheses (red and blue) are introduced later to explain the top and bottom pixels. However, the pixel in the middle is also explained by new hypotheses, so that the original (yellow) hypothesis is not needed anymore and it can be eliminated.

are provided as input to the parsing algorithm.

For each hypothesis h the best prototype $m_h \in \{1, \ldots, K\}$ from the learned object model is sought (Fig. 5.4). For abnormal objects all prototypes will obviously have high matching costs. Consequently, the probability that prototype m_h is matched to a hypothesis h in a query video depends on how similar they are in both appearance and motion, $\Delta(d_h, d_{m_h})$. Here, Δ denotes a distance function that measures the dissimilarity of spatio-temporal descriptors in the corresponding feature space. Given the spatio-temporal descriptor d_h of hypothesis h, the probability of matching prototype m_h with the hypothesis h is the Gibbs distribution,

$$P(m_h|d_h) = \frac{1}{Z(d_h)} \exp\big(-\beta\Delta(d_h, d_{m_h})\big), \tag{5.2}$$

where $Z(d_h)$ is the partition function used to normalize the probability distribution.

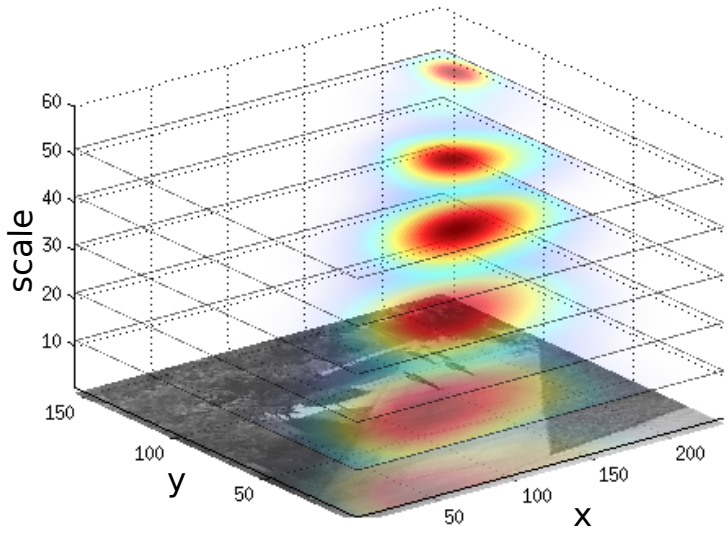


Figure 5.6: The distribution of locations of normal object prototypes estimated by the Parzen windows at multiple scales (represented as horizontal slices).

Moreover, normal objects typically occupy some location in a scene more often than other, and also tend to move at a certain speed. For example, cars are more likely to drive on roads than on sidewalks, whereas pedestrians are more likely to walk on sidewalks. Consequently, the probability of observing hypothesis h that matches the prototype m_h depends on its location l_h^t and velocity $l_h^t - l_h^{t-1}$,

$$P(l_h|m_h) \propto P_{m_h}^{loc}(l_h^t) \cdot P_{m_h}^{vel}(l_h^t - l_h^{t-1}). \tag{5.3}$$

The two distributions $P_\bullet^{loc}$ and $P_\bullet^{vel}$ are learned for the K model prototypes using Parzen window density estimation (see Fig. 5.6).

Therefore, the probability that hypothesis h matches to the normal object prototype m_h is

$$P(m_h|o_h, d_h, l_h) \propto o_h \cdot P(m_h|d_h) \cdot P(l_h|m_h). \tag{5.4}$$

Explaining foreground pixels. Video parsing selects hypotheses, $\{h : o_h = 1\}$, and finds corresponding normal object prototypes m_h to explain the foreground. The foreground probability of a pixel j depends on all hypotheses h that overlap with pixel j. Given the support regions $\mathcal{S}_h^t$ of all hypotheses h, $\{h : j \in \mathcal{S}_h\}$ is the set of hypotheses that cover the pixel j. The probability that a pixel is background is equal to the product of background probabilities of the pixel given by each single hypothesis h. Even if all hypotheses assert a pixel j to be background, $P\big(f_j^t = 0|o_h, m_h, l_h\big) = 1,\ \forall h$, we allow for a small probability $P_0 > 0$ that the pixel is foreground. Thus, foreground probability

of pixels j given all hypotheses is

$$P(f_j^t = 1|\{o_h, m_h, l_h\}_h) = \\ 1 - (1 - P_0) \prod_h \Big(1 - P(f_j^t = 1|o_h, m_h, l_h)\Big). \tag{5.5}$$

The foreground probability given an individual hypothesis, $P(f_j^t = 1|o_h, m_h, l_h)$, depends on the shape of the normal object prototype m_h. In the training data, the prototype m_h covers pixels j' with some probability $P_{m_h}(f_{j'}^t = 1)$. Thus, the foreground probability of pixel j under hypothesis h is obtained by taking its corresponding object prototype m_h and "pasting" the foreground probability of m_h at the location of h. The model now needs to be brought into the reference frame of h by scaling and translating it, i.e. $l_j^t = s_h^t \cdot l_{j'}^t + (x_h^t \; y_h^t)^\top$. Then the foreground probability of pixel j given h becomes

$$P(f_j^t = 1|o_h, m_h, l_h) = o_h \cdot \mathbf{1}[j \in \mathcal{S}_h^t] \\ \cdot \sum_{j'} \mathbf{1}[l_j^t = s_h^t \cdot l_{j'}^t + (x_h^t \; y_h^t)^\top] \cdot P_{m_h}(f_{j'}^t = 1). \tag{5.6}$$

Here $\mathbf{1}[\cdot]$ denotes the indicator function. In Eq. 5.6 the foreground probability of pixel j is set to zero if hypothesis h is inactive, $o_h = 0$, or the pixel j does not belong to the support region of hypothesis h, $j \notin \mathcal{S}_h^t$.

5.3 Inference by Foreground Parsing

The goal is now to estimate which of the hypotheses are actually needed for explaining the foreground and to find a matching normal object prototype for each hypothesis. For abnormal hypotheses Eq. 5.4 will yield low probabilities. If foreground $f_j^t = 1$ is observed and the pixel is covered by a hypothesis h, and no other hypothesis can be found that could explain the presence of the foreground at that pixel, then the probability of the hypothesis h increases. This leads to the statistical inference of *explaining away*. For an observed variable f_j^t different hypotheses h that share the same pixel j become statistically dependent so that the absence of one hypothesis can dictate the presence of another (Fig. 5.5).

5.3.1 Joint Inference by MAP

Based on the foreground segmentation mask f_j^t and the shortlist of hypotheses h with spatio-temporal descriptors d_h and trajectories l_h, we need to jointly infer all latent variables $\{o_h, m_h\}_h$ in our graphical model (Fig. 5.3). Following a maximum a posteriori (MAP) approach yields a set of hypotheses that best explain the foreground and are

themselves explained by the normal object prototypes,

$$\{\bar{o}_h, \bar{m}_h\}_h = \max_{\{o_h, m_h\}_h} P(\{o_h, m_h\}_h | \{d_h, l_h\}_h, \{f_j^t\}_j)$$
$$\propto \prod_j P\big(f_j^t | \{o_h, m_h, l_h\}_h\big) \prod_h P(o_h|d_h) P(m_h|o_h, d_h, l_h). \tag{5.7}$$

Instead of explicitly maximizing the posterior probability, we take a negative logarithm of Eq. 5.7 and thereby obtain the energy function $J(\cdot)$ which is then minimized. Furthermore, we decompose the energy function $J(\cdot)$ into two terms, $J_j(\cdot)$ covering the explanation of foreground pixels j, and $J_h(\cdot)$, which involves the explanation of hypotheses h by the normal object prototypes,

$$J(\{o_h, m_h\}_h) := \underbrace{-\sum_j \log P(f_j^t | \{o_h, m_h, l_h\}_h)}_{=: J_j(\{o_h, m_h\}_h)}$$
$$\underbrace{-\sum_h \Big(\log P(o_h|d_h) + \log P(m_h|o_h, d_h, l_h) \Big)}_{=: J_h(\{o_h, m_h\}_h)}. \tag{5.8}$$

To find the MAP solution, we introduce a parsing indicator $z_{h,k} \in \{0, 1\}$, that equals one if hypothesis h is active, $o_h = 1$, and their corresponding normal object prototype is $m_h = k$,

$$z_{h,k} := o_h \cdot \mathbf{1}[m_h = k], \ \forall h, \ \forall k \in \{1, \ldots, K\}. \tag{5.9}$$

To keep the notation simple, let the vector $\mathbf{z}_h := (z_{h,1}, \ldots, z_{h,K})^\top$ denote the parsing indicators of hypothesis h, and the vector $\mathbf{z} := \{\mathbf{z}_h\}_h$ denote the parsing indicators of all hypotheses together. The following lemma now states that the hypotheses explanation $J_h(\cdot)$ can be expressed as a linear function of the parsing indicator $\mathbf{z}$.

Lemma 5.3.1 *The hypotheses explanation term $J_h(\{o_h, m_h\}_h)$ in Eq. 5.8 is a linear function of the parsing indicator $\mathbf{z}$, i.e.*

$$J_h(\{o_h, m_h\}_h) = \mathbf{b}^\top \mathbf{z} + b_0, \tag{5.10}$$

where the parameter vector $\mathbf{b} = \{b_{h,k}\}_{h,k}$ and scalar b_0 do not depend on the parsing indicator $\mathbf{z}$.

Proof of Lemma 5.3.1 The hypotheses explanation $J_h(\{o_h, m_h\}_h)$ (Eq. 5.8) can be written as follows,

$$J_h(\{o_h, m_h\}_h) = \sum_h \Big\{ -(1 - o_h) \log P(o_h = 0|d_h)$$
$$- o_h \log P(o_h = 1|d_h) + o_h \cdot \log Z(d_h)$$
$$+ \sum_{k=1}^{M} \underbrace{o_h \cdot \mathbf{1}[m_h = k]}_{= z_{h,k}} \cdot \Big(\beta \Delta(d_h, d_k)$$
$$- \log P_k^{loc}(l_h^t) - \log P_k^{vel}\big(l_h^t - l_h^{t-1}\big) \Big) \Big\}.$$

By replacing o_h with the sum from Eq. 5.15, we see that the hypotheses explanation term $J_h(\{o_h, m_h\}_h)$ can be expressed as a linear function of the parsing indicator $\mathbf{z}$,

$$J_h(\mathbf{z}) = \mathbf{b}^\top \mathbf{z} + b_0,$$

where the parameter vector $\mathbf{b} = \{b_{h,k}\}_{h,k}$ and scalar b_0 are defined in the following way,

$$\begin{aligned} b_{h,k} = &- \log P(o_h = 1|d_h) + \log P(o_h = 0|d_h) \\ &+ \log Z(d_h) + \beta\Delta(d_h, d_k) - \log P_k^{loc}(l_h) \\ &- \log P_k^{vel}\big(l_h^t - l_h^{t-1}\big) \\ b_0 = &- \sum_h \log P(o_h = 0|d_h), \end{aligned}$$

and they do not depend on the parsing indicator $\mathbf{z}$. ■

To express the foreground explanation term $J_j(\cdot)$ as a function of the parsing indicator $\mathbf{z}$, we first define a function $\Phi_{f_j^t}(\cdot)$ that is parametrized by the foreground value f_j^t of pixel j,

$$\Phi_{f_j^t}(x) := (1 - f_j^t)x - f_j^t \cdot \log\big(1 - e^{-x}\big),\ x > 0. \tag{5.11}$$

The introduced function $\Phi_{f_j^t}(\cdot)$ is convex as we show in the following lemma.

Lemma 5.3.2 *The function $\Phi_{f_j^t}(x),\ x > 0$ (Eq. 5.11) is convex for nonnegative values of the parameter f_j^t.*

Proof of Lemma 5.3.2 The second derivative of the function $\Phi_{f_j^t}(x),\ x > 0$ is given as follows,

$$\Phi''_{f_j^t}(x) = f_j^t \cdot \frac{e^{-x}}{(1 - e^{-x})^2}.$$

We see that the second derivative is positive, $\Phi''_{f_j^t}(x) > 0$, if the parameter f_j^t is positive, $f_j^t > 0$, so in this case the function $\Phi_{f_j^t}(x)$ is strictly convex. If the parameter f_j^t equals zero, $f_j^t = 0$, the function $\Phi_{f_j^t}(x)$ is linear, $\Phi_{f_j^t}(x) = x$, and therefore convex as well. ■

We also introduce a joint shape prototype vector $\boldsymbol{w} := [\boldsymbol{w}_1^\top \cdots \boldsymbol{w}_K^\top]^\top$ that is obtained by concatenating all individual shape prototype vectors $\boldsymbol{w}_k,\ k \in \{1, \ldots, K\}$ (c.f. Fig. 5.4). The component $\boldsymbol{w}_{k,j'}$ equals the negative logarithm of the background probability of pixel j' in the normal shape prototype $\boldsymbol{w}_k$,

$$\boldsymbol{w}_{k,j'} = -\log\big(1 - P_k(f_{j'}^t = 1)\big). \tag{5.12}$$

The following lemma establishes a relationship between the foreground explanation term $J_j(\mathbf{z})$, the parsing indicator $\mathbf{z}$ and the joint shape prototype vector $\boldsymbol{w}$.

Lemma 5.3.3 *The foreground explanation term $J_j(\cdot)$ is the sum over all pixels j of convex functions $\Phi_{f_j^t}(\cdot)$ whose argument is a bilinear function of the parsing indicator $\mathbf{z}$ and the joint shape prototype $\boldsymbol{w}$,*

$$J_j(\mathbf{z}) = \sum_j \Phi_{f_j^t}\big(\boldsymbol{w}^\top \mathbf{C}_j \mathbf{z} + c_0\big). \tag{5.13}$$

The parameter matrices $\mathbf{C}_j$ and scalar c_0 do not depend on the parsing indicator $\mathbf{z}$ or joint shape prototype $\boldsymbol{w}$.

Proof of Lemma 5.3.3 The foreground explanation $J_j(\{o_h, m_h\}_h)$ depends on all hypotheses that cover pixel j,

$$\begin{aligned}
&J_j(\{o_h, m_h\}_h) = \\
&\quad \sum_j \Big\{ -(1 - f_j^t) \log P(f_j^t = 0 | \{o_h, m_h, l_h\}_h) \\
&\quad - f_j^t \cdot \log\big(1 - P(f_j^t = 0 | \{o_h, m_h, l_h\}_h)\big) \Big\} \\
&\quad = \sum_j \Phi_{f_j^t}\big(-\log P(f_j^t = 0 | \{o_h, m_h, l_h\}_h)\big).
\end{aligned}$$

The argument of the function $\Phi_{f_j^t}(\cdot)$ in the last equation is bilinear in the parsing indicator $\mathbf{z}$ (Eq. 5.9) and the joint shape prototype vector $\boldsymbol{w}$ (Eq. 5.12),

$$\begin{aligned}
&-\log P(f_j^t = 0 | \{o_h, m_h, l_h\}_h) \\
&= -\log(1 - P_0) - \sum_h \log\big(1 - P(f_j^t = 1 | o_h, m_h, l_h)\big) \\
&= -\log(1 - P_0) - \sum_h \sum_k \underbrace{o_h \cdot \mathbf{1}[m_h = k]}_{= z_{h,k}} \cdot \mathbf{1}[j \in \mathcal{S}_h^t] \\
&\cdot \sum_{j'} \mathbf{1}[l_j^t = s_h^t \cdot l_{j'}^t + (x_h^t \; y_h^t)^\top] \cdot \underbrace{\log P_k(f_{j'}^t = 0)}_{=: -\boldsymbol{w}_{k,j'}} \\
&= \boldsymbol{w}^\top \mathbf{C}_j \mathbf{z} + c_0,
\end{aligned}$$

where $\mathbf{C}_j$ is a sparse matrix with following elements,

$$\mathbf{C}_j(k, j'; h, k) = \mathbf{1}[j \in \mathcal{S}_h^t] \cdot \mathbf{1}[l_j^t = s_h^t \cdot l_{j'}^t + (x_h^t \; y_h^t)^\top],$$

and the scalar c_0 has the value $c_0 = -\log(1 - P_0)$.

Thus, the foreground explanation term $J_j(\{o_h, m_h\}_h)$ can be written as

$$J_j(\mathbf{z}, \boldsymbol{w}) = \sum_j \Phi_{f_j^t}\Big(\boldsymbol{w}^\top \mathbf{C}_j \mathbf{z} + c_0\Big). \qquad \blacksquare$$

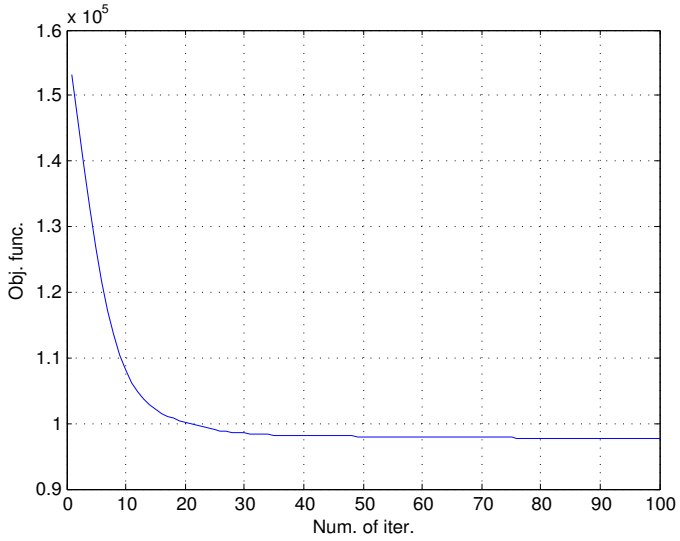


Figure 5.7: Values of the objective function $J(\mathbf{z})$ (Eq. 5.14) that are obtained as part of the convex optimization procedure that is used to solve the proposed video parsing problem.

In Lemmas 5.3.3 and 5.3.1 we expressed the foreground and hypotheses explanation terms $J_j(\cdot)$ and $J_h(\cdot)$ as convex functions of the parsing indicator $\mathbf{z}$. Therefore, the video parsing objective function $J(\cdot) := J_j(\cdot) + J_h(\cdot)$ (Eq. 5.8) is a convex function of the parsing indicator $\mathbf{z}$. To efficiently solve the optimization problem, we relax the parsing indicator $\mathbf{z}$ to the positive simplex, $\mathbf{z}_h \succeq 0$ and $\mathbf{1}^\top \mathbf{z}_h \leq 1, \ \forall h$. The last inequality follows from Eq. 5.9 and the fact that $o_h \leq 1$.

The MAP inference in the spatio-temporal video parsing model is thus equivalent to the following constrained convex optimization problem,

$$\begin{aligned} &\underset{\mathbf{z}}{\operatorname{argmin}}\, J(\mathbf{z}) = \mathbf{b}^\top \mathbf{z} + b_0 + \sum_j \Phi_{f_j^t}\big(\boldsymbol{w}^\top \mathbf{C}_j \mathbf{z} + c_0\big), \\ &\text{s.t. } \mathbf{z}_h \succeq 0 \text{ and } \mathbf{1}^\top \mathbf{z}_h \leq 1, \ \forall h. \end{aligned} \tag{5.14}$$

After finding the optimal value of the parsing indicator $\mathbf{z}$, we calculate the hypothesis indicator o_h, and the matching normal object prototype m_h of hypothesis h, as

$$o_h = \sum_{k=1}^{K} z_{h,k}, \tag{5.15}$$

$$m_h = \underset{k}{\operatorname{argmax}}\, z_{h,k}. \tag{5.16}$$

5.3.2 Solving the Convex Optimization Problem

In the previous section, we showed that the joint inference of variables $\{o_h, m_h\}_h$ can be achieved by minimizing the MAP objective function $J(\mathbf{z})$ to obtain the parsing

indicator $\mathbf{z}$ (Eq. 5.14), that belongs to the Cartesian product $\mathbf{Z} = \mathbf{Z}_h \times \cdots \times \mathbf{Z}_h$ of positive simplexes,

$$\mathbf{Z}_h = \{\mathbf{z}_h : \mathbf{z}_h \succeq 0 \text{ and } \mathbf{1}^\top \mathbf{z}_h \leq 1\}. \tag{5.17}$$

The function $J(\mathbf{z})$ is convex, smooth and bounded on the set $\mathbf{Z}$. The *projected gradient* method [CP11],

$$\mathbf{z}^{n+1} = \text{Proj}_{\mathbf{Z}}(\mathbf{z}^n - \alpha_n \nabla_{\mathbf{z}} J(\mathbf{z}^n)), \tag{5.18}$$

is used to find the global optimum of the convex optimization problem in Eq. 5.14. The projection $\text{Proj}_{\mathbf{Z}}(\cdot)$ requires each $\mathbf{z}_h$ to be projected onto the positive simplex $\mathbf{Z}_h$. The projection onto the positive simplex is calculated by applying the method of Duchi et al. [DSSSC08]. The projected gradient method finds the solution of the video parsing problem after few tens of iterations (Fig. 5.7).

5.3.3 From Inference to Abnormalities

Video parsing analyses the foreground in a video and identifies objects that have atypical appearance or behave suspiciously, to label these as abnormal. Abnormalities can also be localized on the level of pixels, where it leads to a segmentation of regions in the video that contain irregular spatio-temporal patterns. Subsequently, we see how both the object-level and pixel-level abnormalities can be detected in video, based on the inference results of the spatio-temporal video parsing approach.

Object-level abnormalities. A hypothesis h is an abnormal object, $a_h = 1$, if it is indispensable for explaining the foreground, $\bar{o}_h = 1$, but it does not have a matching normal object prototype, i.e., the best estimate $\bar{m}_h$ of a matching prototype is unlikely to explain the hypothesis (cf. Eq. 5.4),

$$\begin{aligned} &P(a_h = 1 | o_h = \bar{o}_h, m_h = \bar{m}_h) \\ &\propto \bar{o}_h P(o_h = 1|d_h) P(m_h \neq \bar{m}_h | o_h = \bar{o}_h, d_h, l_h) &(5.19)\\ &\propto \bar{o}_h P(o_h = 1|d_h) \Big(1 - P(m_h = \bar{m}_h|d_h) P(l_h|m_h = \bar{m}_h)\Big). &(5.20) \end{aligned}$$

Pixel-level abnormalities. Similarly, a pixel j is part of an abnormal object, $a_j^t = 1$, if it is in the foreground, $f_j^t = 1$, and at least one of the hypotheses that extend over this pixel, $\{h : j \in \mathcal{S}_h^t\}$, is abnormal,

$$\begin{aligned} &P(a_j^t = 1 | f_j^t, \{a_h\}_{h:j \in \mathcal{S}_h^t}) \\ &\propto f_j^t \cdot P(f_j^t = 1) \cdot \max_{h:j \in \mathcal{S}_h^t} P(a_h = 1 | o_h, m_h). &(5.21) \end{aligned}$$

5.4 Learning an Object Model for Video Parsing

Parsing query videos for abnormality detection requires an object model. We use training videos that contain a large number of normal object samples but no abnormalities to train the normal object model that consists of prototypes representing the normal

object shape, appearance, and motion. As ground truth locations of objects in the training videos are not provided, we infer them by video parsing. However, for video parsing we need to know the normal object prototypes. A standard approach for solving such a problem of mutual dependencies is *expectation-maximization* (EM) [DLR77]. Given an initial estimate of the normal object prototypes, we use them to parse the training videos, i.e. discover hypotheses that best explain the foreground and are matched to the object prototypes. Thereafter, we update the object prototypes using the matched hypotheses. We find the object model by iterating these two steps until convergence.

The goal of learning is to estimate the normal object shape prototypes $\{\boldsymbol{w}_k\}_k$ (Eq. 5.12) and their corresponding spatio-temporal descriptors $\{d_k\}_k$, $k \in \{1, \dots, K\}$. The objective function for learning is the same as for the inference (Eq. 5.14), except that it is now minimized jointly in terms of shape prototypes $\{\boldsymbol{w}_k\}_k$, their spatio-temporal descriptors $\{d_k\}_k$, as well as the parsing indicator $\mathbf{z}$ (Eq. 5.9),

$$\begin{aligned} &\underset{\{d_k, \boldsymbol{w}_k\}_k, \mathbf{z}}{\operatorname{argmin}} \; J(\mathbf{z}, \{d_k, \boldsymbol{w}_k\}_k) = J_h(\mathbf{z}, \{d_k\}_k) + J_j(\mathbf{z}, \{\boldsymbol{w}_k\}_k), \\ &\text{s.t. } \boldsymbol{w}_k \succeq 0, \; \forall k, \; \mathbf{z}_h \succeq 0 \text{ and } \mathbf{1}^\top \mathbf{z}_h \leq 1, \; \forall h. \end{aligned} \tag{5.22}$$

The hypotheses explanation term $J_h(\cdot)$ is a function of the parsing indicator $\mathbf{z}$ and the spatio-temporal descriptors $\{d_k\}_k$,

$$J_h(\mathbf{z}, \{d_k\}_k) = \beta \sum_h \sum_k z_{h,k} \Delta(d_h, d_k) + \tilde{\mathbf{b}}^\top \mathbf{z} + b_0, \tag{5.23}$$

where the parameters $\tilde{\mathbf{b}}$ and b_0 do not depend on the parsing indicator $\mathbf{z}$ or the spatio-temporal descriptors $\{d_k\}_k$ (see the proof of Lemma 5.3.1).

From Eq. 5.13 we see that the foreground explanation term $J_j(\cdot)$ depends in a convex way on both the parsing indicator $\mathbf{z}$ and the joint shape prototype vector $\boldsymbol{w}$.

Procedure for the object prototype learning. We now explain the algorithm used for solving the optimization problem of Eq. 5.22:

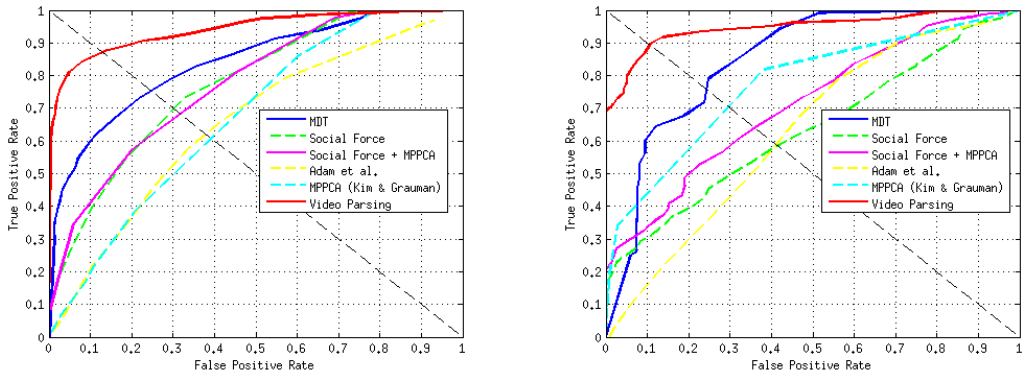
Video Parsing. Given the object prototypes, we parse the training videos to infer the parsing indicator $\mathbf{z}$ (Eq. 5.14) that yields the hypothesis indicator o_h for each hypothesis h, and its corresponding normal object prototype m_h (Eq. 5.15 and 5.16).

Updating object prototypes. We estimate the shape prototypes $\{\boldsymbol{w}_k\}_k$ and their spatio-temporal descriptors $\{d_k\}_k$ from the results of video parsing. As hypotheses overlap in training videos, the corresponding shape prototypes become mutually dependent and thus need to be learned jointly. We estimate the joint shape prototype vector $\boldsymbol{w}$ by the following convex optimization,

$$\boldsymbol{w} = \underset{\tilde{\boldsymbol{w}} \succeq 0}{\operatorname{argmin}} \, J_j(\mathbf{z}, \tilde{\boldsymbol{w}}) = \sum_j \Phi_{f_j^t}(\tilde{\boldsymbol{w}}^\top \mathbf{C}_j \mathbf{z} + c_0). \tag{5.24}$$

The convex optimization problem of Eq. 5.24 can be solved efficiently by the projected gradient method that we used for solving the MAP inference problem (Eq. 5.18),

$$\boldsymbol{w}^{n+1} = \text{Proj}_{\mathbb{R}_+^{|w|}}(\boldsymbol{w}^n - \alpha_n \nabla_{\boldsymbol{w}} J_j(\mathbf{z}, \boldsymbol{w}^n)). \tag{5.25}$$



(a) (b)

Figure 5.8: (a) Frame-wise abnormality labeling on the UCSD *ped1* dataset. Performance measures AUC and EER given in Tab. 5.1 are calculated from the ROC curves. (b) Frame-wise abnormality labeling for the UCSD *ped2* dataset.

The spatio-temporal descriptors $\{d_k\}_k,\ k \in \{1, \ldots, K\}$ are estimated separately for each normal object prototype,

$$d_k = \underset{\tilde{d}_k}{\operatorname{argmin}} \sum_h z_{h,k} \Delta(d_h, \tilde{d}_k). \tag{5.26}$$

In case of a squared Euclidean distance function, $\Delta(d_h, d_k) = \|d_h - d_k\|^2$, there is a closed-form solution for d_k, given as an average of spatio-temporal descriptors d_h of those hypotheses that are matched to prototype k by video parsing,

$$d_k = \frac{\sum_h z_{h,k} d_h}{\sum_h z_{h,k}}. \tag{5.27}$$

The algorithm assumes uniform location and velocity distributions (Eq. 5.3) for normal object prototypes. However, after the algorithm is converged, we estimate the prototype's location and velocity distributions from matched object hypotheses by the non-parametric *Parzen* windows [Bis06].

Initialization. To start the algorithm, we need an initial estimate of the normal object model. After background subtraction, some foreground segments correspond to isolated normal objects that can be used to initialize the spatio-temporal object prototypes. However, foreground/background segmentation produces also many foreground segments which correspond to interacting objects (doublets, triplets etc.). These segments are more complex and can be analyzed only by video parsing. Consequently, we need to infer which of the training foreground segments correspond to isolated normal objects and estimate object prototypes based upon them. We observe that isolated normal objects create compact clusters in the feature space. On the other hand, segments that are mixtures of two or more objects are diverse and spread out in the feature space. To detect isolated normal objects, we cluster all the foreground segments and then select

Table 5.1: Performance measures on the UCSD *ped1* dataset

	frame-wise		pixel-wise partial		pixel-wise full	
	AUC (%)	EER (%)	AUC (%)	RD (%)	AUC (%)	RD (%)
Social force [MOS09]	67.5	31	19.7	21	-	-
MPPCA [KG09]	59	40	20.5	18	-	-
Social force + MPPCA	67	32	21.3	28	-	-
Adam [ARSR08]	65	38	13.3	24	-	-
Sparse [CYL11]	86	19	46.1	46	-	-
LSA [SC12]	92.7	16	-	-	-	-
SCL [LSJ13]	91.8	15	63.8	59.1	-	-
MDT [MLBV10]	81.8	25	44.1	45	-	-
HMDT CRF [LMV13]	-	17.8	66.2	64.8	82.7	74.5
SVP [AO11]	91	18	75.6	68	83.6	77
STVP	**93.9**	**12.9**	**80.3**	**75.2**	**84.2**	**79.5**

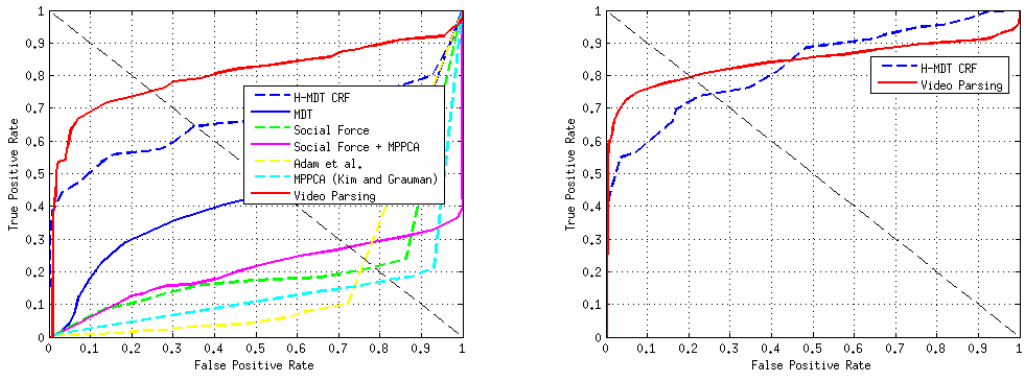
compact clusters in the feature space that correspond to isolated objects. We use *Ward's* method for agglomerative clustering to minimize the variance of clusters. Normal object prototypes are then computed as the centers of compact clusters.

5.5 Creating Spatio-Temporal Object Hypotheses

To initialize video parsing, we need a shortlist of spatio-temporal object hypotheses h (Sect. 5.2). A spatio-temporal hypothesis h consists of a sequence of object candidates in individual frames that are linked temporally. In this section we explain a method for producing per-frame object candidates and group them temporally based on their motion to obtain the shortlist of Sect. 5.2. Thereafter, we explain how to fill-in per-frame candidates that were missed during temporal grouping.

Temporal grouping of per-frame object candidates. To detect per-frame object candidates, we apply an inverted background detector that is trained to distinguish background patterns from everything else. The inverted background detector is trained on background and normal foreground segments obtained from training videos by background subtraction. The discriminative appearance-based classifier retains in each frame the object candidates that are least likely to be background. The standard non-maximum suppression (NMS) then removes some of the candidates based on the overlap criteria. The discriminative classifier is trained using a linear SVM [CL11] with per-frame features of Eq. 4.31 extracted from background/foreground segments of training videos.

We then employ agglomerative clustering to perform a temporal grouping of candidates. This yields spatio-temporal hypotheses h, which are sequences of per-frame



(a) (b)

Figure 5.9: (a) Pixel-wise abnormality prediction evaluated by the partially annotated UCSD *ped1* dataset. (b) Pixel-wise abnormality prediction that is evaluated using the full annotation of the complete UCSD *ped1* dataset that we have assembled. In all of these cases the spatio-temporal video parsing significantly improves upon the state-of-the-art, which can also be seen from the corresponding AUC and RD values provided in Tab. 5.1 and 5.2.

candidates. As usual, the clustering starts with singleton clusters (each candidate being a cluster). Then, in each round of the recursive clustering, those groups of per-frame object candidates which are most similar based on their motion and which do not share the same frames are grouped. The motion of a candidate is represented by the set of trajectories obtained by tracking the edge points inside the support region of a candidate. For tracking the feature points we use optical flow vectors that are previously computed by the method of [LFAW08]. We now define similarity of two object candidates as the ratio of the number of feature point trajectories that are shared by two candidates over the total number of trajectories in two candidates. As the result of temporal grouping, we obtain a shortlist of spatio-temporal hypotheses h.

Filling-in missing candidates by Kalman filter. The inverted background detector used for producing object candidates in each frame typically has a number of missed detections. These are the frames in which none of the object candidates is associated with a hypothesis h. We fill-in the missed object detections with the contextual help of other per-frame candidates that belong to the same hypothesis h. Therefore, the location of a missed object candidate l_h^t at time t is estimated from the available object candidate locations at times $\{t_1, t_2, \dots\}$ by a non-causal Kalman filter.

The shortlist of object hypotheses established by temporal grouping has a high recall at the cost of low precision. By maximizing the recall, the shortlist includes all relevant hypotheses, while still maintaining a reasonable total number thereof (about one hundred). Since hypotheses are created by bottom-up grouping, there will, however, be many spurious hypotheses that can only be eliminated by video parsing.

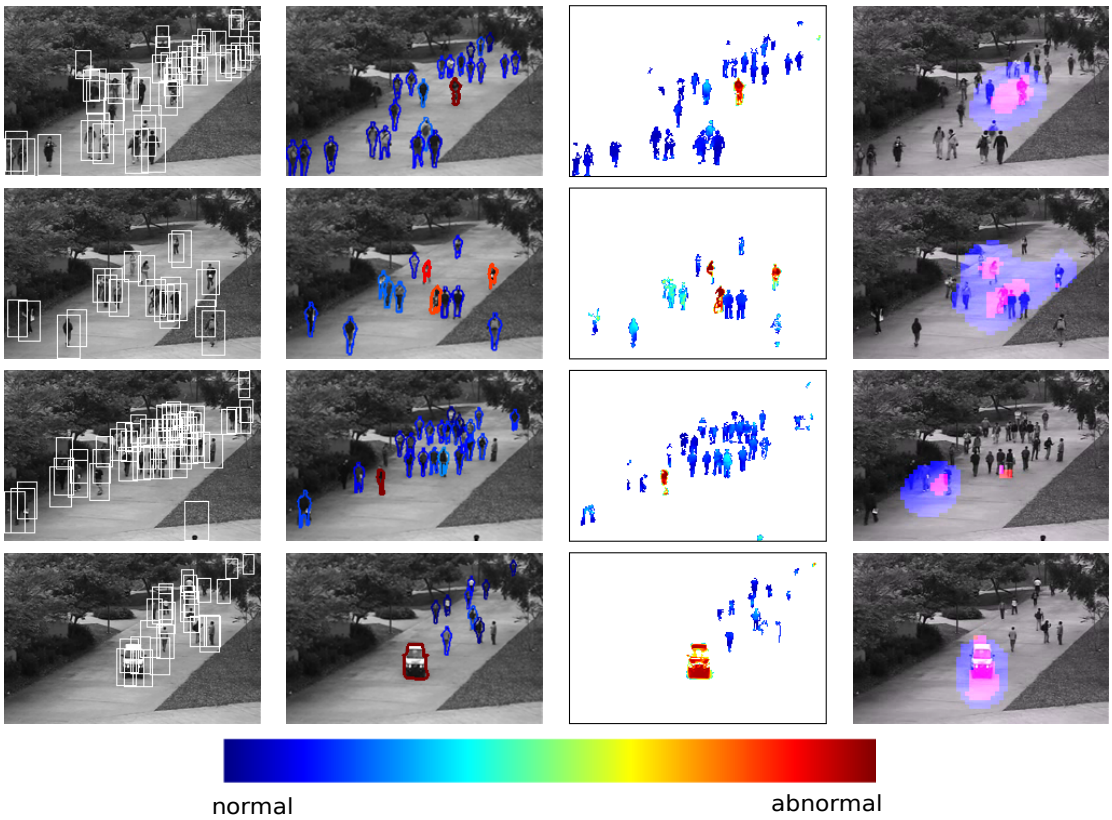


Figure 5.10: Rows show results on different frames of the UCSD *ped1* benchmark. Column i) the initialization of the video parsing by a shortlist of object hypotheses, column ii) hypotheses selected by video parsing with the best matching shape prototype colored according to abnormality probability $P(a_h^t = 1)$, column iii) foreground pixel abnormality probabilities $P(a_j^t = 1)$, column iv) results by the H-MDT CRF approach [LMV13]. Best viewed in color.

5.6 Experimental Evaluation

We use three standard state-of-the-art benchmark sets for evaluating the spatio-temporal video parsing approach and comparing its performance to the other state-of-the-art methods. We first analyze the detection results of the spatio-temporal video parsing on the UCSD benchmark sets *ped1* and *ped2*, then we present additional results on the UMN benchmark set. We apply the standard evaluation protocol of the datasets.

5.6.1 Evaluation on the UCSD Anomaly Datasets

Fig. 5.10 compares the abnormality localization of our video parsing to the H-MDT CRF method [LMV13] on UCSD *ped1* test videos. The first row shows a person riding a

Table 5.2: Performance measures on the UCSD ped2 dataset

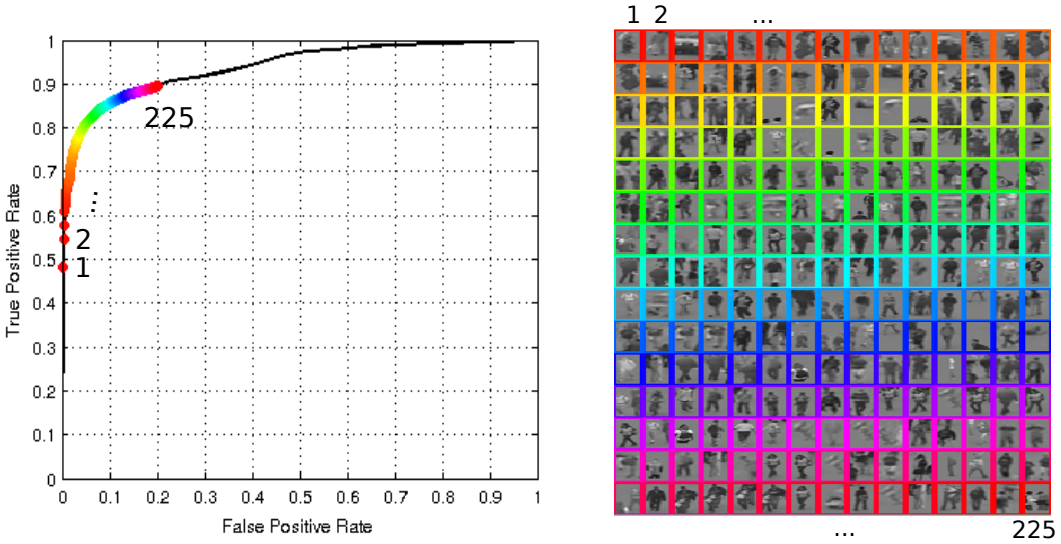
	frame-wise		pixel-wise	
	AUC (%)	EER (%)	AUC (%)	RD (%)
Social force [MOS09]	63	42	-	-
MPPCA [KG09]	77	30	-	-
Social force + MPPCA	71	36	-	-
Adam [ARSR08]	63	42	-	-
MDT [MLBV10]	85	25	-	-
H-MDT CRF [LMV13]	-	18.5	-	70.1
SVP [AO11]	92	14	-	-
STVP	**94.6**	**10.6**	**81.1**	**78.8**

bike in a group of walking persons. In the second row there are three abnormalities in the scene: a person riding a bike, and two persons running along the walkway. The third row shows a person skateboarding along the walkway, and the fourth row shows an unusual object (car) in the scene. The columns show: (i) initial hypotheses of video parsing, (ii) hypotheses selected by video parsing, (iii) abnormality localization results of video parsing, (iv) abnormality localization results of H-MDT CRF method [LMV13]. Due to our learned normal shape model used for explaining the foreground, we achieve better localization of the abnormalities in videos.

In Fig. 5.12 we show more examples of the video parsing on UCSD *ped1* test videos. Row 1 shows two persons skateboarding and cycling on a very crowded walkway, row 2 a skateboarder in a group of pedestrians, and row 3 two cyclists and a person walking across the walkway. By comparing the first two columns one can see that most hypotheses from the shortlist are discarded by video parsing because they get statistically explained away.

We also compare quantitatively the spatio-temporal video parsing approach to the state-of-the-art methods on the challenging UCSD *ped1* and *ped2* benchmarks [MLBV10]. The methods used in our comparison are the mixture of dynamic textures (MDT) [MLBV10], H-MDT CRF [LMV13], social force model (SF) [MOS09], mixture of optical flow (MPPCA) [KG09], optical flow method (Adam et al.) [ARSR08], SF+MPPCA [MLBV10], sparse reconstruction (Sparse), local statistical aggregates (LSA) [SC12], sparse combination learning (SCL) [LSJ13], and sequential video parsing (SVP) [AO11] which parses video frames individually, one after another. We denote by STVP the full spatio-temporal video parsing proposed in this chapter.

Our study shows that video parsing outperforms all other methods in experiments on both UCSD *ped1* and *ped2* datasets. Fig. 5.8(a) shows ROC curves for the frame-wise labeling of the UCSD *ped1* set. Tab. 5.1 gives the performance measures for the *ped1* dataset. We see that the inclusion of the temporal component and the improved in-



(a) (b)

Figure 5.11: Analysis of the false positive instances generated by the spatio-temporal video parsing on the UCSD *ped1* dataset. Instances are sorted in the decreasing order of their abnormality score.

ference enables spatio-temporal video parsing to improve upon our previous sequential video parsing by 2.9% in AUC and 5.1% in EER. From Tab. 5.1 we also see that the spatio-temporal video parsing improves upon recently proposed powerful methods such as LSA [SC12] (1.2% gain in AUC and 3.1% in EER) as well as SCL [LSJ13] (2.1% gain in AUC and EER). All ROC plots for the pixel-wise labeling on *ped1* are shown in Fig. 5.9(a) and 5.9(b). For the partial pixel-wise labeling of *ped1*, the spatio-temporal video parsing achieves an improvement of 4.7% AUC and 7.2% RD over the sequential video parsing. We outperform the closest competitor (HDMT CRT [LMV13]) by 14.1% in AUC and 10.4% in RD. For the full pixel-wise labeling of *ped1*, we achieve an improvement of 2.5% in RD over the sequential video parsing. The competing HMDT CRF [LMV13] method we outperform in this case by 1.5% in AUC and 5.0% in RD.

The ROC curves for the frame-wise labeling of UCSD *ped2* are given in Fig. 5.8(b). The numerical results are given in Tab. 5.2. We observe an improvement in performance of spatio-temporal parsing over sequential parsing by 2.6% in AUC and 3.4% in EER. The best method so far, MHDT CRF [LMV13], we improve upon by 6.9% in EER. For the pixel-wise labeling of *ped2* dataset, we outperform the competing HMDT CRF method by 8.7% RD (AUC values for HMDT CRF are not provided in [LMV13]). Overall we see that our spatio-temporal reasoning and the convex optimization based inference yield a significant improvement over the state-of-the-art.

Due to temporal grouping of per-frame object candidates (Sect. 5.5), spatio-temporal video parsing requires significantly less hypotheses (only about a hundred for the whole spatio-temporal domain) than sequential video parsing [AO11], which needs

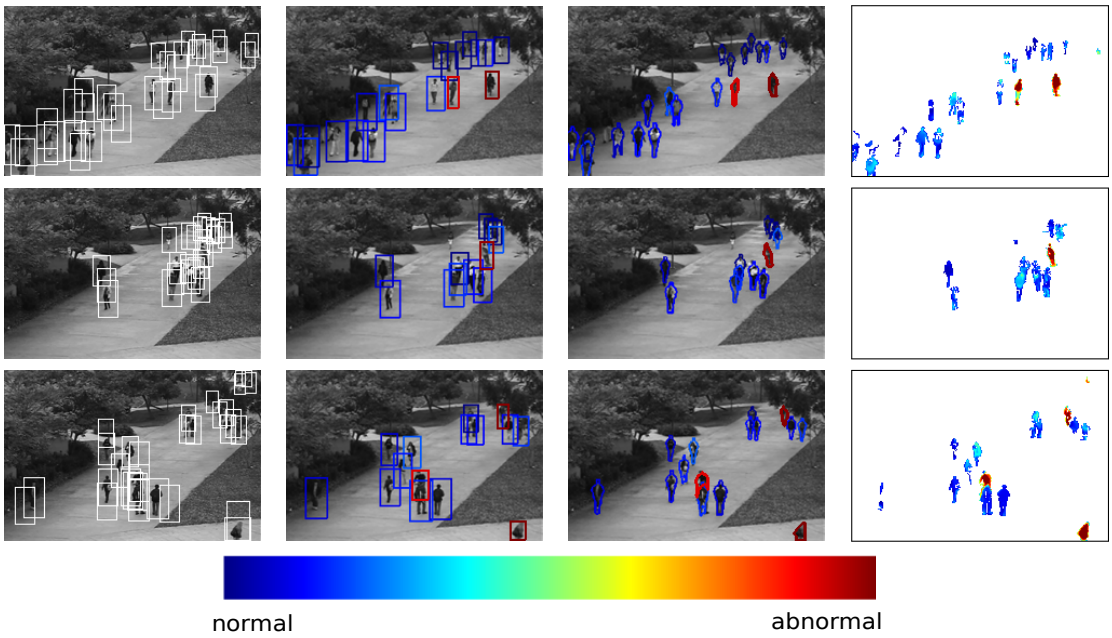


Figure 5.12: Additional results of video parsing on the UCSD *ped1* dataset. Rows correspond to different examples. The first, third and fourth column correspond to the first three columns of Fig. 5.10. The second column shows hypotheses that are selected from the shortlist by video parsing. Other hypotheses are discarded by explaining away using the selected hypotheses. Best viewed in color.

the same number of hypotheses for representing single frames. Since there remain fewer hypotheses to process, spatio-temporal video parsing takes less time to execute than sequential video parsing. Our non-optimized Matlab implementation on a Dual-Core 2.7GHz CPU runs at about 1 fps, whereas our previous sequential video parsing took 5-10 secs per frame. This is on par with recent H-MDT CRF [LMV13] and Sparse [SC12] methods, with a notable exception of extremely fast SCL method [LSJ13].

Analysis of False Detections

To get a full understanding of the detection performance of proposed video parsing, we analyze the false detections on the UCSD *ped1* dataset. In Fig. 5.11 we see the first 225 false detections sorted in the decreasing order of their probability of abnormality. We observe several reasons for false detections: i) In many cases, false detections appear as a result of artifacts in the foreground segmentation. In such cases, wrongly segmented pixels cannot be explained by the learned shape model and thus they are classified as abnormal. ii) Large variability of the normal human gait can sometimes be interpreted in video parsing as abnormal (e.g. running vs. fast walking). iii) Seldom errors in the provided video annotation cause that correctly detected abnormalities are sometimes considered as false (e.g. cars or running persons in Fig. 5.11). iv) When the true-positive hypothesis is missing from the shortlist due to a non-maximal recall, video

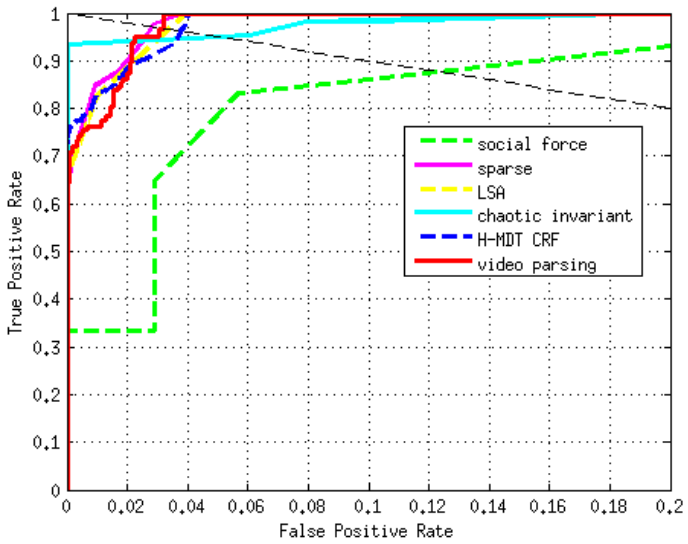


Figure 5.13: Abnormality detection on the UMN dataset: ROC curves for frame-wise labeling.

parser can select an incorrect hypothesis as a next best fit.

5.6.2 Evaluation on the UMN Anomaly dataset

We additionally evaluate the spatio-temporal video parsing on the UMN dataset that is widely used for benchmarking abnormality detection. The UMN dataset consists of three scenes in which periods of normal activity are followed by periods of emergency that are staged by people in the scene. In normal cases people are walking around alone or in groups. However, in emergency cases people start in panic to run away. For each scene several normal and abnormal events are happening one after another. In scene one, two and three there are two, six and three abnormal events, respectively. The dataset does not provide pixel-wise ground-truth abnormality maps, so we follow the standard protocol for this dataset and evaluate the detection results only in a frame-wise manner. Fig. 5.13 shows ROC curves for the frame-wise labeling. The performance measures AUC and EER are given in Tab. 5.3. For scene one, our performance is on par with the best competing methods in terms of AUC (99.5%) and EER(3.2%). For scene two we achieve 97.5% AUC that is equal to the best performing method (Sparse [CYL11]). For the scene three we achieve 99.9% AUC that improves upon the best competitor (Sparse [CYL11]) by 3.5%. A qualitative comparison of our method to HMDT CRF [LMV13] on two frames is shown in Fig. 5.14 . We see that our method achieves best localization of abnormalities that is consistent with findings from earlier experiments on UCSD *ped1* and *ped2*.

Table 5.3: Performance measures on the UMN dataset

	AUC (%)	EER (%)
chaotic invariants [WMS10]	99.4	5.3
social force [MOS09]	94.9	12.6
LSA [SC12]	**99.5**	3.4
H-MDT CRF [LMV13]	**99.5**	3.7
Sparse [CYL11] (scene1)	**99.5**	-
Sparse [CYL11] (scene2)	**97.5**	-
Sparse [CYL11] (scene3)	96.4	-
STVP (scene1)	**99.5**	3.2
STVP (scene2)	**97.5**	6.2
STVP (scene3)	**99.9**	**1.5**

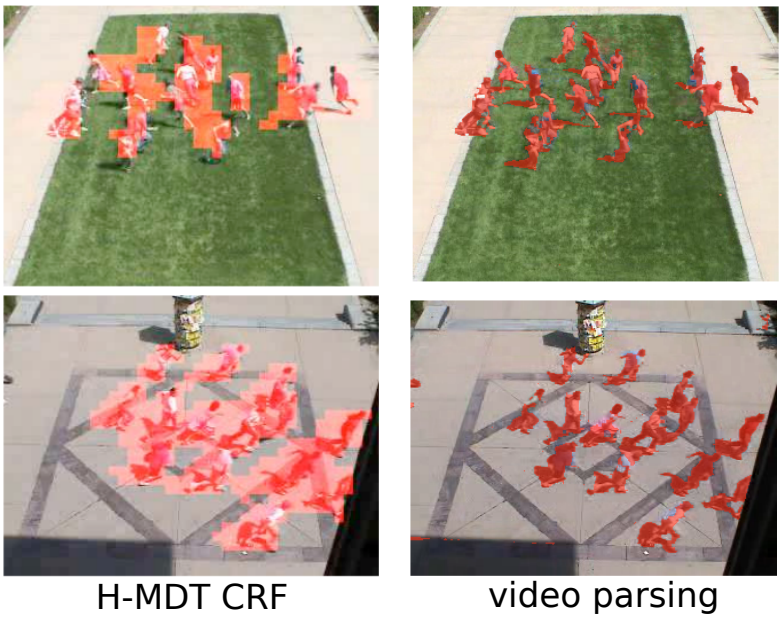


Figure 5.14: Abnormality detection on the UMN dataset: Detection results of the H-MDT CRF [LMV13] (left column) and video parsing (right column). Our approach exhibits competitive performance as can also be seen from the corresponding AUC and EER statistics in Tab. 5.3.

5.7 Discussion

This chapter introduced the full spatio-temporal video parsing. Abnormalities are detected by searching for a set of spatio-temporal object hypotheses that jointly explain the video foreground and which are themselves explained by normal training samples. Video parsing does not independently detect individual hypotheses, but infers their joint spatio-temporal layout that collectively describes objects in the scene. MAP inference in a graphical model is transformed to a convex optimization problem and efficiently solved. The full spatio-temporal video parsing is evaluated on several challenging benchmark sets, which showed that the proposed method advances the state-of-the-art both in terms of abnormality detection and pixel-wise localization.

Chapter 6

Conclusion and Discussion

This thesis proposed latent structured representations which significantly improved action recognition and abnormality detection in videos and contributed to better video understanding. The thesis developed latent variable models that represent unknown action instances or abnormal/normal object instances in video as hypotheses that are jointly inferred within the model. Multiple-instance learning is used to discover latent action instances in a set of action subsequence hypotheses extracted from weakly labeled training videos. For abnormality detection in videos, a set of latent object hypotheses is created and by probabilistic inference a subset of hypotheses is found that best explains the foreground information in videos. Novel methods are developed that increase efficiency of the latent hypotheses creation and inference and robustness of classifier training in proposed latent variable models.

In this thesis, a fundamental issue of widely used multiple-instance learning is addressed. In the underlying optimization algorithm, training of classifier and inference of latent instance labels are typically performed on the same training samples. That leads to overfitting and increases variance of label estimates. This is avoided by introducing the concept of superbags (ensemble of bags of bags). Superbags allow to effectively decouple the inference and learning processes by performing them on different training samples. Experiments on standard benchmark sets for multiple-instance learning show that the concept of superbags consistently improves several widely used approaches to multiple-instance learning after it is integrated into the optimization routine.

The thesis proposed a method based on multiple-instance learning to automatically infer the latent action instances as subsequences extracted from weakly labeled training videos and jointly train an action classifier on them. To increase robustness of the multiple-instance learning, the thesis proposed a method for creating action subsequence hypotheses in each video whose number and length are sequentially reduced until only one subsequence remains in a video. The fusion of the automatically trained subsequence classifier and the full sequence classifiers led to an improvement over the state-of-the-art in action recognition as evaluated on the challenging Hollywood2 benchmark set. The detection of action subsequences is also evaluated on two categories of the Hollywood2 benchmark set which achieved favorable performance.

A significant part of the thesis dealt with the problem of abnormality detection,

where sequential and spatio-temporal methods for video parsing were proposed to circumvent the ill-posed problem of directly searching for abnormalities in individual local image patches or regions. Abnormalities are detected by searching for a set of latent object hypotheses that jointly explain the video foreground and which are themselves explained by normal training samples. All object hypotheses that are needed to explain the foreground of a video are jointly inferred. At the same time, each hypothesis seeks to be explained by a normal training example. In proposed probabilistic graphical models, sets of hypotheses are jointly explaining the foreground while they are also able to explain each other away, simultaneously. Thus, hypotheses are not detected individually but their layout is jointly discovered. Abnormalities are then indirectly detected as hypotheses which are needed to explain the scene but which themselves cannot be explained by the normal training samples.

In the method for sequential video parsing, a single video frame is processed at the time. Only spatial interactions between competing hypotheses in a single frame are exploited to find indispensable hypotheses in the shortlist. Inference is performed as the standard local search in a graphical model. Video parsing is then extended to the spatio-temporal domain. The spatio-temporal video parsing method is proposed to jointly parse of all frames in a video. This methodological extension allowed to resolve both spatial and temporal dependencies between object hypotheses in the scene. MAP inference in the video parsing graphical model is used to find all indispensable object hypotheses in video and it is solved efficiently by convex optimization. The video parsing approaches are evaluated on several challenging benchmark sets, and they showed improved performance over the state-of-the-art.

In conclusion, the thesis verified the premise that the use of latent structured models can significantly improve the performance of action recognition and abnormality detection in videos that is crucial for video understanding. Despite the lack of action or abnormality instance labels is training videos, the missing information is discovered by selecting from a larger set of instance hypotheses using inference in latent variable models. Therefore, techniques for efficient inference and learning in latent structure models are developed in this thesis in addition to novel latent representations of the actions and abnormalities that are proposed. In future, the work on latent structured modeling will be continued in the direction of complex activity or event representations in videos.

Publications

This dissertation has led to the following scientific publications:

- Antic, B., and Ommer, B. Video Parsing for Abnormality Detection. In IEEE International Conference on Computer Vision (ICCV), 2011.
- Antic, B., and Ommer, B. Robust Multiple-Instance Learning with Superbags. In Asian Conference on Computer Vision (ACCV), 2012.
- Antic, B., Milbich, T., and Ommer, B. Less is More: Video Trimming for Action Recognition. In IEEE International Conference on Computer Vision (ICCV), Workshop on Understanding Human Activities: Context and Interactions, 2013.

Under submission is currently the following publication:

- Antic, B., and Ommer, B. Spatio-temporal Video Parsing for Abnormality Detection. Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI).

Bibliography

[AMO13] Borislav Antić, Timo Milbich, and Björn Ommer. Less is more: Video trimming for action recognition. In *ICCV (HACI)*, pages 1–8, 2013.

[AO11] Borislav Antić and Björn Ommer. Video parsing for abnormality detection. In *ICCV*, pages 2415–2422, 2011.

[AO12] Borislav Antić and Björn Ommer. Robust multiple-instance learning with superbags. In *ACCV*, pages 242–255, 2012.

[ARSR08] Amit Adam, Ehud Rivlin, Ilan Shimshoni, and David Reinitz. Robust real-time unusual event detection using multiple fixed-location monitors. *PAMI*, 30:555–560, 2008.

[AT08] N. Ahuja and S. Todorovic. Connected segmentation tree: A joint representation of region layout and hierarchy. In *CVPR*, pages 1–8, 2008.

[ATH03] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems 15*, pages 561–568. MIT Press, 2003.

[Bar12] David Barber. *Bayesian Reasoning and Machine Learning*. Cambridge University Press, New York, NY, USA, 2012.

[BC06] Gabriel J. Brostow and Roberto Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *IEEE Computer Vision and Pattern Recognition*, pages I: 594–601, 2006.

[BGS08] Arslan Basharat, Alexei Gritai, and Mubarak Shah. Learning object motion patterns for anomaly detection and improved object detection. In *CVPR*, pages 1–8, 2008.

[BI05] Oren Boiman and Michal Irani. Detecting irregularities in images and in video. In *ICCV*, pages 462–469, 2005.

[Bis06] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[BJS11] Yannick Benezeth, Pierre-Marc Jodoin, and Venkatesh Saligrama. Abnormality detection using low-level co-occurring events. *Pattern Recognition Letters*, 32(3):423–431, 2011.

[BM98] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, COLT 98, pages 92–100, New York, NY, USA, 1998. ACM.

[Bre96] Leo Breiman. Bagging predictors. *Mach. Learn.*, 24(2):123–140, 1996.

[BVDB11] Boris Babenko, Nakul Verma, Piotr Dollar, and Serge Belongie. Multiple instance learning with manifold bags. In *International Conference on Machine Learning*, 2011.

[BYB11] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, August 2011.

[CBW06] Yixin Chen, Jinbo Bi, and James Z. Wang. Miles: Multiple-instance learning via embedded instance selection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(12):1931–1947, 2006.

[CDF+04] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[CL11] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):1–27, 2011.

[CP11] Patrick L. Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Springer Optimization and Its Applications, pages 185–212. Springer New York, 2011.

[CYL11] Yang Cong, Junsong Yuan, and Ji Liu. Sparse reconstruction cost for abnormal event detection. In *CVPR*, pages 3449–3456. IEEE, 2011.

[DF10] T. Deselaers and V. Ferrari. A conditional random field for multiple-instance learning. In *ICML*, June 2010.

[DH04] Hannah Dee and David Hogg. Detecting inexplicable behaviour. In *BMVC*, pages 477–486, 2004.

[DL97] Thomas G. Dietterich and Richard H. Lathrop. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71, 1997.

[DLR77] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[DRCB05] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005.

[DSSSC08] John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, ICML, pages 272–279, New York, NY, USA, 2008. ACM.

[FE73] Martin A. Fischler and R.A. Elschlager. The representation and matching of pictorial structures. *Computers, IEEE Transactions on*, pages 67–92, 1973.

[FFFPZ05] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google's image search. In *Proceedings of the 10th International Conference on Computer Vision, Beijing, China*, volume 2, pages 1816–1823, 2005.

[FGMR10] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1627–1645, 2010.

[FL07] Sanja Fidler and Ales Leonardis. Towards scalable representations of object categories: Learning a hierarchy of parts. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2007.

[FR97] Nir Friedman and Stuart Russell. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, UAI, pages 175–181, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[FRKZ11] Zhouyu Fu, Antonio Robles-Kelly, and Jun Zhou. Milis: Multiple instance learning with instance selection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):958–977, 2011.

[GBS+05] Lena Gorelick, Moshe Blank, Eli Shechtman, Michal Irani, and Ronen Basri. Actions as space-time shapes. In *In ICCV*, pages 1395–1402, 2005.

[GC07] Peter V. Gehler and Olivier Chapelle. Deterministic annealing for multiple-instance learning. *Journal of Machine Learning Research - Proceedings Track*, 2:123–130, 2007.

[GHS11] Adrien Gaidon, Zaid Harchaoui, and Cordelia Schmid. Actom Sequence Models for Efficient Action Detection. In *IEEE Conference on Computer Vision & Pattern Recognition*, Colorado Springs, États-Unis, June 2011.

[GIB11] Andrew Gilbert, John Illingworth, and Richard Bowden. Action recognition using mined hierarchical compound features. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):883–897, May 2011.

[HGX09] Timothy Hospedales, Shaogang Gong, and Tao Xiang. A markov clustering topic model for mining behaviour in video. In *ICCV*, 2009.

[HLGX11] Timothy M. Hospedales, Jian Li, Shaogang Gong, and Tao Xiang. Identifying rare and subtle behaviors: A weakly supervised joint topic model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2451–2464, 2011.

[HlT12] Minh Hoai and Fernando De la Torre. Max-margin early event detectors. In *CVPR*, pages 2863–2870, 2012.

[JRL13] Mehrsan Javan Roshtkhari and Martin D. Levine. An on-line, real-time learning method for detecting anomalies in videos using spatio-temporal compositions. *Comput. Vis. Image Underst.*, 117(10):1436–1452, October 2013.

[JYTK11] Fan Jiang, Junsong Yuan, Sotirios A. Tsaftaris, and Aggelos K. Katsaggelos. Anomalous video event detection using spatiotemporal context. *Computer Vision and Image Understanding*, 115(3):323–333, 2011.

[KG09] J. Kim and K. Grauman. Observe locally, infer globally: A space-time mrf for detecting abnormal activities with incremental updates. In *CVPR*, pages 2921–2928, 2009.

[KMSZ10] Alexander Kläser, Marcin Marszalek, Cordelia Schmid, and Andrew Zisserman. Human focused action localization in video. In Kiriakos N. Kutulakos, editor, *ECCV Workshops (1)*, volume 6553 of *Lecture Notes in Computer Science*, pages 219–233. Springer, 2010.

[KN09] L. Kratz and K. Nishino. Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 0:1446–1453, 2009.

[KY09] I. Kokkinos and A. Yuille. Hop: Hierarchical object parsing. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:802–809, 2009.

[Lap05] Ivan Laptev. On space-time interest points. *Int. J. Comput. Vision*, 64(2-3):107–123, September 2005.

[LDXT11] Wen Li, Lixin Duan, Dong Xu, and Ivor Wai-Hung Tsang. Text-based image retrieval using progressive multi-instance learning. In *ICCV*, pages 2049–2055, 2011.

[LFAW08] C. Liu, W. T. Freeman, E. H. Adelson, and Y. Weiss. Human-assisted motion annotation. In *CVPR*, pages 1–8, 2008.

[LMSR08] Ivan Laptev, Marcin Marszałek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *Conference on Computer Vision & Pattern Recognition*, jun 2008.

[LMV13] Weixin Li, Vijay Mahadevan, and Nuno Vasconcelos. Anomaly detection and localization in crowded scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(PrePrints):1, 2013.

[Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[LS10] Fuxin Li and Cristian Sminchisescu. Convex multiple-instance learning by estimating likelihood ratio. In *Advances in Neural Information Processing Systems*, pages 1360–1368, 2010.

[LSB10] Christian Leistner, Amir Saffari, and Horst Bischof. Miforests: multiple-instance learning with randomized trees. In *Proceedings of the 11th European conference on Computer vision: Part VI*, ECCV'10, pages 29–42, Berlin, Heidelberg, 2010. Springer-Verlag.

[LSJ13] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *International Conference on Computer Vision (ICCV)*, 2013.

[LWM11] Tian Lan, Yang Wang, and Greg Mori. Discriminative figure-centric models for joint action localization and recognition. In *International Conference on Computer Vision (ICCV)*, 2011.

[LXG10] Chen Change Loy, Tao Xiang, and Shaogang Gong. Stream-based active unusual event detection. In *ACCV*, 2010.

[LYT09] C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing: Label transfer via dense scene alignment. In *CVPR*, pages 1972–1979, 2009.

[LZYN11] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 3361–3368, Washington, DC, USA, 2011. IEEE Computer Society.

[MBM08] Subhransu Maji, Alexander C. Berg, and Jitendra Malik. Classification using intersection kernel support vector machines is efficient. *IEEE Conference on Computer Vision and Pattern Recognition*, 0:1–8, 2008.

[MCB10] Julian John McAuley, Tibério S. Caetano, and Wray L. Buntine. Graphical models. In *Encyclopedia of Machine Learning*, pages 471–479. 2010.

[ME09] Tomasz Malisiewicz and Alexei A. Efros. Beyond categories: The visual memex model for reasoning about object relationships. In *NIPS*, 2009.

[MLBV10] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *CVPR*, 2010.

[MO12] Antonio Monroy and Björn Ommer. Beyond bounding-boxes: Learning object shape by model-driven grouping. In *ECCV (3)*, pages 580–593, 2012.

[MOS09] Ramin Mehran, Alexis Oyama, and Mubarak Shah. Abnormal crowd behavior detection using social force model. *CVPR*, pages 935–942, 2009.

[Mur12] Kevin P Murphy. *Machine learning: a probabilistic perspective*. Cambridge, MA, 2012.

[MW05] Olvi L. Mangasarian and Edward W. Wild. Multiple instance classification via successive linear programming. Technical Report 05-02, Data Mining Institute, 2005.

[NwCFf10] Juan Carlos Niebles, Chih wei Chen, and Li Fei-fei. Modeling temporal structure of decomposable motion segments for activity classification. In *in Proc. 11th European Conf. Comput. Vision, 2010*, pages 392–405, 2010.

[OMB09] Björn Ommer, Theodor Mader, and Joachim M. Buhmann. Seeing the objects behind the dots: Recognition in videos from a moving camera. *International Journal of Computer Vision*, 83(1):57–71, 2009.

[PPLA01] Aleksandra Pizurica, Wilfried Philips, Ignace Lemahieu, and M Acheroy. *The application of Markov Random Field Models to wavelet-based Image denoising.*, pages 43–70. Imaging and Vision Systems : Theory, Assessment and Applications. Vol.9 : Advances in Computation : Theory and Practice. - Huntington : NOVA Science Publishers, 2001. 2001.

[RB07] Stefan Roth and Michael J. Black. On the spatial statistics of optical flow. *International Journal of Computer Vision*, 74, 2007.

[RLBB10] P.M. Roth, C. Leistner, A. Berger, and H. Bischof. Multiple instance learning from multiple cameras. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 17 –24, june 2010.

[Ros98] Kenneth Rose. Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. In *Proceedings of the IEEE*, pages 2210–2239, 1998.

[SC12] Venkatesh Saligrama and Zhu Chen. Video anomaly detection based on local statistical aggregates. In *CVPR*, pages 2112–2119. IEEE, 2012.

[SCZ11] Florian Schroff, Antonio Criminisi, and Andrew Zisserman. Harvesting image databases from the web. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33:754–766, 2011.

[SG99] Chris Stauffer and W Eric L Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 2. IEEE, 1999.

[SLC04a] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36 Vol.3, Aug 2004.

[SLC04b] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004.

[SS01] Bernhard Scholkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[SSP+11] René Schuster, Samuel Schulter, Georg Poier, Martin Hirzer, Josef A. Birchbauer, Peter M. Roth, Horst Bischof, Martin Winter, and Peter Schallauer. Multi-cue learning and visualization of unusual events. In *ICCV Workshops*, pages 1933–1940, 2011.

[TCYZ05] Zhuowen Tu, Xiangrong Chen, Alan L. Yuille, and Song-chun Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of Computer Vision*, 63(2):113–140, 2005.

[TFLB10] Graham W. Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler. Convolutional learning of spatio-temporal features. In *Proceedings of the 11th European Conference on Computer Vision: Part VI*, ECCV'10, pages 140–153, Berlin, Heidelberg, 2010. Springer-Verlag.

[UPL10] Muhammad Muneeb Ullah, Sobhan Naderi Parizi, and Ivan Laptev. Improving bag-of-features action recognition with non-local cues. In *Proceedings of the British Machine Vision Conference*, pages 95.1–95.11. BMVA Press, 2010.

[Vap95] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[VG08] Sudheendra Vijayanarasimhan and Kristen Grauman. Keywords to visual categories: Multiple-instance learning for weakly supervised object categorization. In *CVPR*, 2008.

[web14] YouTube website. Statistics, 2014.

[WGR+09] John Wright, Arvind Ganesh, Shankar Rao, YiGang Peng, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *NIPS*, pages 2080–2088, 2009.

[WKSCL11] Heng Wang, Alexander Kläser, Cordelia Schmid, and Liu Cheng-Lin. Action Recognition by Dense Trajectories. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3169–3176, Colorado Springs, United States, June 2011.

[WKSL13] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, May 2013.

[WMG07] Xiaogang Wang, X. Ma, and Eric Grimson. Unsupervised activity perception by hierarchical bayesian models. In *CVPR*, page 45, 2007.

[WMG09] Xiaogang Wang, X. Ma, and W.E.L. Grimson. Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *PAMI*, 31:539–555, 2009.

[WMS10] Shandong Wu, Brian E. Moore, and Mubarak Shah. Chaotic invariants of lagrangian particle trajectories for anomaly detection in crowded scenes. In *CVPR*, pages 2054–2060, 2010.

[WTG08] Geert Willems, Tinne Tuytelaars, and Luc Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 650–663, Berlin, Heidelberg, 2008. Springer-Verlag.

[WUK+09] Heng Wang, Muhammad Muneeb Ullah, Alexander Kläser, Ivan Laptev, and Cordelia Schmid. Evaluation of local spatio-temporal features for action recognition. In *British Machine Vision Conference*, page 127, September 2009.

[XG05] Tao Xiang and Shaogang Gong. Video behaviour profiling and abnormality detection without manual labelling. In *ICCV*, pages 1238–1245, 2005.

[XG08] Tao Xiang and Shaogang Gong. Incremental and adaptive abnormal behaviour detection. *CVIU*, 2008.

[YGC13] Wanqi Yang, Yang Gao, and Longbing Cao. Trasmil: A local anomaly detection framework based on trajectory segmentation and multi-instance learning. *Comput. Vis. Image Underst.*, 117(10):1273–1286, October 2013.

[YHMU12] Koichiro Yamaguchi, Tamir Hazan, David McAllester, and Raquel Urtasun. Continuous markov random fields for robust stereo estimation. In *ECCV*, 2012.

[ZG01] Qi Zhang and Sally A. Goldman. Em-dd: An improved multiple-instance learning technique. In *In Advances in Neural Information Processing Systems*, pages 1073–1080. MIT Press, 2001.

[ZNRC13] Yingying Zhu, Nandita M. Nayak, and Amit K. Roy-Chowdhury. Context-aware activity recognition and anomaly detection in video. *J. Sel. Topics Signal Processing*, 7(1):91–101, 2013.

[ZNW08] Tao Zhao, Ramakant Nevatia, and Bo Wu. Segmentation and tracking of multiple humans in crowded environments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(7):1198–1211, 2008.

[ZSV04] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *CVPR*, pages 819–826, 2004.