Objects for Spatio-Temporal Activity Recognition in Videos

Pascal Mettes
This book was typeset by the author using latex.

Printing: Off Page

Cover design: Morris Franken

Copyright © 2017 by P. Mettes.
All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission from the author.

ISBN 978-94-6182-851-4
Objects for Spatio-Temporal Activity Recognition in Videos

ACADEMISCH PROEFSCHRIJT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. K.I.J. Maex
ten overstaan van een door het college voor promoties
ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op dinsdag 5 december 2017 te 10:00 uur

door

Pascal Simon Maria Mettes

door

geboren te Heemskerk
The work described in this thesis has been carried out within the graduate school ASCI, dissertation number 381, at the Intelligent Sensory Information Systems lab of the University of Amsterdam. The research is supported by the STW STORY project.
## CONTENTS

1 Introduction ........................................................................................................... 9

1.1 List of publications ......................................................................................... 13

2 No Spare Parts: Sharing Part Detectors for Image Categorization ............... 15

2.1 Introduction .................................................................................................. 15

2.2 Related work ................................................................................................. 17

2.3 Part sharing for image categorization ............................................................ 18

2.3.1 Part representation ................................................................................... 19

2.3.2 Part boosting ............................................................................................ 19

2.3.3 Bootstrap fusion ....................................................................................... 21

2.4 Experiments .................................................................................................. 22

2.4.1 Datasets .................................................................................................. 22

2.4.2 Implementation details ............................................................................. 23

2.4.3 Experimental evaluation ........................................................................... 24

2.5 Conclusions ................................................................................................. 32

3 Bag-of-Fragments: Selecting and Encoding Video Fragments for Event Detection and Recounting ........................................... 33

3.1 Introduction .................................................................................................. 33

3.2 Related work ................................................................................................. 34

3.3 Bag-of-Fragments ......................................................................................... 35

3.3.1 Fragment proposals .................................................................................. 36

3.3.2 Fragment selection ................................................................................... 37

3.3.3 Video encoding ......................................................................................... 38

3.3.4 Concept filtering ..................................................................................... 38

3.4 Experimental setup ....................................................................................... 39

3.4.1 Experiments ............................................................................................. 39

3.4.2 Implementation details ............................................................................. 41

3.5 Experimental results ..................................................................................... 42

3.5.1 Fragment proposal quality ......................................................................... 42

3.5.2 Event detection ........................................................................................ 43

3.5.3 Event recounting ...................................................................................... 44

3.6 Conclusions ................................................................................................. 45

4 The Imagenet Shuffle: Reorganized Pre-Training for Video Event Detection 47

4.1 Introduction .................................................................................................. 47

4.2 Related work ................................................................................................. 48

4.2.1 Event detection with pre-trained networks ............................................. 48

4.2.2 (Re)organizing hierarchies for events ..................................................... 49

4.3 Reorganized pre-training ............................................................................. 50
4.3.1 Bottom-up reorganization ........................................ 50
4.3.2 Top-down reorganization ....................................... 52
4.4 Experimental setup .................................................. 53
  4.4.1 Dataset ......................................................... 53
  4.4.2 Implementation details ....................................... 53
4.5 Experiments .......................................................... 54
  4.5.1 Bottom-up operation parameters .............................. 54
  4.5.2 Standard versus reorganized pre-training ..................... 55
  4.5.3 Fusing representations and modalities ....................... 56
  4.5.4 Comparison to the state-of-the-art .......................... 57
4.6 Conclusions ........................................................... 59

5 SPOT ON: ACTION LOCALIZATION FROM POINTLY-SUPERVISED PROPOSALS 61

  5.1 Introduction ......................................................... 61
  5.2 Related work ....................................................... 62
  5.3 Action localization using cheap annotations .................... 63
    5.3.1 Action class labels and pointly-supervision .............. 64
    5.3.2 Overlap between points and proposals ..................... 64
    5.3.3 Mining proposals overlapping with points .................. 65
  5.4 Experimental setup ................................................ 66
    5.4.1 Datasets ...................................................... 66
    5.4.2 Implementation details ..................................... 66
  5.5 Results ............................................................. 67
    5.5.1 Training without ground truth tubes ....................... 67
    5.5.2 Lowering the annotation frame-rate ......................... 69
    5.5.3 Hollywood2Tubes: Action localization for Hollywood2 .... 71
    5.5.4 Comparison to the state-of-the-art ........................ 72
  5.6 Conclusions ........................................................ 73

6 LOCALIZING ACTIONS FROM VIDEO LABELS AND PSEUDO-ANNOTATIONS 75

  6.1 Introduction ......................................................... 75
  6.2 Related work ....................................................... 76
  6.3 Action localization with Pseudo-annotations ................... 77
    6.3.1 Pseudo-annotations ......................................... 78
    6.3.2 Computing pseudo-annotation overlaps ..................... 79
    6.3.3 Correlation metric for pseudo-annotations ............... 79
  6.4 Experimental setup ................................................ 80
    6.4.1 Datasets ...................................................... 80
    6.4.2 Implementation details ..................................... 81
  6.5 Experimental results .............................................. 81
    6.5.1 Evaluating the pseudo-annotations ........................ 81
    6.5.2 Combining pseudo-annotations ............................... 82
    6.5.3 Pseudo-annotations at test time ............................ 84
    6.5.4 Comparison to state-of-the-art ............................. 85
  6.6 Conclusions ........................................................ 85
# Spatial-Aware Object Embeddings for Zero-Shot Localization and Classification of Actions

## 7.1 Introduction

## 7.2 Related work

### 7.2.1 Supervised action localization / classification

### 7.2.2 Zero-shot action localization / classification

## 7.3 Spatial-aware object embeddings

### 7.3.1 Prior knowledge

### 7.3.2 Scoring actor boxes with object interaction

### 7.3.3 Linking spatial-aware boxes

## 7.4 Local and global object interaction

### 7.4.1 Scoring videos with global objects

### 7.4.2 Spatial- and global-aware embedding

## 7.5 Experimental setup

### 7.5.1 Datasets

### 7.5.2 Implementation details

## 7.6 Experimental results

### 7.6.1 Spatial-aware embedding properties

### 7.6.2 Local and global object interaction

### 7.6.3 Spatial-aware action retrieval

### 7.6.4 Comparison to state-of-the-art

## 7.7 Conclusions

## 8 Conclusions

### 8.1 Thesis summary

### 8.2 General conclusions

## 9 Samenvatting

## 10 Acknowledgements

### A Hollywood2Tubes

#### A.1 Annotation protocol

#### A.2 Annotation statistics

#### A.3 Annotation examples

Bibliography
INTRODUCTION

It was Hemingway who stated that we should “never mistake motion for action”. It has since become a common English saying, typically used in a literary, political, or motivational setting. There is however also a visual truth to the statement. Consider the front cover of this thesis. The design shows an example for the activities changing a car tire, and bowling. The examples provide no motion over time. Still, it is evident which activities are occurring without considering the actors, simply due to the presence of interacting objects and objects in the background. Objects provide a clear cue about which activities are occurring.

In the human visual system, we quickly realize to look beyond motion when learning to recognize and perform activities. Within the first year of life, children gain the ability to segment and detect objects [23, 116, 145]. When conceptual skills grow, children learn to categorize activities based on a complex interplay of spatial relations among objects [5]. Similarly, we are eventually able to quickly decompose and distinguish activities by examining the objects in the scene [145]. Hence within the human visual system, static objects serve as the building blocks towards learning to recognize dynamic activities.

Activity recognition is a crucial capability of the human visual system and automatic recognition by machines is therefore a long standing challenge in computer vision and multimedia retrieval. Automatically recognizing and performing activities allows machines to interact with their environment. Take the task of autonomous driving. A self-driving car should recognize what, when, and where certain objects and activities are present around the car. In turn, the car should act accordingly to prevent accidents and other mishaps. Equivalently in surveillance, the ability to recognize activities will help to identify potential threats in public spaces, help identify poachers that hunt for endangered animals, or aid elderly care by detecting whether senior citizens have fallen or have forgotten to take their medicine. In short, recognizing and performing activities are at the core of human-machine interactions.

Most current works on automatic activity recognition disregard Hemingway’s statement and fixate on motion to represent activities in videos. Traditionally, activities are represented with local motion information, as detailed in various surveys on activity recognition [128, 170, 187]. Recent works have investigated motion representation on a larger scale using deep networks for activities in videos [33, 41, 148]. To gain insight into activities beyond predicting global video labels, two trends in activity recognition have emerged. The first trend is to precisely localize activities in both space and time in videos, visualized by the red tubes in Figure 1. Since the motion of activities needs to be measured more precisely in localization, as the desired activity needs to be explicitly isolated from the rest of the scene, extensive manual supervision is currently required in
Figure 1.: This thesis focuses on the automatic recognition of activities (golf swinging and horse riding) and their spatio-temporal location (red tube). We investigate how the use of object likelihoods and locations (green tube and blue tube), helps to identify what, when, and where activities occur in visual content.

The form of box annotations in each video frame [65, 135, 159]. While such annotations provide precise information about activities, datasets are scarce and small-scale, which hampers a deeper analysis into the spatio-temporal extent of activities.

The second trend in activity recognition is to learn from fewer pre-defined activity labels. Partially, this second trend is a result of the first trend; given the extensive supervision currently required for localization, fewer unique videos can be annotated. Another motivation for learning from fewer pre-defined examples is the desire to find and localize activities on-the-fly, i.e. learn quickly from user-specified queries [14, 98]. Learning from few or even no visual examples enables a fast generalization to virtually any activity, but is currently still a far cry away from systems that require lots of pre-defined activity examples [14, 62]. In this thesis, we address both trends in activity recognition by broadening the scope of activities beyond motion. We investigate how object likelihoods and locations, illustrated by the green and blue tubes in Figure 1, can help to both specify when and where activities occur and can help specifying which activities occur when activity labels are scarce.

We are empowered by progress in automatic object recognition, building on high-dimensional representations [141, 176] and deep neural networks [78, 149, 164]. Machines are able to automatically recognize objects on even the most diverse datasets [26] given enough data to learn from. Beyond object recognition, several recent approaches have shown to effectively detect objects locally in images [132, 134]. We are reaching an era where object recognition is evolving from research topic to application. Several works have provided initial links between objects and activities in visual content [63, 97, 100, 206]. The evolution within object recognition allows us to deeper investigate the link between objects and activities on both a local and global level.

Akin to how infants learn to recognize activities and entrusted by the current state of automatic object recognition, we investigate what, when, and where specific activities occur in visual content by examining object representations. The main research question is posed as:

What do objects tell about the extent of activities in visual space and time?
In Chapter 2, we first investigate the recognition of actions from a spatial perspective, when no temporal information is available. We perform recognition by learning a set of image parts that describe different activities. To determine to what spatial extent parts should be incorporated in the visual representation to recognize an activity, we formulate the following question:

**What do objects and their parts tell about the spatial extent of activities?**

A first instinct for part-based activity recognition is to rely on parts describing the activity itself [32, 69]. We hypothesize that the parts coming from the context around activities – such as interacting objects – and parts from other activities improve recognition. We analyze this hypothesis by backtracking where selected parts come from and conclude that part selection should not be done separately for each activity, but instead be shared and optimized over all activities. To incorporate part sharing between categories, we present an algorithm to optimize part sharing and selection. Evaluation shows that all part types matter for recognition and improves not only the recognition of activities in images, but also of object and scene categories in general. We conclude that the spatial extent of activities goes beyond the activity itself to include its surrounding context and even parts of other activities.

In Chapter 3, we investigate the recognition of activities temporally in videos. We focus on high-level activities describing events such as renovating a home, giving directions, and town hall meeting. We pose the following research question:

**Which object preferences matter for the temporal extent of activities?**

Consider the case of the activity birthday party. A prototypical approach for representing such activity videos from objects is to average the scores of a large set of objects over each video [54, 71, 98]. However, a video activity like birthday party has many temporal fragments, such as cutting a cake, singing, and unwrapping presents. The fragments are connected to the activity, but all have different object preferences. We propose a video representation dubbed bag-of-fragments, that incorporates the presence of multiple fragments in activity videos. We automatically discover the fragments that describe each activity from examples. Experimentally, we show that using multiple fragments with different object preferences aids activity recognition. Furthermore, by using objects, we are able to provide a temporally localized recounting of why a certain activity was recognized. This leads us to conclude that fragments composed of different object classifier scores matter for activity recognition and recounting.

In Chapter 4, we study the object representations used to help recognize event activities and pose the following question:

**What objects should represent activities?**

Object representations are typically trained on large-scale image datasets, such as ImageNet [26] or MS-COCO [89]. Where the current standard is to learn from the 1,000 classes defined in the ImageNet Large Scale Visual Recognition Challenge [26, 136], we investigate how to leverage the complete ImageNet hierarchy for pre-training deep networks. To deal with the problems of over-specific objects and objects with few images,
we introduce a bottom-up and top-down approach for reorganization of the ImageNet hierarchy based on all its 21,814 classes and more than 14 million images. Experimental evaluation shows that leveraging and reorganizing the full object hierarchy has a clear effect on the recognition of activities, indicating the importance of employing the right set of objects and hierarchical level for representing activities.

In Chapter 5, we investigate the spatial and temporal domains jointly by moving beyond recognition towards the localization of activities in videos. Before incorporating object knowledge, we tackle the primary bottleneck when training action localizers, namely the dependence on box annotations. We address this problem through the following research question:

**What do point annotations tell about the spatio-temporal extent of activities?**

An accepted standard in activity localization is to use action proposals at test time and select the best one with a classifier trained on carefully annotated box annotations [61, 118, 175]. Annotating action boxes in video is cumbersome, tedious, and error prone. In Chapter 5, we propose to annotate actions in video with points instead of box annotations. With point annotations, we maintain the spatio-temporal location of actions, but loose the spatial extent of actions. We introduce an overlap measure between action proposals and points and incorporate them into the objective of a non-convex Multiple Instance Learning optimization that trains on unsupervised action proposals directly. Through experimentation, we show that training on proposals guided by a few point annotations performs as well as training on box annotations, while being much faster to annotate.

The method and training approach from Chapter 5 using points opens up the possibility to link objects to activities in both space and time. Given the effectiveness of point annotations, we ask the research question:

**What do objects, actions, and motion tell about the spatio-temporal extent of activities?**

Where we ask users to provide point supervision in Chapter 5, in Chapter 6 we propose fully automatic visual cues that replace manual point annotations, dubbed pseudo-annotations. These pseudo-annotations include cues about objects [217] and actors [205], as well as actions [175], motion [61], and center bias [169]. We show how the pseudo-annotations can be used to localize activities and we propose a correlation metric to automatically select and combine them. Through experimentation, we show that replacing points with pseudo-annotations reaches comparable results. This indicates that auxiliary information about objects, actors, and motion can help to point out when and where certain activities occur in videos, resulting in activity localization using only video labels as activity annotations.

Finally in Chapter 7, we take the link between activities and objects to its penultimate step. We continue our goal from the last two chapters to reduce the annotation effort for activity localization and pose the research question:

**What do spatial relations of objects tell about the spatio-temporal extent of activities?**

The main contribution of Chapter 7 is a spatial-aware object embedding for activities. To arrive at spatial awareness, we build our embedding on top of freely available actor
and object detectors. Relevance of objects is determined in a word embedding space and further enforced with estimated spatial preferences. Besides local object awareness, we also embed global object awareness into our embedding to maximize actor and object interaction. Experimental evaluation shows that activity localization without training examples is possible when jointly embedding actors, objects, and their spatial relations over time.

Invariably, Chapter 7 combines the lessons learned from previous chapters. We incorporate the insight that local spatial context around activities matters (Chapter 2), that global object preferences matter (Chapter 4), and that actors and objects can help to point at activities (Chapters 5 and 6). Taking these lessons together, we conclude that objects provide valuable information about the presence and spatio-temporal extent of activities in videos.

1.1 LIST OF PUBLICATIONS

Chapter 2
Title No Spare Parts: Sharing Part Detectors for Image Categorization
Published in Computer Vision and Image Understanding, issue 152, 2016.
Author contributions
Pascal Mettes all aspects
Jan C. van Gemert guidance and technical advice
Cees G. M. Snoek supervision and insight

Chapter 3
Title Bag-of-Fragments: Selecting and Encoding Video Fragments for Event Detection and Recounting
Published in ACM International Conference on Multimedia Retrieval, 2015.
Author contributions
Pascal Mettes all aspects
Jan C. van Gemert guidance and technical advice
Spencer Cappallo helped with designing the method
Thomas Mensink guidance and technical advice
Cees G. M. Snoek supervision and insight

Chapter 4
Title The ImageNet Shuffle: Reorganized Pre-training for Video Event Detection
Published in ACM International Conference on Multimedia Retrieval, 2016.
Author contributions
Pascal Mettes all aspects
Dennis C. Koelma designed and outlined the method
Cees G. M. Snoek supervision and insight
<table>
<thead>
<tr>
<th>Chapter 5</th>
<th>Spot on: Action Localization from Pointly-Supervised Proposals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Published in</td>
<td><em>European Conference on Computer Vision, 2016.</em></td>
</tr>
</tbody>
</table>
| Author contributions | Pascal Mettes all aspects  
Jan C. van Gemert guidance and technical advice  
Cees G. M. Snoek supervision and insight |

<table>
<thead>
<tr>
<th>Chapter 6</th>
<th>Localizing Actions from Video Labels and Pseudo-annotations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accepted to</td>
<td><em>British Machine Vision Conference, 2017.</em></td>
</tr>
</tbody>
</table>
| Author contributions | Pascal Mettes all aspects  
Cees G. M. Snoek supervision and insight  
Shih-Fu Chang supervision and insight |

<table>
<thead>
<tr>
<th>Chapter 7</th>
<th>Spatial-aware Object Embeddings for Zero-shot Localization and Classification of Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accepted to</td>
<td><em>International Conference on Computer Vision, 2017.</em></td>
</tr>
</tbody>
</table>
| Author contributions | Pascal Mettes all aspects  
Cees G. M. Snoek supervision and insight |
2.1 INTRODUCTION

In this work, we aim to categorize images into their object, scene, and action category. Image categorization has been studied for decades and tremendous progress has been made [85, 141, 153, 174, 176, 211], most recently by the introduction of very deep convolutional neural networks [78, 149, 164, 210]. These networks learn to categorize images from examples by back-propagating errors through stacked layers of convolutional filters pooled over image regions. Despite their implicit local nature, deep nets result in a global scene representation only. Thereby, ignoring known benefits of explicitly encoding local image blocks, i.e. discriminative power [30, 219], image interpretation [32, 43], and complementarity [69, 219]. In this paper we make a case for sharing parts for image categorization, studying what parts to consider, which parts to select and how to share them between categories.

The notion of sharing has been well studied in data mining [133, 207]. These works repeatedly show that sharing boosts classification performance and provides connections between co-occurring elements. Motivated by these examples, we investigate sharing of parts when learning part-representations for image categorization.

Consider Figure 2, where examples of image categories sofa and horse utilize parts from their own category as well as parts from other categories and the background context. When a classifier is trained exclusively using parts from its own category, relevant information is missed [4, 124]. As illustrated in Figure 2, for object categories such as sofa, it is informative to use parts from cat and dog categories as well as parts from sofa itself [4, 124]. By giving the sofa classifier access to dog and cat training images, the recognition of sofa is improved even though these images may not contain a sofa at all. Even when global image categories differ, their representation can share similar parts and should thus be modeled as such [69, 124].

To obtain insight in part sharing, we track where a part comes from. We define three types of part-origin. Own: parts coming from the defining image category; Other: parts coming from other labeled categories; Context: parts not overlapping with labeled categories. Our analysis shows that the global image representation captures the scene; that Own parts benefit out-of-context and small images (e.g. sofa in the grass); Other parts aid for co-occurring categories (e.g. person on a horse) and Context parts help to recognize supporting structures (e.g. fence for sheep or cows).
Figure 2: Shared part responses of our method for an example of sofa (a) and horse (b). We show that image categories do not only benefit from own category part responses, but also from responses of other categories (e.g. dog for sofa and person for horse) and from context (e.g. horizontal bars for horse).

Several recent works have focused on local parts as representation for image classification [10, 30, 32, 69] by performing part selection for each category separately. These works follow a common four-step pipeline. First, a set of potentially interesting parts is proposed. Second, a selection algorithm discovers the most discriminative parts. Third, classification is performed on the part-based representations. And fourth, fusion with a global image representation. These four steps have shown to yield robust and complementary image representations [10, 30, 218]. However, by separating the part selection, category classification, and fusion steps into disjoint steps these works leave room for a number of improvements. Most notably, (1) the models do not share parts and perform part selection for each category independently, as also observed by Parizi et al. [124], (2) the models use different objective functions for the part selection step and the final classification step, and (3) the models perform part selection independently of the global representation with which they are fused. As we will show, these drawbacks result in sub-optimal image categorization results.

We make three contributions in this paper: (i) we establish that three part types are relevant for image categorization, which are all naturally shared between categories when learning a part-representation for image categorization; (ii) we present an algorithm for part selection, sharing, and image categorization based on boosting. It embeds all three part types, without the need for explicit part type definition; (iii) we introduce a fusion technique for combining part-based with global image representations. We report results competitive to the state-of-the-art on object, scene, action, and fine-grained categorization challenges, further improving over the very deep convolutional neural networks of [149, 164].

The rest of the paper is outlined as follows. In section 2, we describe related work, while section 3 outlines our proposed method. This is followed by the experimental evaluation in section 4. We draw conclusions in section 5.
2.2 RELATED WORK

The state-of-the-art in image categorization relies on deep convolutional neural networks (ConvNets) [78, 149, 164, 210]. These networks learn image feature representations in the form of convolution filters. Locality is implicitly incorporated by stacking and pooling local filter responses, resulting in increasingly larger filter responses, cumulating in a single global image representation. Two recent network architectures, the VGG network of Simonyan et al. [149] and the GoogleNet network of Szegedy et al. [164], have shown that further increasing the network depth results in state-of-the-art performance on a wide range of image categorization datasets [149, 164]. A global representation from such networks benefits from augmenting it by aggregating local ConvNets features of densely sampled image parts in a VLAD [50] or Fisher Vector [21, 204] representation. We follow such approaches and augment the global representation by using ConvNets to represent discriminative local parts.

An excellent source of part-based information for classifying object images is the response of a detector for the object at hand. In the pioneering work of Harzallah et al. [57], it is shown that an object detector improves image classification, and vice versa [29, 139]. Others improved upon this idea by adding context [156], and deep learning [121]. However, when no bounding box annotations are available, a supervised detector cannot be trained. In our method, we do not use any bounding box annotations and rely exclusively on the global image label.

In the absence of bounding box annotations, one may focus on automatically discovering discriminative parts in images. The work of Singh et al. [150] proposes an unsupervised method for finding parts, by iterative sampling and clustering large collections of HOG features, SVM training on the clusters, and assigning new top members to each cluster. Other part-based methods follow a supervised approach, e.g., using discriminative mode seeking [30], random forests [10], average max precision gains [35], or group sparsity [163] for discovering the best parts for each category separately from image-level labels. Recent work moves away from HOG features in favor of local ConvNet activations for part selection [87], based on association rule mining per category. In this work, we also leverage image-level labels for part selection. In contrast to [10, 30, 32], we also perform part selection by sharing over all categories, rather than performing selection per category. Furthermore, we optimize the part selection with the global image representation during fusion.

The work by Juneja et al. [69] shows that using image-level labels leads to better categorization performance using fewer parts than the unsupervised method of Singh et al. [150]. Their method selects parts that have the lowest entropy among categories. In effect, this limits the sharing over a few categories only. We strive to share parts as much as possible. Rather than relying on entropy for the selection, we prefer a boosting objective that optimizes sharing for all categories simultaneously.

The seminal work of Torralba et al. [168] has previously explored the idea of sharing among categories. We follow this line of work in a part-based setting. Similar to Torralba et al. [168], we opt for a boosting approach, as we can exploit the inherent feature selection and sampling methods in AdaBoost [42] for jointly finding the best parts while training the category classifiers. However, where Torralba et al. [168] learn which
parts distinguish multiple categories simultaneously from a common background, our objective is to learn what parts to share to distinguish categories from each other. We extend boosting to explicitly focus on part sharing, and propose a bootstrapping method for fusion with the global image representation.

The work of Azizpour et al. [4] generalizes latent variable models and shows how explicitly incorporating both Own (referred to as foreground) and Other (referred to as background) parts benefits objects in a binary setting. We find empirically that Own and Other parts are indeed important, in addition to parts from Context. Furthermore, we extend the scope from a binary setting to both a multi-class and multi-label setting. And finally, we introduce a method for fusion that exploits the strength of both global and part-based image representations.

Recent work of Parizi et al. [124] opts for a joint optimization of part detection and image categorization, where the part detectors and image classifiers are optimized alternatively. They show that such a joint optimization over all categories simultaneously improves categorization performance over independently optimized part-based methods, be it that their approach requires significant feature dimension reduction to be feasible. Similar to [4], Parizi et al. [124] consider Own and Other part types, where we establish the relevancy of three part types (Own, Other, and Context) for sharing among image categories and we demonstrate their effectiveness. Moreover, we use state-of-the-art ConvNet features without the need for dimension reduction. Finally, our algorithm naturally incorporates fusion with the global image representation.

In [155], Song et al. learn tree classifiers on image parts for object detection. We also rely on trees, be it that we prefer to have many of them (order of $10^3$), and next to objects, we also consider scenes, actions, and fine-grained birds.

### 2.3 PART SHARING FOR IMAGE CATEGORIZATION

In Figure 3, we provide an overview of our approach. First, we perform part representation, where parts with their features and learned detectors are extracted from training images. Second, we perform our part-based optimization based on boosting to yield a part-based classifier. Third, we perform our bootstrap fusion between the part-based classifier and a classifier trained on global image representations.
2.3 PART SHARING FOR IMAGE CATEGORIZATION

2.3.1 PART REPRESENTATION

We decompose each image into part proposals which offer a modest set of bounding boxes with a high likelihood to contain any part or object. More formally, let the function $B(X)$ map an image $X$ to $k$ part proposals, $B(X) \rightarrow \{p_1, p_2, \ldots, p_k\}$. For $n$ training images, $T = \{X_1, X_2, \ldots, X_n\}$, we extract and combine all parts as $P_{\text{train}} = \bigcup_{X \in T} B(X)$.

In our work, we opt for selective search [171], but our method applies to any part partition [18, 76, 217]. For each part $p_i$, we learn a part detector $d(\psi(p_i))$, where $\psi(p_i) \in \mathbb{R}^d$ denotes the feature representation of $p_i$. For the detector $d()$, we follow [69] and transform the part representations into linear classifiers. Each linear classifier serves as an exemplar SVM for the corresponding part representation. We use the fast exemplar SVM approach of Hariharan et al. [56] to compute the classifiers. Let $\mu$ and $\Sigma^1$ denote the mean and covariance estimated from sampled part representation, ignoring their category label. Then the detector function is defined as: $d(p) = \Sigma^{-1}(p - \mu)$.

Given a detector for each part, we represent an image $X$ by a feature vector $v_X \in \mathbb{R}^{|P|}$, where each dimension corresponds to the response of a part detector on $X$. Each dimension $j$ in our image representation vector $v_X^{(j)}$ is the best part-detector response for a training part $p_j$ in image $X$,

$$v_X^{(j)} = \max_{p \in B(x)} d(\psi(p_j)) \cdot \psi(p). \quad (2.1)$$

Our image representation has $s = |P|$ dimensions where each dimension is a part from the set $P \subset P_{\text{train}}$. Our goal is to select the set of shared parts $P$ to use as detectors over all categories.

2.3.2 PART BOOSTING

We unify part selection $P \subset P_{\text{train}}$ and image categorization by extending Adaboost [42]. Boosting minimizes the training error in an iterative manner, converging asymptotically.
to the global minimum exponential loss [42]. At each iteration $t$, a weak learner $f_t(\cdot; \Phi_t)$ selects the parts $\Phi_t$ and weights $\alpha_t$ that maximally decrease the current weighted image classification error while at the same time optimizing the part selection $P$. The classification of a test image $v_X$ is a weighted sum of the $T$ weak learners under the global constraint of using at most $s$ parts.

$$h^t(X) = \sum_{t=1}^{T} \alpha_t \cdot f_t(v_X; \Phi_t), \quad \text{s.t. } |P| \leq s,$$

where $P = \bigcup_{t=1}^{T} \Phi_t$ is the set of all selected parts. Here, the set $\Phi_t$ denotes the parts used to construct a weak learner at iteration $t$. Our formulation extends standard boosting with a global constraint $|P| \leq s$ on the number of parts. By limiting the number of parts, we aim to arrive at a discriminative part-based representation [10, 30, 69]. The convergence is dependent on the value of $s$; the higher the value, the more training iterations are required.

In the objective, the weak learners $f_t(t = 1,...,T)$ take the form of orthogonal decision trees [27]. The decision trees are constructed by selecting the part with the lowest weighted classification error at each binary split of the tree [42]. The weights of each weak learner $\alpha_t$ are computed as $\alpha_t = \frac{1}{2} \ln \left( \frac{1 - \epsilon_t}{\epsilon_t} \right)$, where $\epsilon_t$ denotes the error rate of the weak learner [42].

For all $L$ categories we have $L$ corresponding binary classifiers of Eq. 2.2. In a multi-label setting each classifier is independent; yet the set of parts $P$ are shared over all categories. In a multi-class setting we have to make a single choice per image:

$$h(X) = \arg \max_{l \in L} \frac{\sum_{t=1}^{T} \alpha^l_t \cdot f_t^l(v_X; \Phi^l_t)}{\sum_{t=1}^{T} \alpha^l_t}, \quad \text{s.t.} |P| \leq s,$$

with now $P = \bigcup_{t=1}^{T} \bigcup_{l=1}^{L} \Phi^l_t$. Note that our added constraint $|P| \leq s$ in Eq. 2.3 states that at most $s$ unique parts over all categories are used, enforcing the sharing of parts between categories. By limiting the number of parts over all categories simultaneously, each part selected during representation learning should be discriminative for as many categories as possible, i.e. should be shared between categories. We opt for a one-vs-rest objective instead of a multi-class objective [34, 138] to allow for both multi-class and multi-label image categorization.

**Maximum exploiting sampling.** We introduce a part sampling algorithm for re-using parts and sharing parts between categories, called maximum exploiting. The main idea of maximum exploiting is to gradually increase the cardinality of $P$ by balancing exploration: selection of new parts from $P_{\text{train}}$; and exploitation: reusing and sharing of previously selected parts $P$.

The idea behind Maximum Exploit sampling builds upon the work by Freund and Shapire [42], which states that the training error on a strong learner reduces exponentially with the number of non-random weak learners. Here, we use this result for joint part selection and image categorization. As long as we can train non-random weak learners on the same set of parts, we are guaranteed to reduce the training error rate. We exploit this maximally in our sampling approach.

At train iteration $t$, we force the decision tree $f_t(\cdot; \Phi_t)$ to select parts $\Phi_t$ exclusively from $P$, exploiting the shared pool. When the classification performance saturates and $P$
2.3 PART SHARING FOR IMAGE CATEGORIZATION

Figure 5.: The effect of power normalization for a synthetic set of weights from 50 training examples. On the left we show the weights before power normalization, on the right after normalization. By re-balancing the weights of the training examples, we focus on the hard examples without disregarding the other examples.

is maximally exploited, we allow a single exploration step, selecting new parts $\Phi_t$ from $P_{\text{train}} \setminus P$. In Fig. 4 we illustrate our algorithm. Our sampling scheme minimizes the cardinality of the selected parts $P$, forcing the decision trees to reuse the shared parts from all categories as long as this positively affects the empirical loss over the training examples.

To determine if $P$ is maximally exploited, we measure the saturation of $P$ by the classification error within the last range of exploitation iterations. For a category $l$, let this range start at iteration $u$ and let the current iteration be $v$. Then the error is defined as:

$$\epsilon_{l}^{u,v} = \sum_{i=1}^{N} \left[ \text{sign} \left( \sum_{t=u}^{v} \alpha_{l}^{t} \cdot f_{l}^{t}(\psi_{X_{i}}; \Phi_{l}^{t}) \right) \neq Y_{l}^{i} \right],$$

where $Y_{l}^{i} \in \{-1, +1\}$ states whether image $i$ is of category $l$. The error of Eq. 2.4 states the number of miss-classifications using the weak classifiers created in the current exploitation iterations. Let $\epsilon_{u,v}$ denote the average error over all categories within range $[u, v]$, i.e.:

$$\epsilon_{u,v} = \sum_{l=1}^{L} \frac{\epsilon_{l}^{u,v}}{L}. \quad (2.5)$$

We keep exploiting if $\epsilon_{u,v} < \epsilon_{u,v-1}$, otherwise, we perform a single exploration step and restart our exploitation evaluation, setting $u = v$.

2.3.3 BOOTSTRAP FUSION

We extend our shared part selection and sampling to a fusion with the global image representation. Instead of early or late fusion that would combine the part-based representation independently with the global image representation [30, 69, 154], we fuse the representations from the start. We propose bootstrap fusion, which jointly optimizes
part-selection and image classification by selecting those parts that gain the most from fusing with the global image representation.

In our bootstrap fusion, we start by applying our maximum exploitation on the global image representation F. Afterwards, our idea is to bootstrap the AdaBoost weights of the training images as initialization for the part representations. This bootstrapping provides information about the difficulty per training image for categorization according to the global representation. By transferring the weights to the part-based model, the part-based representation learning is forced to focus on the mistakes caused by the global representation. This fusion procedure enhances the complementary nature between the part and global representations. In our fusion approach, the dimensionality is equal to the dimensionality of the global representation and the number of selected parts $s$ in the part representation.

When transferring the weights, certain training examples have weight values $w_i$ up to several orders of magnitude higher or lower than the average weight value. The reason for this is the presence of the exponent in the weight update in AdaBoost [42]. Training examples that are consistently correctly or incorrectly classified in the initial boosted classifier will yield extreme weight values. Although we want to focus on the hard examples, we are not interested in focusing solely on these examples. To combat the high variance of the weights, we perform a power normalization step [141], followed by an $\ell_1$ normalization to make the weights a distribution again:

$$w_i = \frac{w_i^\alpha}{\sum_{j=1}^{N} w_j^\alpha}.$$  (2.6)

Throughout our experiments, we set $\alpha = \frac{1}{2}$. The use of power normalization results in a more balanced weight distribution, which in turn results in better fusion performance. Fig. 5 highlights the effect of power normalization on the weight values.

During testing, we apply the same part and feature extraction as during training. We apply our boosted classifier with the $s$ selected parts to the parts of the test image. Idem, the boosted global classifier is applied to the global image representation of the test image. The confidence scores of both classifiers are summed and the category with the highest confidence is selected as the target class.

2.4 EXPERIMENTS

2.4.1 DATASETS

In our experiments, we consider 4 datasets.

**Pascal VOC 2007** [38] consists of 9,963 images and 20 object categories, where we report on the provided trainval/test split using the Average Precision score [38].

**MIT 67 Indoor Scenes** [130] consists of 6,700 images and 67 scene categories, where we use the provided 80/20 split for each scene category [130], reporting results using multi-class classification accuracy.

**Willow Action** [25] consists of 911 images and 7 action categories, where we use the train/test split of [25], reporting with the Average Precision score.
Figure 6.: Average Precision (%) on Pascal VOC 2007 for the global image representation [164] and its combination with each type of category parts (approximated from provided bounding boxes). We conclude that all types of category parts matter.

Caltech-UCSD Birds 200-2011 [181] consists of 11,788 images and 200 fine-grained bird categories, where we use the train/test split of [181], reporting with the multi-class classification accuracy.

2.4.2 IMPLEMENTATION DETAILS

**Part extraction.** During both training and testing, we use selective search as the function $B(X)$ which splits each image into roughly 2,000 parts [171]. Due to the hierarchical nature of selective search, we observe it in fact generates parts with varying sizes, from superpixels to the complete image. Although selective search is intended for objects, we observe it in fact generates parts suitable for our purpose. On Pascal VOC 2007, we derived there are over 22 parts for each labeled object for which the part of score from Vezhnevets and Ferrari [177] is at least 0.5.

**Feature extraction.** As features, we employ a GoogLeNet convolutional neural network [164], pre-trained on 15k ImageNet categories [26]. For a given part, we rescale it to $256 \times 256$ pixels and feed it to the network, resulting in a 1,024-dimensional feature vector, which is subsequently $\ell_2$ normalized.

**Part detection.** We use the fast exemplar SVM of Hariharan et al. [56] to transform the part representations into linear classifiers. The linear classifier of a part is used for the max-pooling operation of Eq. 1. For fair comparison, the max-pooling operation is only applied to the whole image. Both [69] and our approach will improve further when we incorporate pooling over the scene layout by [85]. We sample roughly 400k part features from train images to estimate the corresponding $\mu$ and $\Sigma$ parameters.

**Part boosting.** For the weak learners, we opt for orthogonal decision trees, as AdaBoost with decision trees have shown to yield excellent categorization performance [27]. At each split, we select the part that minimizes the normalized weighted miss-classification error of the examples in the corresponding node. As shown by Bühlmann and Yu [11], AdaBoost is resistant to overfitting when using a large number of training iterations. Therefore, we use 2,000 iterations for boosting throughout our experiments for training the image classifiers.
Figure 7.: Qualitative examples of when the different part types improve/decrease the categorization performance. Shown are improvements using own (a–d), other (e–h), and context (i–l) parts. The first three examples have increased performance, the last example has decreased performance. The icon denotes the category, while the numbers state the original and new ranking position.

2.4.3 EXPERIMENTAL EVALUATION

EXPERIMENT 1: ALL PART TYPES MATTER

We first motivate our main hypothesis that part-based methods benefit from three types of category parts. We rely on the Pascal VOC 2007 dataset for this experiment. As a surrogate for each type of part, we use the object bounding boxes as part locations. For each category, we use the features from the bounding boxes of the same category as its own parts. Similarly, the features from the bounding boxes of all other categories are used as other parts. For context parts, we use the features from selective search boxes without overlap to any ground truth bounding box. As a baseline, we use the performance of the global image representation. We add the features of each part type to the global representation to evaluate their effect.

Results. We show the image categorization results for the global representation and its combination with each part type in Figure 6. As the Figure indicates, adding each of the different part types boosts the categorization performance. Not surprisingly, the use
of own parts for each category yields the best improvement (89.6% vs. 85.3% mean Average Precision (mAP)). We especially note the increase obtained for small objects such as bottle (from 55.1% to 76.3%) and potted plant (from 64.2% to 74.4%).

Surprisingly effective is the addition of other parts, with an mAP of 88.6%. For multiple categories, such as sofa, it is even more favorable to add parts from other categories over parts from their own category. More specifically, using parts from cats and dogs yields a better performance than using parts from sofa itself. This result highlights the importance of exploiting and sharing the co-occurrence of categories. Lastly, the overall performance of context parts is marginally beneficial, compared to own and other parts (86.1% mAP). For categories such as car (94.0%), motor bike (91.7%), and tv (84.1%), the use of context parts may be a good choice.

We note that experiment 1 serves to show that all part types matter for image categorization. Since we rely for this experiment on ground truth bounding boxes to approximate parts, the results should not directly be compared to other part-based methods, who all rely on the class labels only. A direct comparison will be discussed in experiment 2 and beyond.

**Qualitative analysis.** To understand why each part type matters for image categorization, we provide additional qualitative results. In Figure 7, we show test examples that significantly improve or decrease in predictive performance when each of the part types is added.

For own parts, we observe that the improvement is most significant for small objects, objects out of context, and occluded objects, as shown in Figure 7a-7c for respectively bottle, sofa, and bus. Exploiting and sharing parts from other categories is beneficial when there is a high co-occurrence between categories in the images. This pattern is shown in Figure 7e-7g, for sofa / dog, chair / dining table, and horse / person.
Figure 9.: Examples of the parts in the image where our method fires for nine object categories. Our parts focus on parts from the object itself, as well as other category parts and context. Note that our method automatically discovers when to share and not to share; dog parts are not useful for bike (e), but human parts are useful for dog (f).

Parts from context are similarly beneficial in case of a co-occurrence between the categories and contextual elements. Notable examples are buildings for boat, cat basket for cat, and fence for sheep, as shown in Figure 7i-7k.

Figure 7 furthermore shows where each part type fails. For own category parts, performance decreases e.g. when the object can not be located, as shown in Figure 7d. Other category parts fail for inconsistent co-occurrence, such as car and tv in Figure 7h. For context parts, performance decreases when context is either not present or consistent with other object categories, as shown in Figure 7l.

Figure 2 also shows that different part types focus on different elements in the scene, taking in the complete information from the image. Based on the quantitative and qualitative results of this experiment, we conclude that all part types matter for image categorization and should be exploited to improve image categorization.

EXPERIMENT 2: EVALUATING OUR PART-BASED METHOD

We evaluate our joint part selection and image categorization without box information in three steps: (i) evaluate whether we capture all part types, (ii) compare to separate part optimization, and (iii) compare our maximum exploit sampling to baseline samplers.

Do we capture all types of parts? First, we validate that our method follows our hypothesis and we ask ourselves; do we capture all part types? We perform this evaluation on Pascal VOC 2007, as the object annotations of this dataset enables such an evaluation. To validate that our method is capable of incorporating all part types, we analyze the importance of each selected part as a function of the box agreement. We run our algorithm to select a total of 500 parts across all 20 categories in Pascal VOC 2007. We note that this setting yields an mAP of 89.1%, significantly higher than the global representation and on par with the representations from the strongly supervised bounding
(a) Our method vs. the separate optimization of Juneja et al. [69].

Figure 10.: Classification results on MIT 67 Indoor Scenes. In 10a, we compare our method to the separate optimization of [69]. In 10b, we compare the maximum exploit to baseline AdaBoost samplings. In both experiments, our approach is preferred over the baselines.

boxes used in the first experiment. For each selected part $p$, we compute its importance in a single decision tree as the normalized miss-classification reduction; this value is summed over all the decision tree where the part is used. For the selected part $p$ and the best overlapping box $b$, the box agreement is computed here as $\frac{p \cap b}{p}$ [177]. Intuitively, the box agreement states to what extend $p$ is part of $b$.

The relation between part importance and box agreement is shown in Figure 8. The Figure shows two clear peaks. The leftmost peak indicates that our method utilizes parts with no overlap to ground truth boxes, i.e. context parts. The other peak is at a box agreement of 1; when a part is contained in an object. The red and blue bars indicate that each category uses parts from its own object and from other objects. From the Figure we conclude that our method uses own (blue), other (red), and context (green) parts.

We also provide part responses by our method in Figure 9. The Figure shows how our method deals with occluded objects (9a), small objects (9b), and atypical object views (9c). The primary focus is on the visible portions of the objects and co-occurring other objects, but contextual information is also taken into account. Examples include car jack (car, 9a), steam (train, 9d), and horizontal bars (horse, 9g). Our method automatically discovers that people parts should be shared for dog (9f), but dog parts should not be shared for bike (9e).

Joint vs. separate selection and categorization. Second, we compare our method to the bag-of-parts of Juneja et al. [69], which applies a separate optimization of part selection and image categorization. To ensure a fair comparison, we replicate the setup described in [69]. This means that we use the same dataset (MIT 67 Indoor Scenes), the same part proposals, and the same part features. For the HOG features used in [69], we also employ the outlined iterative part refinement. For full details, we refer to [69]. We also provide a comparison using the GoogLeNet features for the parts. For the bag-of-parts, we report the results with both a linear and $\chi^2$ SVM, as is proposed in [69]. We only exclude image flipping and spatial pyramids, as these elements are not specific to either method and can therefore cloud the comparison.
(a) Pascal VOC 2007.  (b) MIT Scenes (GoogLeNet).  (c) MIT Scenes (HOG+IFV).

Figure 11.: Combination results on Pascal VOC 2007 and MIT 67 Indoor Scenes. Ours (blue) is always favored over early (red) and late (gray) feature fusion for combining part-based and global image representations.

In Figure 10a, we show the classification accuracies using respectively the HOG and GoogLeNet features. The scores the Figure are compared for four values of constraint $s$ to evaluate the influence on the performance as a function of $s$ for both our approach and the method of Juneja et al. [69]. We enforce four global part limits: 335, 670, 1675, and 3350 parts on MIT 67 Indoor Scenes and 100, 200, 500, 1000 on Pascal VOC 2007. For the baseline bag-of-parts approach, this corresponds to 5, 10, 25, and 50 parts selected per category.

For the HOG features, we observe that our joint part selection outperforms the separate optimization in the bag-of-parts of Juneja et al. [69]. The difference even grows as we use stricter part limits. At 5 parts per category, our method outperforms bag-of-parts by 7.6% ($\chi^2$) and 12.6% (linear). For the GoogLeNet features, we first observe that all results significantly outperform the HOG features (note the different ranges on the y-axis). We again outperform the bag-of-parts, albeit with a less pronounced difference. At five parts per category, we outperform the bag-of-parts by 3.6% ($\chi^2$) and 5.6% (linear). We attribute this to the strength of the features, resulting in saturation.

The run time of our approach is similar to other part-based methods, as they all compute part responses and apply learned classifiers. Naturally, the training time of our joint selection and classification is more involved than methods with separate selection and classification [30, 69], yet fast enough to avoid dimensionality reduction, as in [124]. On a single Intel Xeon E52609 core, it takes roughly 8 hours to jointly optimize all 20 objects in Pascal VOC 2007. On a test image, it takes 2 seconds to max-pool the selected parts and apply the trees.

We conclude that our joint part selection and categorization is preferred over a separate optimization.

Evaluating maximum exploiting. Third, we evaluate the maximum exploiting itself within the joint AdaBoost optimization. We reuse the HOG settings of [69] and draw a comparison to three baseline sampling strategies. The first baseline, Tieu-Viola, follows [167]. At each boosting iteration, a single new part for each category is selected until the specified part limit is reached. The second baseline, uniform, is a variant of LazyBoost [37], where we randomly sample parts from the set of all parts (ignoring whether parts have been selected or not), until the limit is reached, after which we keep retraining on the selected parts. The third baseline follows the $\epsilon$-greedy strategy, where
we have a static probability $\epsilon$ of exploring new parts and a static probability $1 - \epsilon$ of exploiting the current set of selected parts.

The comparison of maximum exploit sampling versus the baseline boosting sampling methods is shown in Figure 10b. The maximum exploiting joint optimization is preferable across the whole range of part limits. These results indicate that a higher focus on exploitation is an important aspect in the joint optimization, since the baselines have a higher focus on exploration. It is also interesting to note that, with the exception of the Tieu-Viola baseline, all variants improve over Juneja et al. [69], indicating the effectiveness of a joint optimization.

**EXPERIMENT 3: EVALUATING BOOTSTRAP FUSION.**

We evaluate our bootstrap fusion for combining our category parts with the global representation. We perform this evaluation on both Pascal VOC 2007 and MIT 67 Indoor Scenes and provide a comparison to two baseline fusions [154]. For the bootstrap procedure, we train a model on the global representations and then train our joint part selection and image categorization on the updated weights. The combination is then formed by concatenating the decision trees from both models. The first baseline, early fusion, combines the global representations and the representations from the parts selected by [69], after which a $\chi^2$ SVM is trained on the concatenated representations. The second baseline, late fusion, is similar to the bootstrap procedure, but without updating the weights. In the remaining experiments we always use an average of 50 parts per category for our approach.

**Results.** In Figure 11 we show the performance of our bootstrap fusion against the two baselines. Across all settings and part limits, our fusion (blue bars) outperforms the baselines. For Pascal VOC 2007 our method outperforms the other combiners between 2-4% across all part limits, with an mAP up to 90.7% at 50 category parts per object. For MIT 67 Indoor Scenes, the difference in performance varies for both the GoogLeNet and HOG features. For the HOG features with the Fisher Vector as the global representation, the improvement is most significant, similar to the results of Figure 10a. For the GoogLeNet features the improvement is less significant, but our bootstrap is still better than the baselines. We conclude that our bootstrap procedure is preferred when combining the part and global representations.
Comparison to the state-of-the-art. We also compare our results for the fusion to related work. The comparison of our results to the state-of-the-art on Pascal VOC 2007 is provided in Table 1. Our results provide an improvement over the references. Our final result compares favorably to the convolutional neural network applied to the whole image (90.7% vs. 85.3%), a notable improvement in mAP. We note that recent concurrent work on (very deep) convolutional networks similar in spirit to [164] has yielded an mAP of 89.3% [149], which is a 4% improvement over the network that we employ. Both our part-based and combined results still outperform [149] and we expect a further improvement when employing their network to represent the parts. In Figure 13, we highlight 3 selected parts during training on MIT 67 Indoor Scenes, along with top responses in three test images.

The comparison for MIT 67 Indoor Scenes and Willow Actions is also provided in Table 2. For scenes, we see that our results improve upon existing combinations of global and part-based representations [30, 69]. Similar to Pascal VOC 2007, we note that the recently introduced scene-specific convolutional network of Zhou et al. [216] has shown to be effective for global scene classification, with an accuracy of 70.8%. Subsequent analysis on this network has indicated that convolutional layers contain object detectors [215]. This analysis not only speaks in favor of the network for global classification, but also in our part-based setting, as this will positively influence the part representations.

Compared to Cimpoi et al. [21], we note that their results are better on MIT 67 Indoor Scenes, while we perform better on Pascal VOC 2007. This result is not surprising, as Pascal VOC 2007 is a multi-label dataset and Indoor Scenes is a multi-class dataset. As
Table 1.: Average Precision (%) scores on Pascal VOC 2007, comparing our results to related work. Our part-based results improve upon existing global representations (from 85.3% for [164] to 89.7%), while the combination with [164] yields the best results.

Table 2.: Comparison with state-of-the-art on MIT 67 Indoor Scenes, Willow Actions, and Caltech-UCSD Birds 200-2011. Here, ft denotes fine-tuning. Using a single feature only [164], our approach is competitive on all datasets and best for Pascal Objects and Willow Actions and Fine-grained Birds without fine-tuning.

From this evaluation, we conclude that our method improves upon global ConvNet image representations and yields competitive results compared to the current state-of-the-art.

a result, the objects in Pascal VOC 2007 have a higher level of co-occurrence, which is exploited by our method.

We perform an evaluation on Willow Actions, where we see that our final result of 81.7% outperforms current methods on the same dataset. In Figure 12 we show selected part for two action categories. The Figure paints a similar picture as the previous experiments. For both action categories, our method selects parts from the dominant objects in the scene (horses and instruments), but also from contextual information. For horse riding, parts such as sand and policemen are deemed discriminative, while parts from playing music include lumberjack prints from clothing.

Finally, we also provide results for the fine-grained Caltech-UCSD Birds 200-2011 dataset in Table 2. In this experiment, we exclude the use of bounding box information and only rely on the image class labels. For the results in Table 2, we use 10,000 parts (50 parts on average per category). The results show that our approach is competitive to the current state-of-the-art without the need for bounding box supervision and without any fine-grained optimizations. For example, [77, 90, 212] fine-tune their deep networks to the fine-grained images during training, which delivers a 4-5% improvement. We yield the highest scores when no fine-tuning is applied and expect a similar gain as [77, 90, 212] when fine-tuning is applied.

From this evaluation, we conclude that our method improves upon global ConvNet image representations and yields competitive results compared to the current state-of-the-art.
2.5 CONCLUSIONS

We have investigated image categorization using a representation of parts. We start from the intuition that parts are naturally shared between categories. We analyze the type of region where a part comes from, namely: parts from its own category, parts from other categories, and context parts. Experimental validation confirms that all the types of parts contain valuable and complementary information. To share parts between categories we extend Adaboost, profiting from joint part selection and image classification. Experimental evaluation on object, scene, action, and fine-grained categories shows that our method is capable of incorporating all three types of parts. Furthermore, our method provides a further boost in performance over a convolutional representation of the whole image for categorization. The strong performance of our and other state-of-the-art methods on Pascal VOC 2007, MIT 67 Indoor Scenes, and Caltech-UCSD Birds 200-2011 indicates the saturation on these datasets and the need for the community to explore bigger datasets for part-based image categorization in the near future. Our method opens up new possibilities for part-based methods, such as image captioning by sharing parts over similar words within the captions.
3

BAG-OF-FRAGMENTS: SELECTING AND ENCODING VIDEO FRAGMENTS FOR EVENT DETECTION AND RECOUNTING

3.1 INTRODUCTION

In this work, we focus on detecting events in videos and recounting why an event is relevant by providing the most relevant semantic concepts. This problem is typically addressed by globally aggregating concept detector scores over the whole video [92, 98, 100]. Global aggregation of detector scores poses two problems to event detection and recounting. First, event videos are a complex interplay of various sub-events with a varying degree of relevance [7] which are blended into a single representation. Second, in a global aggregation, the event recounting is unable to state where in the video relevant concepts occur. Similar to related work, we compute concepts scores for frames in a video as the semantic representation, but we aim to perform event detection and recounting on the level of video fragments.

We propose a pipeline to encode a video using fragments that form discriminative sub-events for a complex event, which we call bag-of-fragments. For such an encoding, we first need to generate fragments from a video. As the search space of all possible fragments in a video is vast, we propose a hierarchical clustering algorithm to yield a concise set of semantically coherent fragment proposals. The algorithm, inspired by object proposals in images [58, 171], iteratively merges only the most informative fragments. As a result, fragments are generated across all temporal locations and scales of a video, without exhaustively preserving the full search space of fragments.

Based on the fragments proposals of a set of training videos, we select the most discriminative ones of an event. Fig. 14 highlights a number of fragments with various levels of discrimination according to our selection. The selected discriminative fragments form the basis of our bag-of-fragments encoding. The discriminative fragments of an event are matched and pooled over the fragment proposals of a single video, resulting in an effective encoding for event detection. What is more, as the encoding is performed per fragment, information regarding the most informative fragments of a video is retained. Event recounting can therefore be performed by providing the most relevant concepts within the most informative fragments of a video. Lastly, we show how co-occurrence statistics from social tagged images can aid our event recounting by filtering irrelevant concepts.

Experimental evaluation on challenging web videos from the THUMOS [68] and TRECVID benchmarks [123] highlights the effectiveness of our bag-of-fragments for
Figure 14.: We propose bag-of-fragments, a video representation that finds and encodes the most discriminative fragments for event detection and recounting. The figure shows the middle frame of five fragments for three events, ordered by level of discrimination. The most discriminative fragments are exemplary for the event and will be included in the bag-of-fragments, while the more ambiguous and less discriminative fragments are ignored.

event detection and recounting. First, we show that our fragment proposals are able to retain the most informative fragments at a fraction of full search space of fragments. Second, we show the effectiveness and complementary nature of our bag-of-fragments encoding for event detection compared to a global aggregation of concept scores. Third, we show qualitatively that our bag-of-fragments with concept filtering yields desirable event recounting results.

3.2 RELATED WORK

Video encodings for event and action detection typically use low-level features that describe the spatio-temporal signal [39, 84, 183]. While these features are well-suited for recognition, they lack any semantic interpretation which complicates recounting why a video is recognized. A solution is offered by Videostory [53], that learns a representation to jointly embeds user tags with video features. The features of a new video can be mapped to this joint feature-tag space and the embedded tags allow recounting the detection evidence. Instead of user tags, which may be noisy, we use high-quality concept classifiers to allow recounting. We will experimentally compare against Videostory [53].

Instead of low-level features, recent video encodings apply a bank of concept classifiers to individual frames and average the frame-based responses for forming a video
representation [92, 98, 100]. Such a representation has shown excellent accuracy for event recognition [98, 100] with the added benefit that the concept classification scores provide valuable clues for recounting why the whole video is relevant [53, 92]. Where these works recount a complete video, we instead recount on video fragments, which offer a more precise fine-grained temporal granularity. Rather than averaging concept scores over the whole video, we aggregate scores on coherent video fragments and use them for event detection and recounting.

Although a complex event consists of various sub-events, it can be recognized by a human after seeing only a few well-chosen discriminative video fragments [7]. In automatic event recognition, fragments have been used as latent variables in an SVM optimization [162, 173]. Since latent-SVM is computationally expensive, only a limited number of latent fragments can be used. Instead, our method can exploit a larger set of possible discriminative fragments, which increases the likelihood of finding the most discriminative ones. We draw inspiration from mid-level parts as used in image classification [30, 69]; we automatically discover discriminative video fragments and use them to encode full videos as a bag of their best matching fragments.

To perform a bag-of-fragments encoding of a video, we need to first split a video into coherent fragments that are likely to contain a discriminative sub-event. Instead of a brute-force sliding window [120] or detecting shot boundaries [208] we base our fragments on proposal methods [58, 61, 171]. We propose a fast clustering method to generate a small set of fragment proposals with a high sub-event recall. We will experimentally evaluate our proposals against sliding windows and shot boundary detection.

For event recounting, the highest scoring concept scores in a video are typically used as evidence [28, 92]. Because the highest score is sensitive to noise, Sun et al. [161] propose a manually defined white-list of acceptable concepts per event. We extend this work by replacing the manual white-list with an automatically found list based on co-occurrence statistics using a high-level event description and tagged images, as recently proposed for zero-shot image classification [99]. Automatic white-listing eliminates any manual effort, which is labor intensive and may be prone to errors or subjectivity.

3.3 BAG-OF-FRAGMENTS

The key contribution of this paper is an encoding of discriminative fragments, which we call bag-of-fragments, for event detection and recounting. The pipeline consists of four major stages. The first stage generates fragment proposals by splitting a video into
Figure 16: A video of Making a sandwich, where the fragment proposals are hierarchically merged using the combined similarity. Note how semantically more correlated proposals of the video are merged earlier (indicated by the colors of the bars).

a set of fragments. The second stage performs fragment selection, by identifying the most discriminative fragment proposals for provided event examples. In the third stage we utilize the discriminative fragments to generate a bag-of-fragments encoding of a video. The encoding forms the basis for both event detection and recounting. The fourth component, concept filtering, is introduced to generate a relevant event recounting. We summarize the pipeline in Fig. 15 and detail the stages next.

3.3.1 Fragment proposals

To generate a set of fragment proposals for a video, we employ a hierarchical clustering algorithm to cluster fragments into proposals. The main idea behind the hierarchical clustering is to iteratively merge only the most informative fragments, rather than considering all fragment merges. For the clustering, we employ two similarity measures, a semantic and syntactic similarity. The two similarity measures aim to merge the most semantically similar fragments (semantic similarity), while maintaining a balanced cluster tree (syntactic similarity).

Semantic fragment similarity. Let \( f_i \in \mathbb{R}^d \) denote the semantic representation of fragment \( i \) containing the scores of \( d \) concepts. The semantic similarity between two fragments \( i \) and \( j \) is then estimated as:

\[
S_c(f_i, f_j) = \sum_{k=1}^{d} |f_i(k) - f_j(k)|, \quad (3.1)
\]

where \( f_i(k) \) denotes the \( k^{th} \) concept of \( f_i \). Using this equation as a similarity measure, the two consecutive fragments to be merged at each iteration are the fragments for which Eq. 3.1 is minimized. Such a clustering algorithm combines the semantically most similar fragments at each iteration, generating semantically coherent fragments. Updating fragments \( i \) and \( j \) into fragment \( t \) can be done efficiently, as we apply average pooling of the concepts within a fragment:

\[
\begin{align*}
  f_t(k) &= \frac{r(f_i) \cdot f_i(k) + r(f_j) \cdot f_j(k)}{r(f_i) + r(f_j)}, \\
  r(f_t) &= r(f_i) + r(f_j),
\end{align*}
\]

where \( r(f_i) \) denotes the number of frames in fragment \( i \).
3.3 Bag-of-Fragments

**Syntactic fragment similarity.** To prevent that a single video fragment gobbles up small fragments one by one, we add another similarity measure that enforces a more balanced cluster tree:

\[ S_s(f_i, f_j) = r(f_i) + r(f_j). \]  

(3.3)

The idea behind the similarity measure of Eq. 3.3 is to penalize large fragments from merging in favor of smaller fragments.

**Combined similarity.** A combination of semantic and syntactic similarities is given by a linear combination of the terms,

\[ S(f_i, f_j) = S_c(f_i, f_j) + \alpha \cdot S_s(f_i, f_j). \]  

(3.4)

As the ranges of the two similarity measures are different, the variable \( \alpha \) is set to the sum of the concept scores divided by the sum of the sizes for all consecutive fragments at each iteration. This makes both similarity measures of equal importance.

The hierarchical clustering with the combined similarity measure results in a concise set of fragment proposals, ideally retaining those fragment proposals that are semantically coherent. An example of the fragment proposals generated for a video with eleven sampled frames is shown in Fig. 16.

### 3.3.2 Fragment Selection

From the set of fragment proposals \( P \) generated for a set of event training videos, we aim to select the most discriminative ones. We utilize the training videos in two stages. In the first stage, the training videos are used to select which fragment proposals are most discriminative for a given event. In the second stage, a bag-of-fragments encoding is generated for each training and test video based on the discriminative fragments. The encodings are then used to train an event classifier with an off-the-shelf SVM classifier. During training, we are given \( N \) training videos \( X = [x_1, \ldots, x_N] \), where \( x_i \in \mathbb{R}^{s_i \times d} \) denotes the \( f^{th} \) training video containing \( s_i \) fragment proposals. Furthermore, video labels are provided as \( Y = [y_1, \ldots, y_N] \), where \( y_i \in \{-1, +1\} \) states whether training video \( i \) contains the event. We outline a three step procedure for selecting the discriminative fragments.

1) **Generating event fragment classifiers.** We first compute an event classifier for each fragment proposal of the positive event training videos. As negative examples we simply use the fragment proposals of negative videos. Rather than explicitly training an SVM classifier for each (positive) proposal separately, we prefer a faster alternative using discriminative decorrelation [56]. We assume that the maximum likelihood estimate of the covariance matrix \( \Sigma \) used in linear discriminant analysis is the sample covariance over all the fragments in the training set \( X \), ignoring class labels. As a result the linear discriminant analysis parameters \( \mu \) and \( \Sigma \) only need to be computed once over the whole training set. Then, a classifier \( w_{ij} \) for \( x_{ij} \), proposal \( j \) of training video \( i \), is efficiently computed as:

\[ w_{ij} = \Sigma^{-1}(x_{ij} - \mu). \]  

(3.5)

Eq. 3.5 results in a classifier for each fragment proposal, ready to be evaluated on all \( N \) training videos.
2) **Matching fragment classifiers for video pooling.** For each fragment proposal $p \in P$, we perform a matching to all the $N$ training videos by computing the dot-product between the event fragment classifier of $p$ and the fragment proposals of the training video. After the matching we perform a max-pooling operation, which simply retains the maximum dot-product value of the matching for the entire video. The pooling value expresses how much the training video is related to proposal $p$. By ranking the training videos according to their max-pooled values and comparing the ranking to labels $Y$, we are able to determine how well the fragment proposal is able to distinguish positive from negative videos containing a specific event. We note that each fragment proposal has a bias in the matching and pooling, namely to the video from which it has originally been retrieved. However, as the bias is equal among all the proposals, it does not lead to overfitting towards specific fragment proposals.

3) **Selecting discriminative classifiers.** From the set of all fragment proposals $P$, we aim to select a discriminative subset $F \subset P$. This is performed by selecting the fragment proposals with the best ranking scores. To avoid inclusion of visually similar and therefore redundant proposals, we enforce a constraint on each fragment proposal. The event fragment classifier of each fragment proposal should have a cosine distance of at least 0.5 with respect to the better performing proposals, otherwise, it is removed. Such a constraint results in a diverse set of discriminative fragments [69].

### 3.3.3 **VIDEO ENCODING**

Now that we have the discriminative fragments for the event, we utilize them to perform a fragment encoding for both the training and test videos. The encoding is performed with the same matching and pooling operation as in step 2 above. Let $f$ denote the number of selected discriminative fragments, i.e. $f = |F|$, then the fragment encoding results in an $f$-dimensional feature vector for a video. The number of discriminative fragments $f$ is a hyperparameter. In the experiments, we evaluate the influence of the number of selected discriminative fragments per event on the detection performance. The encoding is performed over all training and test videos.

### 3.3.4 **CONCEPT FILTERING**

Apart from a video representation for event detection, our encoding is also able to perform event recounting. To that end, the fragment proposals of the test video that have resulted in the highest max-pooled values are selected as the most informative fragments. For each of these informative fragments we select the concepts that have contributed most to the corresponding dot-product of the max-pooled value as the informative concepts. For each test video this results in a list of informative fragments and their corresponding concepts.

As indicated by Sun et al. [161], directly selecting the top scoring concepts as the recounting for each selected fragment leads to noisy results, mostly because of concept detector noise. The noise results in incorrect or irrelevant concepts in the recounting, as they were erroneously given a high score. Rather than manually determining which
concepts are relevant for an event [161], we propose to automatically filter concepts. More specifically, we use co-occurrence statistics [99] to compare the concepts used in our representation to a textual summary of the event.

As shown in Fig. 15, the concept filtering takes as input a textual summary of the event and a collection of social-tagged images from which we compute the co-occurrence statistics. Each concept in the semantic representation is compared to each concept in the textual event summary using a co-occurrence score. Intuitively, the co-occurrence statistic states that two concepts are related if they occur together relatively often as tags in the same images. As such, concepts with high co-occurrence scores are deemed relevant with respect to the event.

For the social tag dataset, we have retrieved the available subset of 4,770,156 Flickr images in ImageNet [26]. For the set of Flickr images, we have in turn collected the corresponding meta-data, in the form of 14,088,893 unique tags. Roughly 95% of the images contain multiple tags, which make the co-occurrence practical. We have made the meta-data of the Flickr images available online.

Let \( Z \) denote the set of concepts from the textual summary of the event. Then for a concept \( i \) in the semantic representation, we compute the co-occurrence score to all concepts in \( Z \) using the Dice coefficient [99] and keep the maximum score:

\[
C_i = \max_{z \in Z} \left[ 2 \frac{c_{iz}}{c_i + c_z} \right],
\]

where \( c_i \) and denotes the number of images with tag \( i \) and \( c_{iz} \) the number of images with both tag \( i \) and \( z \).

The final score \( C_i \) is used here as the relevancy score of concept \( i \) with respect to the event. For the concept filtering, we select the concepts with the highest scores according to Eq. 3.6, where the number of concepts to retain is a parameter. Finally, the event recounting is altered by only recounting the most informative concepts that are also relevant according to the concept filtering. This makes for an event recounting that is less sensitive to the noise in concept detectors and more relevant for the event, as we will show in the experiments.

### 3.4 EXPERIMENTAL SETUP

#### 3.4.1 EXPERIMENTS

**FRAGMENT PROPOSAL QUALITY**

In the first experiment, we evaluate the fragments generated by our fragment proposal algorithm. This evaluation is performed on the THUMOS’14 temporal localization dataset [68]. The dataset consists of 1010 validation videos, each containing several semantically different action-related events at different time intervals. The annotations of the events and their time intervals are provided.

https://staff.fnwi.uva.nl/p.s.m.mettes/data/imagenet-flickr-metadata.txt
Evaluation. The quality of the fragment proposals is evaluated by examining the recall ratio. For each video, we compare our fragment proposals to the ground truth fragments using the intersection-over-union [38]. Sufficient overlap is achieved if the maximum intersection-over-union is at least 0.5. The ratio of the annotated fragments with sufficient overlap forms the final score.

Baselines. We compare our fragment proposals to two baseline strategies. The first is a shot boundary detection algorithm, where a video is split into a number of non-overlapping fragments based on detected shots within the video. More specifically, we employ a graph partition model, as proposed in [208]. Here, opponent color histograms are used to represent frames and the continuity signal is thresholded to get shot detections [208]. The second baseline is a sliding window procedure, where a video is split into a number of fragments by sliding a temporal chunk across a video at all temporal positions and scales.

EVENT DETECTION

In the second experiment, we investigate the potential of the bag-of-fragments in the context of event detection. We rely on the TRECVID MED 2014 dataset [123]. This dataset consists of 20 events, where 100 positive videos are provided per event. A general set of 4991 background videos is provided as negative set.

Evaluation. We focus on two evaluations. First, we examine the effect of the number of selected discriminative fragments in our encoding and compare it to a bag-of-fragment encoding using all fragment proposals, instead of the discriminative fragments. Second, we compare our bag-of-fragments to baseline encodings and we evaluate their fusion. The performance of a single event is measured using the Average Precision (AP). At test time, roughly 24,000 videos are given for an event, where the goal is to rank the videos displaying the actual event higher than the negative videos. The quality measure on the whole dataset is in turn measured by the mean Average Precision (mAP) across all the events.

Methods. For the second evaluation, a total of four different methods are applied to the TRECVID MED 2014 dataset. The first is a global model, where the concept scores of a video are averaged over the frames [92, 98, 100]. The second is the state-of-the-art VideoStory, which uses a global model as the visual representation [53]. In addition to these two baselines we evaluate our bag-of-fragments. Finally, we consider a fusion to show the complementary power of bag-of-fragments. For each of the individual methods, the corresponding representations are $\ell_2$ normalized and fed to a linear SVM [24]. The SVM outputs are in turn converted to probability values using Platt scaling [127] and the ranking of the videos is performed on these probability values. The fusion is simply performed by computing the product of the probability values for each video.

EVENT RECOUNTING

The third experiment focuses on the recounting using the same dataset as used for event detection. We use the model trained for event detection to recount the semantic evidence
Figure 17.: Achieved recall scores as a function of the number of video fragments. Our fragment proposals yield improved recall scores using a fraction of the fragments.

of a test video, in combination with our concept filtering. We use the textual summary provided by TRECVID for each event to compute the co-occurrence scores.

Evaluation. The evaluation of the event recounting is completed on a qualitative basis. First, we examine the quality of the co-occurrence method for concept selection for 25 concepts in the semantic representation, compared to all 20 events. Second, we show the event recounting results for three test videos from different events, both with and without concept selection.

3.4.2 IMPLEMENTATION DETAILS

Semantic representation. We apply convolutional neural networks to provide the frame-based semantic representations. More specifically, we employ an in-house implementation of [210] trained on 15,293 ImageNet [26] concepts. The frames in each video are sampled once every second. For each frame, we extract the features from the third fully connected layer, the layer before the soft-max, such that the frame is represented by a 15,293-dimensional semantic vector. The aggregation of the frames in a single fragment is performed by averaging the scores per concept [98, 100].

Bag-of-Fragments. For our discriminative fragments, we first extract all the fragment proposals from the positive videos, apply θ₂ normalization on each proposal, and compute the efficient event fragment classifier. All the event fragments classifiers are max-pooled over the train videos and the top discriminative fragments are selected. For the discriminative fragment selection, the Average Precision score is used to evaluate the ranking of the max-pooled values per fragment proposal.
Table 3.: Event detection results on TRECVID MED 2014 for global averaging, VideoStory, and our bag-of-fragments in average precision. We also report the combination between global averaging and bag-of-fragments. Bag-of-fragments is best and highly complementary to existing encodings.

3.5 EXPERIMENTAL RESULTS

3.5.1 FRAGMENT PROPOSAL QUALITY

An overview of the recall as a function of the number of fragments is shown in Fig. 17. For our proposal algorithm and for the sliding window, the number of fragments is varied by varying the size of the initial temporal chunk. For the shot boundary detection, the number of fragments is a function of the shot threshold; the stricter the threshold, the more fragments. We have also added a shot boundary baseline that combines the fragments from multiple thresholds.

As the graph of Fig. 17 shows, our fragment proposal algorithm compares favorably in terms of recall to the two shot detection baselines. Our algorithm yields a peak recall of 0.89, while the shot boundary detection yields a peak recall of 0.52. The limited recall scores of the shot boundary detection are caused by its non-hierarchical nature. By splitting videos solely into non-overlapping fragments, important information is missed. This is further confirmed by the combined shot boundary detection baseline, which yields higher recall scores. The peak recall is however not only 8% lower (at 0.81), but also requires roughly three times as many fragments as our algorithm.

The peak recall of the sliding window baseline is equal to our algorithm, but sliding window requires far more fragments (3.2x). This result highlights the effectiveness of our method. Rather than going through all possible fragment combinations, we only examine...
3.5 EXPERIMENTAL RESULTS

![Figure 18](image)

Figure 18: The influence of the number of discriminative fragments for event detection.

the most promising combinations across the hierarchy. This results in less fragment proposals, without sacrificing recall.

### 3.5.2 EVENT DETECTION

For the event detection, we first evaluate the primary bag-of-fragments parameter, namely the number of selected discriminative fragments. Fig. 18 shows the effect of the number of fragments on the mean Average Precision (mAP) score. Using only two discriminative fragments yields a mAP of 6.0%. The performance increases monotonously as the number of discriminative fragments increases, indicating that a rich set of fragments to represent an event is beneficial. At roughly 2000 discriminative fragments, the performance starts saturating, with an mAP of 27.6% and we will use this setting for the rest of the experiments. Furthermore, we have evaluated the performance using all fragment proposals, which resulted in a mAP of 26.6%. This result indicates that only using discriminative fragments not only results in a more compact encoding, but also leads to improved results.

For the comparative evaluation we show an overview of the results for the four different methods for the 20 events Table 3. Compared to the global averaging baseline, our algorithm yields improved Average Precision scores for 15 of the 20 events, with an absolute increase of 4.4% in mean Average Precision, from 23.2% to 27.6%.

Noteworthy is the difference in performance across different events. Our discriminative fragments improves upon the baseline with 20.2%, 17.2%, and 10.2% (absolute difference) for the events Fixing a musical instrument, Dog show, and Felling a tree. The baseline method performs 15.9% and 5.2% better for the events Town hall meeting and Horse riding competition. This indicates a complementary nature between the two models. Indeed, a fusion between the two improves the performance significantly with a mean Average Precision of 37.3%, an notable absolute improvement of 14.1% over the baseline model. Note that from the fragment point-of-view, the fusion with the global
averaging baseline comes computationally for free, as the global average is always the last merge in our hierarchical clustering.

Table 3 also shows the result of VideoStory on the same dataset [53]. Although VideoStory similarly improves upon the global baseline, it does not match the bag-of-fragments result, indicating the effectiveness of the bag-of-fragments.

To further highlight the effectiveness of fragment-based event detection, we show a fragment ranking for three events in Fig. 14. The Figure shows that the most discriminative fragments of an event are exemplary snapshots of the event. Examples of this include a person on a BMX for Attempting a bike trick, a bicycle tire for Non-motorized vehicle repair, and a person on a horse for Horse riding competition. Also, the Figure shows that more ambiguous fragments are deemed less discriminative for the event.

3.5.3 EVENT RECOUNTING

For event recounting, we first highlight the effect of co-occurrence for a number of concepts in the semantic representation. In Fig. 19, the maximum co-occurrence values are shown for 25 concepts with respect to all 20 TRECVID MED 2014 events. For a number of peaks in the plot, the concept-event relation is as expected. Examples include the relationship between rodeo/jockey and Horse riding competition, between kitchen and Cleaning an appliance, and between saw and Felling a tree. These discovered concept-event relationships indicate that visual co-occurrence from social tagged image data may serve as a fruitful proxy for automatic concept selection.

However, Fig. 19 also indicates a limit of our use of co-occurrence. The concept elevator fires on the event Dog show, while they are seemingly uncorrelated. However, the textual summary of the event contains the concept lift. In the context of Dog show, lift is a verb (to lift; to move upward), but the co-occurrence statistics use the concept as a noun (lift; elevator). As the co-occurrence statistic is oblivious to the ambiguity of concepts, seemingly uncorrelated concepts might yield a high co-occurrence value.

Second, we perform a qualitative evaluation on the effect of concept filtering for event recounting. In Fig. 20, the recounting results of our discriminative fragments are shown for three videos from different events, both with and without concept selection. For each video, the three most informative video fragments are selected and for each selected fragment, the two most informative concepts are shown. The discriminative fragments are able to select the fragments of the test video that are correlated to the event. Without the concept filtering, the recounted concepts are at times incorrect or over-specific. This is exemplified in Fig. 20a and Fig. 20b, with noisy concepts such as millipede and abacus for Tuning an instrument and guillotine for Renovating a home. These results indicate the negative influence of the noise, as these concepts do not help to convince a user that the corresponding event is in the video.

However, if our concept selection is added to the recounting, the resulting concepts become both more generic and more relevant. This is for example visible in Fig. 20c. Without concept filtering, incorrect concepts such as dulcimer and wildcat are recounted. Upon adding the concept selection, concepts that are more relevant to the event Beekeeping are shown, such as honeycomb.
3.6 CONCLUSIONS

We propose encoding of videos using fragments. We show how to generate a concise set of fragment proposals for a single video. From the set of fragment proposals of the training videos for an event, we select the most discriminative ones. By matching and pooling these discriminative fragments over the fragment proposals of a video, we arrive at our bag-of-fragments encoding. Experimental evaluation shows the effectiveness of the encoding for event detection, as well as its complementary nature to a global aggregation of semantic concepts. Furthermore, we propose an automatic algorithm to filter relevant concepts in the semantic representation with respect to an event by leveraging co-occurrence statistics from a social tag image dataset. Qualitative evaluation highlights the capability of our bag-of-fragments in combination with concept filtering for event recounting.

*Figure 19.* Plot of the maximum co-occurrence value for 25 concepts with respect to the 20 TRECVID MED 2014 events.
Figure 20.: Qualitative event recounting results with and without the concept filtering for three different videos. For each video, the top three fragments are selected and the top two concepts for each selected fragment are shown, along with their fragment-based score. For each recounted fragment, the matching discriminative fragment from the training set is shown.
4.1 INTRODUCTION

The goal of this work is to detect events such as Renovating a home, Birthday party, and Attempting a bike trick in web videos. The leading approaches [67, 104, 114, 199] attack this challenging problem by learning video representations through a deep convolutional neural network [78, 164]. The deep network is pre-trained on a collection of 1,000 ImageNet classes [26, 136], used to extract features for video frames, and then followed by a pooling operation over the frames to arrive at a video representation. We also learn representations for event detection with a deep convolutional neural network, but rather than relying on the default 1,000 class subset, we investigate how to leverage the complete ImageNet hierarchy for pre-training the representation.

The complete ImageNet dataset consists of over 14 million images and 21,814 classes which are connected in a hierarchy as a subset of WordNet [109]. State-of-the-art event detectors are pre-trained on a 1,000 class (1.2 million images) subset of ImageNet, as prescribed by the Large Scale Visual Recognition Challenge [136]. Hence, more than 90% of the images in the hierarchy remain untouched during pre-training. We present an empirical investigation of the effect of using the full ImageNet dataset for event detection in web videos.

We identify two problems when trying to pre-train a deep network on the complete ImageNet hierarchy. First, there is an imbalance in the number of examples for each class, as is shown in Figure 21a. For example, the class Yorkshire Terrier contains 3,072 images, whereas 296 other classes contain just a single image. Second, some classes seem over-specific for event detection in web videos. Consider for example the ImageNet categories Siderocyte and Gametophyte in Figure 21b. As a result, it seems suboptimal to directly pre-train a deep network on all 21,814 classes.

In this work, we introduce pre-training protocols that reorganize the full ImageNet hierarchy for more effective pre-training. The reorganization tackles the image imbalance and over-specific class problems. We propose two contrasting approaches that utilize the graph structure of ImageNet to combine and merge classes into balanced and reorganized hierarchies. We empirically evaluate our event detection using reorganized pre-training on the 2013 and 2015 NIST TRECVID Multimedia Event Detection datasets, for both datasets leading to state-of-the-art results. The Caffe models and detailed video feature extraction instructions are available at http://tinyurl.com/imagenetshuffle.

Published in *International Conference on Multimedia Retrieval, 2016* [101].
(a) Distribution of the number of images for the 21,814 ImageNet classes. Note the class imbalance.

(b) An image of *Siderocyte* (left) and *Gametophyte* (right), two classes which seem over-specific for event detection.

*Figure 21.* Two problems when using the full ImageNet hierarchy for network pre-training: (a) image imbalance and (b) over-specific classes. In this work, we aim to reorganize the hierarchy into a balanced set of classes for more effective pre-training of video representations for event detection.

### 4.2 RELATED WORK

#### 4.2.1 EVENT DETECTION WITH PRE-TRAINED NETWORKS

The state-of-the-art for event detection in videos focuses on representations learned with the aid of deep convolutional neural networks [46, 67, 104, 114, 199]. The pipeline of these approaches consists roughly of three components. (1) A deep network is pre-trained on a large-scale image collection. Different deep networks have been employed for event detection, such as AlexNet [78] in [67,104,114] and VGGnet [149] in [199]. (2) Sampled video frames are fed to the network and features at fully-connected and/or soft-max layers are used as frame representations. (3) The frame representations are pooled into a fixed-sized video representation. A simple and effective pooling method is average
pooling, where the frame representations are averaged over the video [71, 104]. Recently, several works have shown that clustering deep frame representations into a codebook, followed by a VLAD [64] in [199] or Fisher Vector encoding [141] in [114], leads to strong video representations. In this work, we aim for a similar pipeline of pre-training, frame representation, and video pooling. However, rather than relying on the standard pre-training protocol using 1,000 ImageNet classes, we leverage the complete ImageNet hierarchy for more effective pre-training.

Web videos provide a wide range of information about events, such as visual, motion, audio, and optical character information [123]. Naturally, multiple works have investigated the fusion of information from different modalities [83, 113, 115]. In this work, we also investigate the effect of fusing our deep representations with Motion Boundary Histogram (motion) features [183] and MFCC (audio) features [113], both of which are encoded into a video representation using Fisher Vectors [141]. This fusion allows us to compare the effectiveness of our deep representations to heterogeneous representations and to investigate how well our deep representations fare when combined with other sources of information.

4.2.2 (Re)organizing Hierarchies for Events

Various works have investigated the use of semantic hierarchies and ontologies for event detection. The work of Ye et al. [203] focuses on hierarchical relations between events, to find a large collection of videos and event-specific concept classes. Their proposed EventNet has shown to yield an effective event detection [203]. In our work, we focus on different hierarchical relations, namely between concept classes instead of events, to discover a general set of concepts for deep network pre-training. Other recent work on event detection has investigated relations among concept classes to rerank concept scores in the video representation [67]. We similarly focus on hierarchical relations among concept classes, but for the purpose of merging classes into a reorganized hierarchy for pre-training. For the hierarchy of ImageNet specifically, the work of Vreeswijk et al. [180] has shown that images from different layers of the hierarchy are visually different and that general concepts benefit from including linked concepts deeper in the hierarchy. We build upon these observations in our operations to reorganize the ImageNet hierarchy.

An alternative approach is to adjust concept hierarchies after feature extraction. For example, the selection of event-specific concepts based on the similarity to a textual event description has shown to yield effective event detection results without positive examples [67]. Mazloom et al. [97] show that concept selection is also beneficial for few-example event detection. Habibian et al. [53] in turn, jointly learn a classifier for event detection and combine correlated concepts. Rather than changing the representations a posteriori using text or video examples, we focus in this paper on reorganizing the hierarchical structure of visual ontologies before event training.
4.3 REORGANIZED PRE-TRAINING

The classes in the ImageNet dataset are a subset of the WordNet collection [109] and the classes are therefore connected in a hierarchy. The connectivity between classes provides information about their semantic relationship. We utilize the hierarchical relationship of WordNet for combining classes to generate reorganized ImageNet hierarchies for pre-training. We focus on two opposing approaches for reorganization, namely a bottom-up and top-down approach.

4.3.1 BOTTOM-UP REORGANIZATION

For the bottom-up reorganization, we start from the original ImageNet hierarchy and introduce four reorganization operations. An overview of where in the hierarchy the four operations are performed is shown in Figure 22 and visual examples for each operation are shown in Figure 23. We outline each operation separately.

Roll. The roll operation is performed on single-link sub-trees of classes. In other words, the roll operation merges classes with a single child-parent connection, as shown in Figure 22 on the left. The motivation behind this operation is two-fold: i) there is little semantic difference between a child and a parent if the parent has no other children. Treating the child and parent as separate classes during pre-training will dominate the backpropagation gradients to keep these classes separated. ii) A single child of a parent is more likely to be over-specific for event detection. Single child-parent connections typically occur deep in the hierarchy, where details between classes become more fine-grained. In our evaluation, we indeed observe that the single child-parent connections occur predominantly in the deeper layers of the ImageNet hierarchy. Three chains of single child-parent connections are shown in Figure 23a. For example, the class Mamba is a type of snake and has a single child, namely Black mamba. In turn, the Black mamba has a single child: Green mamba (the green phase of the black mamba). In this example, we move all the images from the Black mamba and Green mamba classes to the Mamba class.
4.3 Reorganized Pre-Training

(a) Roll.

(b) Bind.

(c) Promote.

(d) Subsample.

Figure 23.: Visual examples for each of the four operations for the bottom-up reorganization.

Bind. The bind operation is performed on sub-trees where the individual classes are sparse in the number of images. Let $S$ denote a sub-tree and let $c^i$ denote the number of images in class $c$. Then the bind operation is performed on sub-tree $S$ if $\sum_{c \in S} c^i < t_b$, where $t_b$ denotes the threshold on the number of images. The notion behind the bind operation is to deal with small and semantically coherent classes consisting of a parent and multiple children. The children individually do not contain enough images to treat them as separate classes. However, the combined set of parent and children forms a semantically consistent set with a desirable number of images. Three merged sub-trees that are combined with the bind operation are shown in Figure 23b. For example, the Hammerhead shark has three children with a small number of images, namely Smooth hammerhead, Smalleye hammerhead, and Shovelhead. Therefore, we opt to combine all these shark images into a single class.

Promote. The promote operation is a unary operation. It is performed after the roll and bind have been performed. The promote operation simply promotes a class to its parent if its number of images is below a threshold $t_p$. This operation directly targets the
imbalance problem, by adding images of classes with few examples to parent classes with more images. Figure 23c shows three cases of the promote operation. For example, the class *Triclinium* (a dining table with couches at three sides) only contains 5 images. Therefore, the images are added to the *Dining table* class, such that the *Triclinium* images are still being used for pre-training without creating an imbalance in the hierarchy.

**Subsample.** The subsample operation is also a unary operation and deals with the reverse problem of the other three operations. The subsample operation subsamples images from classes for which the number of images is above a threshold \( t_s \). The operation selects a subset of images from classes with a lot of examples. The reason for this operation is again for balancing purposes. If all images of over-populated classes are used in the optimization of the deep network, the network will overfit to these classes, resulting in suboptimal frame representations for event detection. Four examples of the subsample operation are shown in Figure 23d, such as *Keyboard*, *Coffee mug*, and *Herb*.

We employ the defined operations in the described order. First, all single child-parent connections are rolled up. Second, all sub-trees in the hierarchy are binded based on threshold \( t_b \) for their combined number of images. Third, all remaining classes with less than \( t_p \) images are promoted to their parent. Fourth, during network pre-training, examples for all classes with more than \( t_s \) images are randomly sub-sampled before the stochastic gradient descent optimization.

### 4.3.2 Top-Down Reorganization

An alternative complementary reorganization strategy is not to start from the deepest classes in the hierarchy, but from the head node. Here, we investigate a breath-first search approach. Let \( t_t \) denotes the threshold stating the minimum amount of images required for class in order to be used in the top-down reorganization. Then, starting from layer 0 in the hierarchy, i.e. the head node, we iteratively move down in the hierarchy and keep adding classes with at least \( t_t \) images until we reach a desired amount of classes.

The breath-first search approach is outlined as follows. Let \( l \) denote the previous layer of the hierarchy. We list all ImageNet classes in layer \( l + 1 \) based on connections from classes in layer \( l \) and order the classes in \( l + 1 \) by their number of images. The sorting ensures that we select the classes with the highest number of images first, in case we reach the desired amount of classes before the end of the list. Then, we move through the ordered list and select all classes with at least \( t_t \) images as long as the desired amount of selected classes is not reached. Afterwards, we move to the next layer and repeat the ordering and selection procedure.

By using a top-down approach, we ensure that only the most general classes are maintained for pre-training, while simultaneously keeping a balance in the image distribution through the threshold \( t_t \).
Figure 24.: Mean Average Precision scores for the three bottom-up variants on TRECVID Multimedia Event Detection 2013. We observe that the more classes are maintained in the bottom-up reorganization, the better the performance using the soft-max (i.e. semantic) layer. The reserve happens for the fully-connected layer.

4.4 EXPERIMENTAL SETUP

4.4.1 DATASET

TRECVID Multimedia Event Detection 2013. The TRECVID Multimedia Event Detection 2013 dataset consists of roughly 27,000 test videos [123]. The dataset contains annotations for 20 everyday events, including Birthday Party, Making a sandwich, Attempting a bike trick, and Dog show. The dataset has two different tasks, one where 10 positive videos are given for each event (10 Ex.), and one where 100 positive videos are given for each event (100 Ex.). For an event, a classifier is trained on the 10 or 100 positive videos and a background set of roughly 5,000 negative videos. The classifier is in turn used to rank the 27,000 test videos and its performance is evaluated using the (mean) Average Precision score on the ranked test videos.

4.4.2 IMPLEMENTATION DETAILS

Deep convolutional networks. We focus our evaluation on the recent GoogLeNet of Szegedy et al. [164]. The GoogLeNet is a deep convolutional neural network consisting of 22 layers. We also compare against the AlexNet of Krizhevsky et al. [78]. The AlexNet consists of 5 convolutional layers and 3 fully-connected layers. To pre-train the deep networks, we utilize the open-source Caffe library [66] and the provided layer definitions and hyper-parameters for both networks.

Feature extraction. After pre-training, we extract features both at the fully connected layer and the soft-max layer. In AlexNet, we use the features from the second fully-connected layer, with a 4,096-dimensional frame representation. In GoogLeNet, we use the features at the pool5 layer, with a 1,024-dimensional frame representation. The dimensionality at the soft-max layer, which provides a probability score of each concept, for both networks is equal to the number of classes in the corresponding hierarchy.
Pooling and Event Classification. For event detection, we average the representations of the frames over each video unless stated otherwise, followed by $\ell_1$-normalization. We train an SVM classifier for each event separately with a $\chi^2$ kernel. We set the $C$ parameter to 100 in all our experiments.

4.5 EXPERIMENTS

We consider four experiments. First, we evaluate the effect of different settings of the operations in our bottom-up reorganized pre-training. Second, we compare standard pre-training versus both the bottom-up and top-down reorganized pre-training. Third, we perform various fusions between deep representations and representation from other modalities. Fourth, we compare our results to the state-of-the-art on multimedia event detection.

4.5.1 BOTTOM-UP OPERATION PARAMETERS

Experiment 1. For the first experiment, we investigate the parameters for the bind and promote operations in our bottom-up reorganization, which have a significant influence on the amount of remaining classes. In total, we have trained three separate GoogLeNets [164] based on different parameters for the bind and promote operations:

- **Bottom-up [4k]:** Deep network pre-trained on 4,437 classes with $t_b = 7,000$ and $t_p = 1,250$.
- **Bottom-up [8k]:** Deep network pre-trained on 8,201 classes with $t_b = 7,000$ and $t_p = 500$.
- **Bottom-up [13k]:** Deep network pre-trained on 12,988 classes with $t_b = 3,000$ and $t_p = 200$.

For all the variants, we set the subsample threshold to $t_s = 2,000$. An overview of mean Average Precision scores using the fully-connected and soft-max layers on TRECVID Multimedia Event Detection 2013 is shown in Figure 24. We report the mean Average Precision scores both for the task with 10 positive videos and with 100 positive videos per event.

Results. From Figure 24, we observe that the best scores using the fully-connected layer are achieved with the bottom-up [4k] variant. This result shows that the fully-connected layer translates best to events when merging more classes into a generic hierarchy. Interestingly, we observe the reverse pattern for the soft-max layer; the more classes are maintained the better the event detection performance. This result follows the work of Habibian et al. [54], which states that using more semantic classifiers is preferred over using better semantic classifiers. Here, we show that this observation translates to deep networks for event detection.

From this experiment, we conclude that the choice of the bottom-up reorganized variant depends on the desired deep network representation. The highest overall results are achieved by the features from the non-semantic fully-connected layer of the variant.
4.5 Experiments

(a) TRECVID MED 2013 with 100 positives. (b) TRECVID MED 2013 with 10 positives.

Figure 25.: Mean Average Precision scores for our bottom-up and top-down reorganized pre-training (blue), compared to standard pre-training (green) on TRECVID MED 2013. Our approaches both clearly outperform standard pre-training, while being competitive and potentially complementary to each other.

from 4,437 classes (0.446 mean Average Precision using 100 positives per event, 0.296 using 10 positives). However, the variant from 12,988 classes performs best using the semantic features from the soft-max layer (0.441 using 100 positives, 0.286 using 10 positives).

4.5.2 Standard versus Reorganized Pre-Training

Experiment 2. For the second experiment, we compare our bottom-up and top-down reorganized pre-training against the conventional pre-training setup using the ImageNet 1,000 class subset [136]. For all networks, we report the Average Precision scores using both the fully-connected layer and the soft-max layer. For the bottom-up approach, we use the deep network pre-trained on 4,437 classes. For the top-down approach, we select the top 4,000 classes, with $t_i = 1,200$ for the threshold on the number of images required for each class. We compare our two approaches to two standard pre-trained deep networks:

- **AlexNet [std]:** AlexNet pre-trained on 1,000 ImageNet classes [78].
- **GoogLeNet [std]:** GoogLeNet pre-trained on 1,000 ImageNet classes [164].

Results. An overview of the comparison between standard and reorganized pre-training is shown in Figure 25. We observe that the top-down and bottom-up reorganization approaches achieve comparable performance. While bottom-up performs slightly better using the fully-connected layer, top-down performs slightly better using the soft-max layer. We also note our reorganized pre-training approaches on GoogLeNet significantly outperform the standard pre-trained GoogLeNet. This holds especially for the soft-max layer, where the difference between standard pre-training and our top-down pre-training is 8.6% and 9.2% in absolute mean Average Precision for respectively the 100 and 10 positive video tasks. Lastly, we note that the difference in performance to the
Table 4: Mean Average Precision scores for fusions of different layers, networks, and encodings within deep representations, which all yield complementary results.

<table>
<thead>
<tr>
<th></th>
<th>TRECVID MED 2013</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100 Ex.</td>
</tr>
<tr>
<td><strong>Averaging</strong></td>
<td></td>
</tr>
<tr>
<td>(1) Bottom-up (fc)</td>
<td>0.446</td>
</tr>
<tr>
<td>(2) Top-down (fc)</td>
<td>0.438</td>
</tr>
<tr>
<td>(3) Bottom-up (fc + soft-max)</td>
<td>0.452</td>
</tr>
<tr>
<td>(4) Top-down (fc + soft-max)</td>
<td>0.454</td>
</tr>
<tr>
<td>(3) + (4)</td>
<td>0.475</td>
</tr>
<tr>
<td><strong>VLAD</strong></td>
<td></td>
</tr>
<tr>
<td>Bottom-up (fc + soft-max)</td>
<td>0.465</td>
</tr>
</tbody>
</table>

pre-trained AlexNet is even bigger. This result shows that GoogLeNet provides overall better visual representations, leading to improved event detection.

From this experiment, we conclude that our two approaches to reorganized ImageNet pre-training yield strong event detection results and significantly improve over standard pre-trained deep networks.

### 4.5.3 Fusing Representations and Modalities

**Experiment 3.** For the third experiment, we investigate the effect of feature fusion. Here, fusion is performed in a late fashion, by averaging the classifier scores of different classifiers. We investigate feature fusion in two aspects: i) we investigate the effect of fusing different layers and video encodings from different pre-trained deep networks for event detection, ii) we investigate the effect of fusing our deep visual representations with two other representations:

- **Audio modality:** MFCC features with first and second order derivatives, 30 dimensions for each of the three features, aggregated into a 46,080-dimensional video representation using Fisher Vectors with 256 clusters.

- **Motion modality:** MBHx, MBHy, and HOG features computed along dense trajectories [183], reduced to 128 dimensions using PCA, aggregated into a 65,536-dimensional video representation using Fisher Vectors with 256 clusters.

**Results for Fusing Networks.** In Table 4, we show an overview of fusion results using deep networks. Comparing index (1) to (3) and comparing index (2) to (4), we see that for both the bottom-up and top-down approach, it is beneficial to fuse the scores from the fully-connected and soft-max layers. This result is surprising, given that the layers come from the same network and it indicates that the layers contain different information useful for event detection. This result is furthermore interesting from a computational perspective, as the features from both layers can be extracted from a single pass through the same network. Hence, the improvement is obtained for free.
The fusion of (3) and (4), i.e. the fusion of the bottom-up and top-down reorganizations, also yields complementary results, with a mean Average Precision of 0.475 and 0.324 for respectively the 100 and 10 positive video tasks of the TRECVID Multimedia Event Detection 2013 dataset. This result clearly shows that pre-training deep networks on different hierarchies results in different and complementary representations. Figure 26 shows that, although the mean Average Precision of the individual approaches is similar, the scores per event vary notably (on average 2.7% per event), resulting in improved performance upon fusion.

Multiple recent works have investigated complex and high-dimensional video representations from deep frame representations beyond frame averaging [114, 199]. Here, we similarly investigate such representations using the frame features with reorganized pre-training. We have employed both VLAD and Fisher Vector encoding and report the results for VLAD in Table 4, as that yielded the highest scores. For the VLAD encoding, we create a codebook from 10 clusters per event, resulting in a 10,240-dimensional feature vector using the fully-connected layer and a 44,370-dimensional feature vector using the soft-max layer. The results using the bottom-up reorganization show that a VLAD encoding improves over averaging, especially for the 10 positive videos per event task (0.339 mAP versus 0.305 for averaging).

We conclude from this fusion experiment that combining information from different pre-trained deep networks and even different layers from the same deep network improves the Average Precision scores. Furthermore, performing a VLAD encoding instead of averaging frames results in a boost for individual networks, especially for the 10 positive videos task.

Results for Fusing Modalities. In Table 5, we show the results of the deep networks with the audio and motion modalities. The Table clearly states that individually, the event detection scores using our deep networks improve over the motion and audio scores. Upon combining the modalities, we observe a jump in performance. This result shows the complementary natures of the different modalities: individually the motion and audio features are clearly outperformed, but they contain information not captured in deep networks which result in improved fusion results. This is naturally due to the nature of deep convolutional neural networks, which focus on spatially visual information and exclude temporal and audio information.

We have furthermore attempted to fuse the VLAD encoding of Table 4 with the motion and audio features, but this did not result in improved performance. Since the VLAD encoding requires more computational effort and has a higher storage requirement, we have opted to focus on averaging.

4.5.4 COMPARISON TO THE STATE-OF-THE-ART

Experiment 4. For the fourth experiment, we compare our results to the current state-of-the-art on multimedia event detection. We perform a comparison on both the TRECVID MED 2013 test set and the TRECVID MED 2015 benchmark.

Results on the TRECVID MED 2013 Test set. The comparison to the state-of-the-art on the TRECVID Multimedia Event Detection 2013 dataset is shown in Table 6. As the Table shows, we outperform the current state-of-the-art on both the 100 and 10
positive videos per event task using deep networks only. Upon a fusion with motion and audio features, we improve further over related work.

**Results on the TRECVID MED 2015 Benchmark.** We furthermore compare our results achieved on the latest TRECVID 2015 benchmark for Multimedia Event Detection. This benchmark is similar in nature to the 2013 dataset in training and evaluation. However, the 2015 dataset contains 20 new events. Furthermore, the benchmark comparison is performed in a large-scale setup, with a test set of about 200,000 videos. In Figure 27, we show the inferred mean Average Precision scores for our entries and the entries of the other participants. We report results both for the pre-specified (where the events and video labels are given well before the benchmark deadline) and ad-hoc (where the events and video labels are given shortly before the benchmark deadline) tasks. The Figure paints a similar picture to the results on the 2013 dataset; we outperform the current state-of-the-art by fusing deep representations with motion and audio modalities, while our deep representations only are already among the top contenders.
Table 5.: Mean Average Precision scores for fusing our deep representations with other modalities. We outperform motion and audio features, while the fusion leads to further improvement.

<table>
<thead>
<tr>
<th>Method</th>
<th>100 Ex.</th>
<th>10 Ex.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Individual</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1) Audio (MFCC)</td>
<td>0.114</td>
<td>0.053</td>
</tr>
<tr>
<td>(2) Motion (MBH)</td>
<td>0.341</td>
<td>0.192</td>
</tr>
<tr>
<td>(3) Visual (ours, avg)</td>
<td>0.475</td>
<td>0.324</td>
</tr>
<tr>
<td>Fusion</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2) + (3)</td>
<td>0.504</td>
<td>0.345</td>
</tr>
<tr>
<td>(1) + (2) + (3)</td>
<td>0.526</td>
<td>0.348</td>
</tr>
</tbody>
</table>

Table 6.: Comparison to other works on TRECVID MED 2013 test set for both our best deep network results and our fusion results. We yield better results for both the 100 and 10 positive videos per event task.

<table>
<thead>
<tr>
<th>Method</th>
<th>100 Ex</th>
<th>10 Ex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Habibian et al. [53]</td>
<td>-</td>
<td>0.196</td>
</tr>
<tr>
<td>Sun et al. (visual) [161]</td>
<td>0.350</td>
<td>-</td>
</tr>
<tr>
<td>Nagel et al. [114]</td>
<td>0.386</td>
<td>0.218</td>
</tr>
<tr>
<td>Sun et al. (fusion) [161]</td>
<td>0.425</td>
<td>-</td>
</tr>
<tr>
<td>Xu et al. [199]</td>
<td>0.446</td>
<td>0.298</td>
</tr>
<tr>
<td>Chang et al. [15]</td>
<td>-</td>
<td>0.310</td>
</tr>
<tr>
<td>Ours, deep network</td>
<td>0.475</td>
<td>0.324</td>
</tr>
<tr>
<td>Ours, multimodal fusion</td>
<td>0.526</td>
<td>0.348</td>
</tr>
</tbody>
</table>

4.6 CONCLUSIONS

In this work, we leverage the complete ImageNet dataset for pre-training deep convolutional neural networks for video event detection, rather than the prescribed 1,000 ImageNet subset. We propose two contrasting and complementary approaches to reorganize the ImageNet hierarchy. The bottom-up approach aims to merge classes from the deepest parts of hierarchy upwards, while the top-down approach aims to select rich generic classes starting from the top of the hierarchy. The new hierarchies are in turn used as input to pre-train deep networks and are employed for frame representation in video event detection. Experimental evaluation performed on the challenging TRECVID MED 2013 dataset shows that deep networks trained on our hierarchies \( i \) outperform standard pre-trained networks, \( ii \) are complementary, \( iii \) maintain the benefits of fusion with other modalities, and \( iv \) reach state-of-the-art result. The pre-trained models are
Figure 27.: Comparison between our results (blue) and the results of all other participants in the TRECVID Multimedia Event Detection benchmark 2015. Our deep networks and their fusion with motion and audio information are the top contenders for all tasks.

available online at http://tinyurl.com/imagenetshuffle and can be used directly to extract state-of-the-art video representations using the Caffe library.
5.1 INTRODUCTION

This paper is about spatio-temporal localization of actions like *Driving a car*, *Kissing*, and *Hugging* in videos. Starting from a sliding window legacy [166], the common approach these days is to generate tube-like proposals at test time, encode each of them with a feature embedding and select the most relevant one, e.g. [61, 157, 175, 205]. All these works, be it sliding windows or tube proposals, assume that a carefully annotated training set with boxes per frame is available a priori. In this paper, we challenge this assumption. We propose a simple algorithm that leverages proposals at training time, with a minimum amount of supervision, to speedup action location annotation.

We draw inspiration from related work on weakly-supervised object detection, e.g. [22, 74, 137]. The goal is to detect an object and its bounding box at test time given only the object class label at train time and no additional supervision. The common tactic in the literature is to model this as a Multiple Instance Learning (MIL) problem [3, 22, 117] where positive images contain at least one positive object proposal and negative images contain only negative proposals. During each iteration of MIL, a detector is trained and applied on the train set to re-identify the object proposal most likely to enclose the object of interest. Upon convergence, the final detector is applied on the test set. Methods typically vary in their choice of initial proposals and the multiple instance learning optimization. In the domain of action localization a similar MIL tactic easily extends to action proposals as well but results in poor accuracy as our experiments show. Similar to weakly-supervised object detection, we rely on (action) proposals and MIL, but we include a minimum amount of supervision to retain action localization accuracy competitive with full supervision.

Obvious candidates for the supervision are action class labels and bounding boxes, but other forms of supervision, such as tags and line strokes, are also feasible [195]. In [6], Bearman et al. show that human-provided points on the image are valuable annotations for semantic segmentation of objects. By inclusion of an objectness prior in their loss function they report a better efficiency/effectiveness trade off compared to image-level annotations and free-from squiggles. We follow their example in the video domain and leverage point-supervision to aid MIL in finding the best action proposals at training time.

Published in *European Conference on Computer Vision, 2016* [106].
SPOT ON: ACTION LOCALIZATION FROM POINTLY-SUPERVISED PROPOSALS

Figure 28.: Overview of our approach for a Swinging and Standing up action. First, the video is annotated cheaply using point-supervision. Then, action proposals are extracted and scored using our overlap measure. Finally, our proposal mining aims to discover the single one proposal that best represents the action, given the provided points.

We make three contributions in this work. First, we propose to train action localization classifiers using spatio-temporal proposals as positive examples rather than ground truth tubes. While common in object detection, such an approach is as of yet unconventional in action localization. In fact, we show that using proposals instead of ground truth annotations does not lead to a decrease in action localization accuracy. Second, we introduce an MIL algorithm that is able to mine proposals with a good spatio-temporal fit to actions of interest by including point supervision. It extends the traditional MIL objective with an overlap measure that takes into account the affinity between proposals and points. Finally, with the aid of our proposal mining algorithm, we are able to supplement the complete Hollywood2 dataset by Marszałek et al. [96] with action location annotations, resulting in Hollywood2Tubes. We summarize our approach in Figure 28. Experiments on Hollywood2Tubes, as well as the more traditional UCF Sports and UCF 101 collections support our claims. Before detailing our pointly-supervised approach we present related work.

5.2 RELATED WORK

Action localization is a difficult problem and annotations are avidly used. Single image bounding box annotations allow training a part-based detector [82, 166] or a per-frame detector where results are aggregated over time [49, 188]. However, since such detectors first have to be trained themselves, they cannot be used when no bounding box annotations are available. Independent training data can be brought in to automatically detect individual persons for action localization [94, 185, 205]. A person detector, however, will fail to localize contextual actions such as Driving or interactions such as Shaking hands or Kissing. Recent work using unsupervised action proposals based on supervoxels [61,
or on trajectory clustering \([17, 95, 175]\), have shown good results for action localization. In this paper we rely on action proposals to aid annotation. Proposals give excellent recall without supervision and are thus well-suited for an unlabeled train set.

Large annotated datasets are slowly coming available in action localization. Open annotations benefit the community, paving the way for new data-driven action localization methods. UCF-Sports \([158]\), HOHA \([131]\) and MSR-II \([13]\) have up to a few hundred actions, while UCF101 \([159]\), Penn-Action \([213]\), and J-HMBD \([65]\) have 1–3 thousand action clips and 3 to 24 action classes. The problem of scaling up to larger sets is not due to sheer dataset size: there are millions of action videos with hundreds of action classes available \([51, 71, 79, 159]\). The problem lies with the spatio-temporal annotation effort. In this paper we show how to ease this annotation effort, exemplified by releasing spatio-temporal annotations for all Hollywood2 \([96]\) videos.

Several software tools are developed to lighten the annotation burden. The gain can come from a well-designed user interface to annotate videos with bounding boxes \([107, 178]\) or even polygons \([209]\). We move away from such complex annotations and only require a point. Such point annotations can readily be included in existing annotation tools which would further reduce effort. Other algorithms can reduce annotation effort by intelligently selecting which example to label \([142]\). Active learning \([179]\) or trained detectors \([8]\) can assist the human annotator. The disadvantage of such methods is the bias towards the used recognition method. We do not bias any algorithm to decide where and what to annotate: by only setting points we can quickly annotate all videos.

Weakly supervised methods predict more information than was annotated. Examples from static images include predicting a bounding box while having only class labels \([9, 22, 122]\) or even no labels at all \([19]\). In the video domain, the temporal dimension offers more annotation variation. Semi-supervised learning for video object detection is done with a few bounding boxes \([1, 110]\), a few global frame labels \([184]\), only video class labels \([151]\), or no labels at all \([80]\). For action localization, only the video label is used by \([112, 152]\), whereas \([62]\) use no labels. As our experiments show, using no label or just class labels performs well below fully supervised results. Thus, we propose a middle ground: pointing at the action. Compared to annotating full bounding boxes this greatly reduces annotation time while retaining accuracy.

### 5.3 Action Localization Using Cheap Annotations

We start from the hypothesis that an action localization proposal may substitute the ground truth on a training set without a significant loss of classification accuracy. Proposal algorithms yield hundreds to thousands of proposals per video with the hope that at least one proposal matches the action well \([17, 61, 95, 118, 157, 175]\). The problem thus becomes how to mine the best proposal out of a large set of candidate proposals with minimal supervision effort.
5.3.1 ACTION CLASS LABELS AND POINTLY-SUPERVISION

A minimum of supervision effort is an action class label for the whole video. For such global video labels, a traditional approach to mining the best proposal is Multiple Instance Learning (MIL). In the context of action localization, each video is interpreted as a bag and the proposals in each video are interpreted as its instances. The goal of MIL is to train a classifier that can be used for proposal mining by using only the global label.

Next to the global action class label we leverage cheap annotations within each video: for a subset of frames we simply point at the action. We refer to such a set of point annotations as pointly-supervision. The supervision allows us to easily exclude those proposals that have no overlap with any annotated point. Nevertheless, there are still many proposals that intersect with at least one point. Thus, points do not uniquely identify a single proposal. In the following we will introduce an overlap measure to associate proposals with points. To perform the proposal mining, we will extend MIL’s objective to include this measure.

5.3.2 OVERLAP BETWEEN POINTS AND PROPOSALS

To explain how we obtain our overlap measure, let us first introduce the following notation. For a video \( V \) of \( N \) frames, an action localization proposal \( A = \{BB_i\}_{i=1}^m \) consists of connected bounding boxes through video frames \((f, ..., m)\) where \( 1 \leq f \leq m \leq N \). We use \( BB_i \) to indicate the center of a bounding box \( i \). The pointly-supervision \( C = \{(x_i, y_i)\}_{i=1}^K \) is a set of \( K \leq N \) sub-sampled video frames where each frame \( i \) has a single annotated point \((x_i, y_i)\). Our overlap measure outputs a score for each proposal depending on how well the proposal matches the points.

Inspired by a mild center-bias in annotators [169], we introduce a term \( M(\cdot) \) to represent how close the center of a bounding box proposal is to an annotated point, relative to the bounding box size. Since large proposals have a higher likelihood to contain any annotated point we use a regularization term \( S(\cdot) \) on the proposal size. The center-bias term \( M(\cdot) \) normalizes the distance to the bounding box center by the distance to the furthest bounding box side. A point \((x_i, y_i) \in C \) outside a bounding box \( BB_i \in A \) scores 0 and a point on the bounding box center \( BB_i \) scores 1. The score decreases linearly with the distance to the center for the point. It is averaged over all annotated points \( K \):

\[
M(A, C) = \frac{1}{K} \sum_{i=1}^{K} \max(0, 1 - \frac{||((x_i, y_i) - BB_i)||_2}{\max_{(u,v) \in e(BB_{K_i})} ||((u, v) - BB_i)||_2}) \tag{5.1}
\]

where \( e(BB_{K_i}) \) denotes the box edges of box \( BB_{K_i} \).

We furthermore add a regularization on the size of the proposals. The idea behind the regularization is that small spatial proposals can occur anywhere. Large proposals, however, are obstructed by the edges of the video. This biases their middle-point around the center of the video, where the action often happens. The size regularization term
5.3 action localization using cheap annotations

\( S(\cdot) \) addresses this bias by penalizing proposals with large bounding boxes \( |BB_i| \in A \), compared to the size of a video frame \( |F_i| \in V \),

\[
S(A, V) = \left( \frac{\sum_{i=1}^{m} |BB_i|}{\sum_{j=1}^{N} |F_j|} \right)^2.
\] (5.2)

Using the center-bias term \( M(\cdot) \) regularized by \( S(\cdot) \), our overlap measure \( O(\cdot) \) is defined as

\[
\] (5.3)

Recall that \( A \) are the proposals, \( C \) captures the pointly-supervision and \( V \) the video. We use \( O(\cdot) \) in an iterative proposal mining algorithm over all annotated videos in search for the best proposals.

5.3.3 MINING PROPOSALS OVERLAPPING WITH POINTS

For proposal mining, we start from a set of action videos \( \{x_i, t_i, y_i, C_i\}_{i=1}^{N} \), where \( x_i \in \mathbb{R}^{A_i \times D} \) is the \( D \)-dimensional feature representation of the \( A_i \) proposals in video \( i \). Variable \( t_i = \{\{BB_j\}_{j=f_i}^{m_i}\}^{A_i} \) denotes the collection of tubes for the \( A_i \) proposals. Cheap annotations consist of the class label \( y_i \) and the points \( C_i \).

For proposal mining we insert our overlap measure \( O(\cdot) \) in a Multiple Instance Learning scheme to train a classification model that can learn the difference between good and bad proposals. Guided by \( O(\cdot) \), the classifier becomes increasingly more aware about which proposals are a good representative for an action. We start from a standard MIL-SVM \([3, 22]\) and adapt it’s objective with the the mining score \( P(\cdot) \) of each proposal, which incorporates our function \( O(\cdot) \) as:

\[
\min_{w, b, \xi} \frac{1}{2} \|w\|^2 + \lambda \sum_{i} \xi_i,
\]

\[
\text{s.t. } \forall i : y_i \cdot (w \cdot \arg \max_{z \in x_i} P(z|w, b, t_i, C_i, V_i) + b) \geq 1 - \xi_i,
\]

\[
\forall i : \xi_i \geq 0,
\] (5.4)

where \((w, b)\) denote the classifier parameters, \( \xi_i \) denotes the slack variable and \( \lambda \) denotes the regularization parameter. The proposal with the highest mining score per video is used to train the classifier.

The objective of Equation 5.4 is non-convex due to the joint minimization over the classifier parameters \((w, b)\) and the maximization over the mined proposals \( P(\cdot) \). Therefore, we perform iterative block coordinate descent by alternating between clamping one and optimizing the other. For fixed classifier parameters \((w, b)\), we mine the proposal with the highest Maximum a Posteriori estimate with the classifier as the likelihood and \( O(\cdot) \) as the prior:

\[
P(z|w, b, t_i, C_i, V_i) \propto (\langle w, z \rangle + b) \cdot O(t_i, C_i, V_i).
\] (5.5)

After a proposal mining step, we fix \( P(\cdot) \) and train the classifier parameters \((w, b)\) with stochastic gradient descent on the mined proposals. We alternate the mining and classifier...
optimizations for a fixed amount of iterations. After the iterative optimization, we train a final SVM on the best mined proposals and use that classifier for action localization.

5.4 EXPERIMENTAL SETUP

5.4.1 DATASETS

We perform our evaluation on two action localization datasets that have bounding box annotations both for training and test videos.

**UCF Sports** consists of 150 videos covering 10 action categories [135], such as Diving, Kicking, and Skateboarding. The videos are extracted from sport broadcasts and are trimmed to contain a single action. We employ the train and test data split as suggested in [82].

**UCF 101** has 101 action categories [159] where 24 categories have spatio-temporal action localization annotations. This subset has 3,204 videos, where each video contains a single action category, but might contain multiple instances of the same action. We use the first split of the train and test sets as suggested in [159] with 2,290 videos for training and 914 videos for testing.

5.4.2 IMPLEMENTATION DETAILS

**Proposals.** Our proposal mining is agnostic to the underlying proposal algorithm. We have performed experiments using proposals from both APT [175] and Tubelets [61]. We found APT to perform slightly better and report all results using APT.

**Features.** For each tube we extract Improved Dense Trajectories and compute HOG, HOF, Traj, MBH features [183]. The combined features are reduced to 128 dimensions through PCA and aggregated into a fixed-size representation using Fisher Vectors [141]. We construct a codebook of 128 clusters, resulting in a 54,656-dimensional representation per proposal.

**Training.** We train the proposal mining optimization for 10 iterations for all our evaluations, similar to Cinbis et al. [22]. Following further suggestions by [22], we randomly split the training videos into multiple (3) splits to train and select the instances. While training a classifier for one action, we randomly sample 100 proposals of each video from the other actions as negatives. We set the SVM regularization λ to 100.

**Evaluation.** During testing we apply the classifier to all proposals of a test video and maintain the top proposals per video. To evaluate the action localization performance, we compute the Intersection-over-Union (IoU) between proposal \( p \) and the box annotations of the corresponding test example \( b \) as: \( \text{iou}(p, b) = \frac{1}{|\Gamma|} \sum_{f \in \Gamma} \text{IoU}_{p,b}(f) \), where \( \Gamma \) is the set of frames where at least one of \( p, b \) is present [61]. The function \( \text{IoU} \) states the box overlap for a specified frame. For IoU threshold \( t \), a top selected proposal is deemed a positive detection if \( \text{iou}(p, b) \geq t \).

After combining the top proposals from all videos, we compute the Average Precision score using their ranked scores and positive/negative detections. For the comparison to
5.5 RESULTS

5.5.1 TRAINING WITHOUT GROUND TRUTH TUBES

First we evaluate our starting hypothesis of replacing ground truth tubes with proposals for training action localization classifiers. We compare three approaches: 1) train on ground truth annotated bounding boxes; 2) train on the proposal with the highest IoU overlap for each video; 3) train on the proposal mined based on point annotations and our proposal mining. For the points on both datasets, we take the center of each annotated bounding box.

Training with the best proposal. Figure 29 shows that the localization results for the best proposal are similar to the ground truth tube for both datasets and across all IoU overlap thresholds as defined in Section 5.4.2. This result shows that proposals are sufficient to train classifiers for action localization. The result is somewhat surprising given that the best proposals used to train the classifiers have a less than perfect fit with the ground truth action. We computed the fit with the ground truth, and on average the IoU score of the best proposals (the ABO score) is 0.642 on UCF Sports and 0.400 on UCF 101. The best proposals are quite loosely aligned with the ground truth. Yet, training on such non-perfect proposals is not detrimental for results. This means that a perfect fit with the action is not a necessity during training. An explanation for this result is that the action classifier is now trained on the same type of noisy samples that it will encounter at test-time. This better aligns the training with the testing, resulting in slightly improved accuracy.
Training with proposal mining from points. Figure 29 furthermore shows the localization results from training without bounding box annotations using only point annotations. On both data sets, results are competitive to the ground truth tubes across all thresholds. This result shows that when training on proposals, carefully annotated box annotations are not required. Our proposal mining is able to discover the best proposals from cheap point annotations. The discrepancy between the ground truth and our mined proposal for training is shown in Figure 30 for thee videos. For some videos, e.g. Figure 30a, the ground truth and the proposal have a high similarity. This does however not hold for all videos, e.g. Figures 30b, where our mined proposal focuses solely on the lifter (Lifting), and 30c, where our mined proposal includes the horse (Horse riding).

Analysis. On UCF 101, where actions are not temporally trimmed, we observe an average temporal overlap of 0.74. The spatial overlap in frames where proposals and ground truth match is 0.38. This result indicates that we are better capable of detecting actions in the temporal domain than the spatial domain. On average, top ranked proposals during testing are 2.67 times larger than their corresponding ground truth. Despite a preference for larger proposals, our results are comparable to the fully supervised method trained on expensive ground truth bounding box tubes. Finally, we observe that most false positives are proposals from positive test videos with an overlap score below the specified threshold. On average, 26.7% of the top 10 proposals on UCF 101 are proposals below the overlap threshold of 0.2. Regarding false negatives, on UCF 101 at a 0.2 overlap threshold, 37.2% of the actions are not among the top selected proposals. This is primarily because the proposal algorithm does not provide a single proposal with enough overlap.

From this experiment we conclude that training directly on proposals does not lead to a reduction in action localization accuracy. Furthermore, using cheap point annotations with our proposal mining yields results competitive to using carefully annotated bounding box annotations.
5.5 Results

Figure 31: The annotation speedup versus mean Average Precision scores on (a) UCF Sports and (b) UCF 101 for two overlap thresholds using both box and point annotations. The annotation frame-rates are indicated on the lines. Using points remains competitive to boxes with a 10x to 80x annotation speed-up.

5.5.2 Lowering the Annotation Frame-Rate

The annotation effort can be significantly reduced by annotating less frames. Here we investigate how a higher annotation frame-rate influences the trade-off between annotation speed-up versus classification performance. We compare higher annotation frame-rates for points and ground-truth bounding boxes.

Setup. For measuring annotation time we randomly selected 100 videos from the UCF Sports and UCF 101 datasets separately and performed the annotations. We manually annotated boxes and points for all evaluated frame-rates \{1, 2, 5, 10, \ldots\}. We obtain the points by simply reducing a bounding box annotation to its center. We report the speed-up in annotation time compared to drawing a bounding box on every frame. Classification results are given for two common IoU overlap thresholds on the test set, namely 0.2 and 0.5.

Results. In Figure 31 we show the localization performance as a function of the annotation speed-up for UCF Sports and UCF 101. Note that when annotating all frames, a point is roughly 10-15 times faster to annotate than a box. The reason for the reduction in relative speed-up between the higher frame-rates is due to the constant time spent on...
Figure 32.: Qualitative examples of the iterative proposal mining (blue) during training, guided by points (red) on UCF Sports. (a) and (b): the final best proposals have a significantly improved overlap (from 0.194 to 0.627 and from 0.401 to 0.526 IoU). (c): the final best proposal is the same as the initial best proposal, although halfway through the iterations, a better proposal was mined.

determining the action label of each video. When analyzing classification performance we note it is not required to annotate all frames. Although the performance generally decreases as less frames are annotated, using a frame rate of 10 (i.e. annotating 10% of the frames) is generally sufficient for retaining localization performance. We can get competitive classification scores with an annotation speedup of 45 times or more.

The results of Figure 31 show the effectiveness of our proposal mining after the iterative optimization. In Figure 32, we provide three qualitative training examples, highlighting the mining during the iterations. We show two successful examples, where mining improves the quality of the top proposal, and a failure case, where the proposal mining reverts back to the initially mined proposal.

Based on this experiment, we conclude that points are faster to annotate, while they retain localization performance. We recommend that at least 10% of the frames are annotated with a point to mine the best proposals during training. Doing so results in a 45 times or more annotation time speed-up.
5.5 results

Figure 33.: **Hollywood2Tubes**: Localization results for Hollywood2 actions across all overlap thresholds. The discrepancy between the recall and Average Precision indicates the complexity of the *Hollywood2Tubes* dataset for action localization.

### 5.5.3 Hollywood2Tubes: Action Localization for Hollywood2

Based on the results from the first two experiments, we are able to supplement the complete Hollywood2 dataset by Marszalek et al. [96] with action location annotations, resulting in *Hollywood2Tubes*. The dataset consists of 12 actions, such as *Answer a Phone*, *Driving a Car*, and *Sitting up/down*. In total, there are 823 train videos and 884 test videos, where each video contains at least one action. Each video can furthermore have multiple instances of the same action. Following the results of Experiment 2 we have annotated a point on each action instance for every 10 frames per training video. In total, there are 1,026 action instances in the training set; 29,802 frames have been considered and 16,411 points have been annotated. For the test videos, we are still required to annotate bounding boxes to perform the evaluation. We annotate every 10 frames with a bounding box. On both UCF Sports and UCF 101, using 1 in 10 frames yields practically the same IoU score on the proposals. In total, 31,295 frames have been considered, resulting in 15,835 annotated boxes. The annotations, proposals, and localization results are available at [http://tinyurl.com/hollywood2tubes](http://tinyurl.com/hollywood2tubes).

**Results.** Following the experiments on UCF Sports and UCF 101, we apply proposals [175] on the videos of the Hollywood2 dataset. In Figure 33a, we report the action localization test recalls based on our annotation efforts. Overall, a MABO of 0.47 is achieved. The recall scores are lowest for actions with a small temporal span, such as *Shaking hands* and *Answer a Phone*. The recall scores are highest for actions such as *Hugging a person* and *Driving a Car*. This is primarily because these actions almost completely fill the frames in the videos and have a long temporal span.

In Figure 33b, we show the Average Precision scores using our proposal mining with point overlap scores. We observe that a high recall for an action does not necessarily yield a high Average Precision score. For example, the action *Sitting up* yields an above average recall curve, but yields the second lowest Average Precision curve. The reverse
Figure 34.: **Hard scenarios for action localization** using Hollywood2Tubes, not present in current localization challenges. Highlighted are actions involving two or more people, actions partially defined by context, and co-occurring actions within the same video.

holds for the action *Fighting a Person*, which is a top performer in Average Precision. These results provide insight into the complexity of jointly recognizing and localizing the individual actions of Hollywood2Tubes. The results of Figure 33 shows that there is a lot of room for improvement.

In Figure 34, we highlight a difficult cases for action localization, which are not present in current localization datasets, adding to the complexity of the dataset. In Appendix A, we outline additional difficult cases, such as cinematographic effects and switching between cameras within the same scene, as well as further dataset statistics.

### 5.5.4 Comparison to the State-of-the-Art

In the fourth experiment, we compare our results using the point annotations to the current state-of-the-art on action localization using box annotations on the UCF Sports, UCF 101, and Hollywood2Tubes datasets. In Table 7, we provide a comparison to related work on all datasets. For the UCF 101 and Hollywood2Tubes datasets, we report results with the mean Average Precision. For UCF Sports, we report results with the Area Under the Curve (AUC) score, as the AUC score is the most used evaluation score on the dataset. All reported scores are for an overlap threshold of 0.2.

We furthermore compare our results to two baselines using other forms of cheap annotations. This first baseline is the method of Jain *et al.* [62] which performs zero-shot localization, *i.e.* no annotation of the action itself is used, only annotations from other actions. The second baseline is the approach of Cinbis *et al.* [22] using global labels, applied to actions.

**UCF Sports.** For UCF Sports, we observe that our AUC score is competitive to the current state-of-the-art using full box supervision. Our AUC score of 0.545 is, similar to Experiments 1 and 2, nearly identical to the APT score (0.546) [175]. The score is furthermore close to the current state-of-the-art score of 0.559 [49, 188]. The AUC scores for the two baselines without box supervision can not compete with our AUC scores. This result shows that points provide a rich enough source of annotations that are exploited by our proposal mining.

**UCF 101.** For UCF 101, we again observe similar performance to APT [175] and an improvement over the baseline annotation method. The method of Weinzaepfel *et
Table 7.: **State-of-the-art localization results** on the UCF Sports, UCF 101, and Hollywood2Tubes for an overlap threshold of 0.2. Where * indicates we run the approach of Cinbis et al. [22] intended for images on videos. Our approach using point annotations provides a profitable trade-off between annotation effort and performance for action localization.

Hollywood2Tubes. For Hollywood2Tubes, we note that approaches using full box supervision cannot be applied, due to the lack of box annotations on the training videos. We can still perform our approach and the baseline method of Cinbis et al. [22]. First, observe that the mean Average Precision scores on this dataset are lower than on UCF Sports and UCF 101, highlighting the complexity of the dataset. Second, we observe that the baseline approach using global video labels is outperformed by our approach using points, indicating that points provide a richer source of information for proposal mining than the baselines.

From this experiment, we conclude that our proposal mining using point annotations provides a profitable trade-off between annotation effort and performance for action localization.

5.6 CONCLUSIONS

We conclude that carefully annotated bounding boxes precisely around an action are not needed for action localization. Instead of training on examples defined by expensive bounding box annotations on every frame, we use proposals for training yielding similar results. To determine which proposals are most suitable for training we only require cheap point annotations on the action for a fraction of the frames. Experimental evaluation on the UCF Sports and UCF 101 datasets shows that: (i) the use of proposals over directly using the ground truth does not lead to a loss in localization performance, (ii) action localization using points is comparable to using full box supervision, while being
significantly faster to annotate, (iii) our results are competitive to the current state-of-the-art. Based on our approach and experimental results we furthermore introduce Hollywood2Tubes, a new action localization dataset with point annotations for train videos. The point of this paper is that valuable annotation time is better spent on clicking in more videos than on drawing precise bounding boxes.
LOCALIZING ACTIONS FROM VIDEO LABELS AND PSEUDO-ANNOTATIONS

6.1 INTRODUCTION

The goal of this paper is to determine the spatio-temporal location of actions such as Skateboarding and Shaking hands in video content. This challenging problem is typically solved by classifying sliding cuboids [72, 82, 166] action proposals [17, 61, 118, 157, 160], or by linking detectors over time [70, 140, 188, 205]. In all cases, precise box annotations for actions on training video are a prerequisite for localizing actions in test videos. We challenge the need for spatio-temporal box annotations and propose an intuitive and effective algorithm that localizes actions in video from a video label only.

For our goal of localization from video labels, we build upon the work of Chapter 5. For training, we start from unsupervised action proposals [175], typically about 1,000 sequences of bounding boxes that are generated automatically for a video. In Chapter 5, we show that using the best possible action proposal during training, rather than ground truth annotations, does not lead to a decrease in action localization accuracy. Encouraged by this observation, a variant of the Multiple Instance Learning algorithm [3] is introduced, able to mine proposals with a good spatio-temporal fit to actions of interest by letting humans annotate a limited amount of points on the action in relevant training frames. While surprisingly effective, the approach still demands human supervision beyond the action class label. In this paper, we also rely on unsupervised action proposals during training, but rather than selecting the best proposal using manual human point-supervision we prefer a completely automatic alternative.

We introduce the notion of pseudo-annotations, see Figure 35, which we define as visual cues that replace point-supervision in video. We investigate five of such pseudo-annotations by exploiting sources such as action proposals [175], object proposals [217], person detections [205], motion [61], and center biases [169] to discover which cues are most informative to point on the action locations. The pseudo-annotations specify the likely location of an action in a video, resulting in the automatic selection of a desirable action proposal during Multiple Instance Learning optimization, where the information from pseudo-annotations is combined with action-specific video labels. To automatically select and combine pseudo-annotations from different cues, we introduce a metric based on correlations between the pseudo-annotations.

Thorough evaluation on three action localization datasets shows that individually, each visual cue is informative for localizing actions. Using our correlation metric for
Figure 35.: We introduce pseudo-annotations from visual cues, indicated by different colored dots, that simulate supervision in videos. From the pseudo-annotations and action class labels, we automatically select action proposals (blue tube) for training action localizers.

selecting and combining annotations, we reach results comparable to action localization from full box supervision with the same proposal and classification settings, while outperforming other weakly-supervised alternatives. Furthermore, we demonstrate how pseudo-annotations can be leveraged during testing, to further improve any localization result, be it trained on pseudo-annotations or manually annotated boxes.

6.2 RELATED WORK

Yu and Yuan [205] introduce supervised actor proposals for action localization. They rely on a person detector on successive frames and generate spatio-temporal proposals by assuring sufficient overlap and appearance consistency. Gkioxari and Malik [49] replace the person detector by an action-specific detector using appearance and motion. They link regions with strong overlap over time. Weinzaepfel et al. [188] follow the same scheme, but rather than linking detections they prefer tracking by detection (using boxes and class label) for further fine-tuning over time. It is obvious that by adding more supervision to the action proposal generation, better localization can be achieved, especially with deep learning, see Saha et al. [140]. Rather than using class-specific action detectors and box supervision, we prefer to localize an action in video from its class label only.

Jain et al. [61] introduce unsupervised action proposals that are likely to include the action, ideally achieving high recall with few proposals. They start from super-voxels and group them based on color, texture, motion, size, fill cues, and independent motion. Van Gemert et al. [175] bypass the computationally expensive segmentation step of [61] by creating unsupervised proposals directly from dense trajectories [183] used to represent videos during classification. Chen and Corso [17] also advocate clusters of dense trajectories for unsupervised action proposals. We also rely on unsupervised action proposals, but rather than selecting the best proposals using a classifier that learns from box annotations, we learn from a class label only.

Chapter 5 of this thesis proposes to train action localization classifiers using unsupervised proposals as positive examples rather than ground truth boxes. We introduce a Multiple Instance Learning (MIL) algorithm that mines proposals with a good spatio-temporal fit to actions by including point supervision. It extends the traditional MIL objective with a measure that takes into account the overlap between proposals and points. The approach allows to localize actions in video from class labels and point annotations.
6.3 ACTION LOCALIZATION WITH PSEUDO-ANNOTATIONS

For training an action localizer, we are given a set of \( N \) training videos \( \{X_i, Y_i\}_{i=1}^N \), where \( X_i \in \mathbb{R}^{|A_i| \times D} \) states the \(|A_i|\) action proposals, each of feature dimension \( D \), and \( Y_i \in \{-1, +1\} \) indicates the video label, which is +1 if the action occurs anywhere in the video and −1 otherwise. Each action proposal \( A_i = \{A_i(t)\}_{t=1}^T \) is a tube consisting of \( T \) bounding boxes.

Our goal is to train a classifier using a proposal with high spatio-temporal overlap with the action of interest for each video. We employ a Multiple Instance Learning perspective [3, 22, 106]. Each video is a bag and the proposals in each video are the instances. Using a max-margin objective, the Multiple Instance Learning optimization is given as:

\[
\min_{w,b,\xi} \frac{1}{2}||w||^2 + \lambda \sum_{i} \xi_i, \quad \text{s.t.} \quad \forall_i : Y_i \cdot (w \cdot \arg \max_{z \in X_i} S(z|w, b, P)) \geq 1 - \xi_i, \quad \forall_i : \xi_i \geq 0
\]  

(6.1)

where \( S(z|w, b, P) \) specifies a selection function for proposal \( z \in X_i \), conditioned on both the classifier score \( (w, b) \) and (pseudo-)annotations \( P \).

In this work, we only require video labels. Therefore, we are tasked with automatically discovering annotations \( P \), dubbed pseudo-annotations. They exploit sources such as
Localizing actions from video labels and pseudo-annotations

6.3.1 PSEUDO-ANNOTATIONS

Pseudo-annotations from person detection. Actions are typically human-centered, so a robust detection of people in video frames provides information about the spatio-temporal location of actions. Here, we employ the Faster R-CNN network [134], using the person class after pre-training on MS-COCO [89]. After non-maximum suppression, the network yields roughly 50 box detections per frame, each with a confidence score. We select the bounding box in each frame with maximum confidence score as our pseudo-annotation.

Pseudo-annotations from independent motion. The independent motion at each pixel of a frame $F$ provides information as to where the foreground action occurs in the frame. Here, we employ the interpretation of independent motion from Jain et al. [61]. Independent motion states the deviation from the global motion present in a frame. Let $\xi(x,y,F) \in [0, 1]$ denote the inverse of the residual in the global motion estimation at pixel $(x,y)$ in frame $F$. The higher the value of $\xi(x,y,F)$, the less likely that the pixel contributes to the global motion. Then we compute a point-wise pseudo-annotation for frame $F$ as the center of mass over all the pixels in the frame, where the mass is given by their independent motion estimation:

$$p_{im}(F) = \frac{1}{\xi(F)} \sum_{(x,y) \in F} \xi(x,y,F) \cdot (x,y),$$

where $\xi(F)$ denotes the total independent motion in frame $F$.

Pseudo-annotations from action proposals. We furthermore examine the action proposals themselves as a source of information for pseudo-annotations, using the unsupervised spatio-temporal proposals of [175]. For a frame $F$ and action proposals $A^*$, we examine the spatial distribution of the proposal boxes of $A^*$ in the frame. We make the following assumption about the spatial distribution of the proposals: the more the action proposals are on the same spatial location, the higher the likelihood that the action occurs in that location. The use of action proposals for pseudo-annotations can be interpreted in two ways. First, it is a form of self-supervision [31], as we employ the action proposals to specify which action proposals to train on. Second, it is a form of outlier detection. If many proposals agree on the same location, we give a penalty to the proposals that are outside that location.

For each pixel $(x,y) \in F$, we denote the number of proposals from $A^*$ that contain $(x,y)$ as $C_{A^*}(x,y,F)$. We compute the pseudo-annotation as the center of mass over these counts:

$$p_{pa}(F) = \frac{1}{C_{A^*}(F)} \sum_{(x,y) \in F} C_{A^*}(x,y,F) \cdot (x,y),$$

where $C_{A^*}(F)$ denotes the sum of the proposal counts over all pixels in $F$. 

78
Pseudo-annotations from frame centers. In [106, 169], it is noted that both actions and annotators have a bias towards the center of the video. We exploit this bias directly by adding a point-wise pseudo-annotation on the center of each frame of each video:

\[ p_{fc}(F) = \left(\frac{F_W}{2}, \frac{F_H}{2}\right), \]

where \( F_W \) and \( F_H \) denote the width and height of frame \( F \).

Pseudo-annotations from object proposals. The presence of objects is also correlated with the presence of actions, as observed in [62,63]. Object proposals are computed here from EdgeBoxes [217], using the top 1,000 object proposals per frame. Similar to the action proposal pseudo-annotation, we compute the number of proposals containing the pixel for each pixel in frame \( F \). Let \( C_O(x,y,F) \) denote the number of proposals containing pixel \((x,y)\), then the pseudo-annotation is given as:

\[ p_{oa}(F) = \frac{1}{C_O(F)} \sum_{(x,y)\in F} C_O(x,y,F) \cdot (x,y). \]

where \( C_O(F) \) denotes the sum of the proposal count over all pixels in \( F \). The difference with Equation 6.3 is in that we assume here that the foreground action is the most dominant object in the scene, as defined by the number of object proposals focusing on the action.

6.3.2 Computing Pseudo-Annotation Overlaps

Each visual cue outputs an automatic box (person detection) or point (all others) annotation. Given an action proposal \( A \), we compute the overlap with the box annotations using the spatial-temporal intersection-over-union score [61]. The intersection-over-union with a set of box annotations \( B \) is computed as: 

\[ \frac{1}{|\Gamma|} \sum_{f \in \Gamma} \text{iou}(f_B, f_A), \]

where \( \Gamma \) denotes the set of frames with at least one of \( A \) and \( B \) present. For point-wise pseudo-annotation \( P \), we compute the overlap using the function defined in the previous Chapter: 


Here \( M(A,P) \) states the overlap between action proposal \( A \) and pseudo-annotations \( P \) and is defined as: 

\[ M(A,P) = \frac{1}{|P|} \sum_{i=1}^{|P|} \max \left(0, 1 - \frac{\| (P_{x}, P_{y}) - \bar{A}_{P_{i}} \|_2}{\max_{(u,v)\in(A_{P_{i}})} \| (u,v) - \bar{A}_{P_{i}} \|_2} \right), \]

where \( \bar{A}_{P_{i}} \) denotes the center of the box of proposal \( A \) in frame \( P_{i} \). In turn, \( S(A,V) \) is a size regularization on the action proposal itself: 

\[ S(A,V) = \frac{\sum_{m=1}^{|A|} |A_{m}|^2}{\sum_{m=1}^{|A|} |A_{m}|}, \]

where \( |A_{m}| \) denotes the area of a box, and \( V \) denotes the whole video. Intuitively, the overlap measure for point annotations aims to promote proposals with box centers close to the points while penalizing proposals of large size compared to the whole video.

6.3.3 Correlation Metric for Pseudo-Annotations

For a video \( v \), let \( \{S_v^{(i)}\}_{i=1}^{|P|} \) denote the overlap scores over all pseudo-annotations \( P \) and let \( S_v^{(i)} \in \mathbb{R}^{|A_v|} \) denote the overlap scores for the \( |A_v| \) action proposals of the \( i^{th} \) pseudo-annotation in the video. Since no supervision within the videos is provided, it is a priori
unknown how pseudo-annotations from different cues should be used and to what extent. Given the integral importance of people in detecting and localizing actions [205], we propose a correlation metric for pseudo-annotations using the person detection as an anchor for the correlation.

Let $H_v$ denote the overlap scores of the action proposals in video $v$ given by the person detection. Then we compute the statistical correlation between the pseudo-annotation of the $i^{th}$ cue and the pseudo-annotations from the person detection over all $N_t$ training videos:

$$\eta(P^{(i)}) = \frac{1}{N_t} \sum_{v=1}^{N_t} \frac{\text{cov}(S_v^{(i)}, H_v)}{\sigma(S_v^{(i)}) \cdot \sigma(H_v)}.$$  \hspace{1cm} (6.6)

The covariance and standard deviations in Eq. 6.6 are computed over the pseudo-annotation overlap scores of all action proposals in video $v$. Intuitively, Eq. 6.6 assigns a high score to pseudo-annotations that assign similar overlaps scores to the person detection; the more a pseudo-annotation agrees with the ranking of action proposals, the higher the correlation score. In turn, we can fuse the overlap scores of the pseudo-annotations as:

$$S_v^{\text{fused}} = \sum_{P^{(i)} \in P} \eta(P^{(i)}) \cdot [\eta(P^{(i)}) \geq t] \cdot S_v^{(i)},$$  \hspace{1cm} (6.7)

where $t$ is a threshold to remove pseudo-annotations with overlap scores too dissimilar to the person detection. Note that the person detection itself is also in the set $P$. In accordance with Eq. 6.6, the person detection yields a correlation score of 1.

The correlation metric for pseudo-annotations provides a way of measuring the quality of pseudo-annotations without the need for manual box or point annotations, nor the need for examining test performance to combine and select pseudo-annotations.

By using a single pseudo-annotation per frame for each cue, we assume a single dominant action in each video. This assumption holds throughout our experiments. To handle videos with multiple actions and objects we can extend our approach with a density estimation over the pixel-wise weight of each cue to estimate multiple pseudo-annotations in frames.

6.4 EXPERIMENTAL SETUP

6.4.1 DATASETS

UCF Sports. The UCF Sports dataset consists of 150 videos from sport broadcasts covering 10 action categories [135], including Diving, Riding a Horse, and Skateboarding. We employ the train and test data split as suggested in [82].

UCF 101. The UCF 101 dataset has 101 actions categories [159] where 24 categories have spatio-temporal action localization annotations. This subset has 3,204 videos, where each video contains a single action category, but might contain multiple instances of the same action. We use the first split of the train and test sets as suggested in [159].

Hollywood2Tubes. The Hollywood2Tubes dataset consists of 1,707 videos with ground truth point (training videos) and box (test videos) annotations [106]. The dataset contains the actions from the Hollywood2 dataset [96], including Getting out of a car, Hugging,
and Fighting. We use the train and test data split as suggested in [96].

**A2D.** The A2D dataset contains 3,782 videos of actions performed both by human actors and other actors, such as dogs, cars, and babies [194]. For a limited number of video frames, box annotations are provided. We use the train and test split as suggested in [194].

We stress that throughout our experiments, we do not employ any manual point or box annotations for our approach.

### 6.4.2 IMPLEMENTATION DETAILS

**Proposals.** Following [106], we employ the unsupervised action proposals from [175]. We note that [175] only rely on dense trajectories for creating the proposals and do not use the cues that we employ for the pseudo-annotations.

**Proposal representations.** On all datasets, we represent each action proposal with a Fisher Vector [141] with 128 clusters over the improved dense trajectories [183] within the proposal. This results in a 54,656-dimensional representation per proposal.

**Training.** We train the Multiple Instance Learning algorithm for 5 iterations for all evaluations. Following [22], we split the training videos into multiple folds during training for the classifier and proposal selection steps. We set the regularization parameter $\lambda$ in the max-margin optimization to 10 in all experiments.

**Evaluation.** During testing we apply the classifier of an action to all proposals of a test video and keep the proposal with the highest classifier score [61, 175]. To evaluate the action localization performance, we compute the intersection-over-union in space and time between the top proposal and a ground truth tube as defined in [61]. Only proposals whose overlaps with ground truths exceed the threshold are considered correct.

### 6.5 EXPERIMENTAL RESULTS

#### 6.5.1 EVALUATING THE PSEUDO-ANNOTATIONS

In the first experiment, we evaluate each pseudo-annotation individually for action localization on UCF Sports and UCF-101 with the mean Average Precision score. We compare the pseudo-annotations to two baselines. The first baseline uses full box supervision during training (light gray area). This baseline serves as a supervision upper bound. The second baseline uses the video labels with standard Multiple Instance Learning (dark gray area). This baseline serves as the supervision lower bound. Note that all approaches use the same features and classifier settings.

The scores across all overlap thresholds are shown in Figure 37. On UCF Sports, we observe that each pseudo-annotation performs better than only using the video label, which means that pseudo-annotations provide meaningful information about the location of actions in videos. Furthermore, the person detection performs best, followed by action proposals and independent motion. At low overlap thresholds, these approaches even outperform full supervision. This is surprising, since no human intervention is provided. At higher thresholds, full supervision is still better, while all approaches break down at the highest thresholds.
localizing actions from video labels and pseudo-annotations

Out of the 5 categories, the person detection pseudo-annotation performs best, followed by using frame centers and action proposals. To highlight the effect and limitations of the pseudo-annotations, we show qualitative results in Figure 38.

6.5.2 COMBINING PSEUDO-ANNOTATIONS

In the second experiment, we evaluate the correlation metric for pseudo-annotations. In Figure 39a, we show the correlation scores on UCF Sports. The scores show that person detection, action proposals, and independent motion pseudo-annotations are most relevant, while frame centers and object proposals are less relevant. The discovered order is in line with the order of performance from the first experiment. This means that the correlation metric provides a reliable way of measuring the quality of pseudo-annotations, while automatic selection is possible by using the ones with highest average correlation.

We provide the localization performance in Figures 39b and 39c. For UCF Sports, when using the correlation metric with both the top-two and top-three pseudo-annotations yields results comparable but not identical to full box supervision. When using more pseudo-annotations, the performance at higher overlap thresholds degrades, indicating that not all pseudo-annotations should be used in the combination. On UCF-101 (Figure 39c), using correlation metric with the top pseudo-annotations also yields results comparable or close to full box supervision. Here, the combination using person detection and frame centers (the second highest correlation pseudo-annotation) performs best. Incorporating more pseudo-annotations slightly degrades the performance.

We conclude from this experiment that a correlation metric from the top pseudo-annotations provides a reliable way to merge different visual cues for action localization. On both datasets, the metric with the top 2/3 pseudo-annotations outperform person detections only, while performing comparable to full box supervision.

Non-human action localization. The datasets typically used in action localization are human-centric [106, 135, 159]. Here, we evaluate how well our pseudo-annotations generalize to actions performed by non-human actors using the A2D dataset [193, 194].

Figure 37.: Pseudo-annotation action localization performance on UCF Sports (left) and UCF-101 (right), compared to the supervision upper and lower bounds.

On UCF-101, the pseudo-annotations similarly all outperform the approach using the video label only. The difference between the approaches is smaller since the dataset is larger, making it more robust against accidental hits and misses of the pseudo-annotations. The person detection pseudo-annotation performs best, followed by using frame centers and action proposals. To highlight the effect and limitations of the pseudo-annotations, we show qualitative results in Figure 38.
6.5 EXPERIMENTAL RESULTS

Figure 38.: **Qualitative examples of pseudo-annotations** in non-trivial action videos. In (a), the pseudo-annotations successfully follow the centrally oriented main action. In (b), some pseudo-annotations are distracted either by complex background (*Swinging on a bar*, left), or due to lack of primary action and the presence of other people (*Golf swinging*, right).

These actors include babies, balls, birds, cars, cats, and dogs. Since this dataset does not have action tube annotations, our approach can not be directly evaluated. Individual box annotations for a set of frames per video are provided instead. Therefore, we investigate whether pseudo-annotations are capable of “pointing at” actions performed both by human and non-human actors. We evaluate how the overlap between proposals and pseudo-annotations relates to the overlap between proposals and ground truth boxes.

Over all actors, we find that the Pearson correlation score is 0.29; a high score for the pseudo-annotations correlates with a high score in action overlap, which strengthens the notion of pseudo-annotations for action localization. We also find that the Pearson correlation score is positive for all actor types individually. We do note that the score is higher for the person actor than the other types, indicating a bias towards persons as actors in our pseudo-annotations. Interestingly, when excluding the person detection as pseudo-annotation, person remains the most positively correlated actor type. We conclude that our pseudo-annotations are not restricted to the person as actor type and handle other actor types as well. The person as actor type does have closest relations to the pseudo-annotations, although this is not solely due to the use of the person detection itself.
Localizing actions from video labels and pseudo-annotations

(a) Correlations (UCF Sports). (b) Correlations (UCF Sports). (c) Correlations (UCF Sports).

Figure 39.: Correlation-based combination of pseudo-annotations. On UCF Sports and UCF-101, automatically combining the top two or three correlated pseudo-annotations yields the best results, even comparable to full box supervision.

<table>
<thead>
<tr>
<th></th>
<th>UCF Sports</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>UCF 101</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Full box annotations</td>
<td>77.1</td>
<td>73.1</td>
<td>60.9</td>
<td>42.2</td>
<td>26.8</td>
<td>10.6</td>
<td>46.1</td>
<td>34.5</td>
<td>24.4</td>
<td>13.9</td>
</tr>
<tr>
<td>Pseudo-annotations</td>
<td>88.0</td>
<td>77.2</td>
<td>57.4</td>
<td>49.7</td>
<td>21.0</td>
<td>11.1</td>
<td>50.3</td>
<td>35.1</td>
<td>22.2</td>
<td>9.3</td>
</tr>
<tr>
<td>Pseudo-annotations ++</td>
<td>87.7</td>
<td>81.7</td>
<td>64.4</td>
<td>54.5</td>
<td>37.8</td>
<td>17.5</td>
<td>49.8</td>
<td>37.4</td>
<td>25.8</td>
<td>13.7</td>
</tr>
<tr>
<td>Full box annotations ++</td>
<td>86.7</td>
<td>86.7</td>
<td>74.0</td>
<td>61.2</td>
<td>42.3</td>
<td>23.1</td>
<td>50.6</td>
<td>40.8</td>
<td>28.8</td>
<td>17.5</td>
</tr>
</tbody>
</table>

Table 8.: Localization performance (%) with pseudo-annotations during testing (++) Using pseudo-annotations during testing increases the performance across all overlap thresholds and datasets, even outperforming full box supervision. Pseudo-annotations can also be employed to improve models trained with full box supervision.

6.5.3 PSEUDO-ANNOTATIONS AT TEST TIME

Since pseudo-annotations are automatically generated for videos, their use is not restricted to training videos only. In the third experiment, we employ pseudo-annotations during testing to help select the best proposal per video. We do this by combining the classifier score with the overlap scores from the pseudo-annotations. We employ the correlation metric with the top pseudo-annotations for this experiment.

Results on UCF Sports and UCF-101 are shown in Table 8. On both datasets, we observe a jump in performance when adding pseudo-annotations during testing, even outperforming the full box supervision results. This performance shows the effectiveness of the pseudo-annotations for action localization. We also evaluate the effect of using pseudo-annotations during testing with a model trained on full box supervision, which yields a similar increase in performance. We conclude from this experiment that pseudo-annotations during testing improves any model trained on unsupervised proposals. With only the video labels as manual annotations, we even outperform the standard full box supervision setup.
6.6 Conclusions

In this work, we introduce pseudo-annotations for localizing actions in videos. We investigate pseudo-annotations from person detection, independent motions, action proposals, center biases, and object proposals. Using a correlation metric for pseudo-annotations, we reach results comparable or better to using full box supervision with the same settings, while outperforming other weakly-supervised approaches. As our approach relies on action class labels as the only manual annotations, it enables action localization on any action classification dataset, such as Sports 1M [71], ActivityNet [12], and EventNet [203].

Table 9. Localization results (%) at an overlap of 0.2. A dash (-) states that results are not provided, while n.a. states that the approach can not be applied due to the dataset’s lack of required annotations. The sign (++) denotes the use of pseudo-annotations during testing. We achieve results comparable to approaches that train on unsupervised proposals and box annotations, while outperforming approaches using video labels or zero-shot information considerably.

6.5.4 Comparison to State-of-the-Art

We compare our results on three action localization datasets to the current state-of-the-art. In Table 9, we show the performance of the methods, ordered by their level of supervision. To maximize the number of comparisons, we show the results at a threshold of 0.2.

On all datasets, we perform comparable to approaches that rely on expensive manual box or point annotation during training and unsupervised proposals during testing. This result is encouraging, as it states that video labels and automatic pseudo-annotations can provide enough information for localization. We improve over approaches using only video labels [22] or zero-shot information [62], resulting in state-of-the-art performance on the Hollywood2Tubes dataset. We also compare against the approaches of Weinzaepfel et al. [188] and Saha et al. [140]. On UCF Sports, we achieve comparable AUC scores. On UCF-101, these approaches report higher scores. While effective, these approaches require full box supervision both for making proposals and training action classifiers. They can therefore not generalize to weaker forms of supervision.

6.6 Conclusions

In this work, we introduce pseudo-annotations for localizing actions in videos. We investigate pseudo-annotations from person detection, independent motions, action proposals, center biases, and object proposals. Using a correlation metric for pseudo-annotations, we reach results comparable or better to using full box supervision with the same settings, while outperforming other weakly-supervised approaches. As our approach relies on action class labels as the only manual annotations, it enables action localization on any action classification dataset, such as Sports 1M [71], ActivityNet [12], and EventNet [203].
7.1 INTRODUCTION

We strive for the localization and classification of human actions like Walking a dog and Skateboarding without the need for any video training examples. The common approach in this challenging zero-shot setting is to transfer action knowledge via a semantic embedding build from attributes [75, 91, 197] or objects [14, 62, 192]. As the semantic embeddings are defined by image or video classifiers, they are unable, nor intended, to capture the spatial interactions an actor has with its environment. Hence, it is hard to distinguish who is Throwing a baseball and who is Hitting a baseball when both actions occur within the same video. We propose a spatial-aware object embedding for localization and classification of human actions in video, see Figure 40.

We draw inspiration from the supervised action classification literature, where the spatial connection between actors and objects has been well recognized, e.g. [52, 111, 190]. Early work focused on capturing actors and objects implicitly in a low-level descriptor [20, 200], while more recently the benefit of explicitly representing detected objects [40], their scores, and spatial properties was proven effective [55, 172, 201, 202]. Both [36] and [129] demonstrate the benefit of temporal actor and object interaction, by linking detected bounding boxes over time via trackers. By doing so, they are also capable of (supervised) action localization. We also detect actors and objects, and link them over time to capture spatio-temporal interactions. Different from all of the above works, we do no rely on any action class and/or action video supervision to get to our recognition. Instead, we introduce an embedding built upon actor and object detectors that allows for zero-shot action classification and localization in video.

Our main contribution is a spatial-aware object embedding for zero-shot action localization and classification. The spatial-aware embedding incorporates word embeddings, box locations for actors and objects, as well as their spatial relations, to generate action tubes. This enables us to both classify videos and to precisely localize where actions occur. Our spatial-aware embedding is naturally extended with contextual awareness from global objects. We furthermore show how our embedding generalizes to any query involving objects, spatial relations, and their sizes in a new spatio-temporal action retrieval scenario. Action localization and classification experiments on four contemporary action video datasets support our proposal.

Accepted to International Conference on Computer Vision 2017 [102].
7.2 RELATED WORK

7.2.1 SUPERVISED ACTION LOCALIZATION / CLASSIFICATION

A wide range of works have proposed representations to classify actions given video examples. Such representations include local spatio-temporal interest points [84, 93, 189] and dense trajectories [2, 183], typically aggregated into VLAD or Fisher vector representations [119, 125]. Recent works focus on learning global representations from deep networks, pre-trained on optical flow [148] or large-scale object annotations [63, 71, 199]. We also rely on deep representations for our global objects, but we emphasize on local objects and we aim to classify and localize actions without the need for any video example.

For spatio-temporal action localization, a popular approach is to split videos into action proposals; spatio-temporal tubes in videos likely to contain an action. Annotated tubes from example videos are required to train a model to select the best action proposals at test time. Action proposal methods include merging supervoxels [61, 157], merging trajectories [17, 106], and detecting actors [205]. The current state-of-the-art action localizers employ Faster R-CNN [134] trained on bounding box annotations of actions in
7.3 SPATIAL-AWARE OBJECT EMBEDDINGS

In our zero-shot formulation, we are given a set of test videos $\mathcal{V}$ and a set of action class names $\mathcal{Z}$. We aim to classify each video to its correct class and to discover the spatio-temporal tubes encapsulating each action in all videos. To that end, we propose a
spatial-aware object embeddings

Figure 41.: Examples of preferred spatial relations of objects relative to actors. In line with our intuition, skateboards are typically on or below the actor, while bicycles are typically to the left or right of actors and traffic lights are above the actors.

spatial-aware embedding; scored action tubes from interactions between actors and local objects. We present our embeddings in three steps: (i) gathering prior knowledge on actions, actors, objects, and their interactions, (ii) computing spatial-aware embedding scores for bounding boxes, and (iii) linking boxes into action tubes.

7.3.1 PRIOR KNOWLEDGE

Local object detectors. We first gather a set of local detectors pre-trained on images. Let $O = \{O_D, O_N\}$ denote the objects with detectors $O_D$ and names $O_N$. Furthermore, let $A = \{A_D, actor\}$ denote the actor detector. Each detector outputs a set of bounding boxes with corresponding object probability scores per video frame.

Textual embedding. Given an action class name $Z \in Z$, we aim to select a sparse subset of objects $O_Z \subset O$ relevant for the action. For the selection, we rely on semantic textual representations as provided by word2vec [108]. The similarity between object $o$ and the action class name is given as:

$$w(o, Z) = \cos(e(o_N), e(Z)),$$

where $e(\cdot)$ states the word2vec representation of the name. We select the objects with maximum similarity to the action.

Actor-object relations. We exploit that actors interact with objects in preferred spatial relations. To do so, we explore where objects tend to occur relative to the actor. Since we can not learn precise spatial relations between actors and objects from examples, we aim to use common spatial relations between actors and objects, as can be mined from large-scale image data sets. We discretize the spatial relations into nine relative positions, representing the preposition in front of and the eight basic prepositions around the actor, i.e. left of, right of, above, below, and the four corners (e.g. above left). For each object, we obtain a nine-dimensional distribution specifying its expected location relative to the actor, as detailed in Figure 41.
7.3 Spatial-Aware Object Embeddings

Figure 42: Example of our spatial-aware embedding. The actor sitting on the left horse (green box) is most relevant for the action Riding horse based on the actor detection, horse detection, and spatial relations between actors and horses.

7.3.2 Scoring Actor Boxes with Object Interaction

We exploit our sources of prior knowledge to compute a score for the detected bounding boxes in all frames of each test video \( V \in \mathcal{V} \). Given a bounding box \( b \) in frame \( F \) of video \( V \), we define a score function that incorporates the presence of (i) actors, (ii) relevant local objects, and (iii) the preferred spatial relation between actors and objects. A visual overview of the three components is shown in Figure 42. More formally, we define a score function for box \( b \) given an action class \( Z \) as:

\[
s(b, F, Z) = p(\text{Actor}|b) + \sum_{o \in \mathcal{O}_Z} r(o, b, F, Z), \tag{7.2}
\]

where \( p(\text{Actor}|b) \) is the probability of an actor being present in bounding box \( b \) as specified by the detector \( \text{Actor} \). The function \( r \) expresses the object presence and relation to the actor, it is defined as:

\[
r(o, b, F, Z) = w(o, Z) \cdot \left( \max_{f \in F_n} p(o|f) \cdot m(o, \text{Actor}, b, f) \right), \tag{7.3}
\]

where \( w(o, Z) \) states the semantic relation score between object \( o \) and action \( Z \) and \( F_n \) states all bounding boxes within the neighbourhood of box \( b \) in frame \( F \). The second part of Equation 7.3 states that we are looking for a box \( f \) around \( b \) that maximizes the joint probability of the presence of object \( o \) (the function \( p(o|f) \)), the match between the spatial relations of \( (b, f) \) and the prior relations of the actor and object \( o \) (the function \( m \)). We define the spatial relation match as:

\[
m(o, \text{Actor}, b, f) = 1 - JS\Delta_2(d(o, \text{Actor}, b) || d(b, f)), \tag{7.4}
\]

where \( JS\Delta_2(\cdot || \cdot) \in [0, 1] \) denotes the Jensen-Shannon Divergence with base 2 logarithm [88]. Intuitively, the Jensen-Shannon Divergence, a symmetrized and bounded variant of the Kullback-Leibler divergence, determines to what extent the two 9-dimensional distributions match. The more similar the distributions, the lower the divergence, hence the need for the inversion as we aim for maximization.

7.3.3 Linking Spatial-Aware Boxes

The score function of Equation 7.2 provides a spatial-aware embedding score for each bounding box in each frame of a video. We apply the score function to the boxes of all...
actor detections in each frame. We form tubes from the individual box scores by linking them over time [49]. We link those boxes over time that by themselves have a high score from our spatial-aware embedding and have a high overlap amongst each other. This maximization problem is solved using dynamic programming with the Viterbi algorithm. Once we have a tube from the optimization, we remove all boxes from that tube and compute the next tube from the remaining boxes.

Let \( T \) denote a discovered action tube in a video. The corresponding score is given as:

\[
t_{\text{emb}}(T, Z) = \frac{1}{|T|} \sum_{t \in T} s(t_b, t_F, Z),
\]

(7.5)

where \( t_b \) and \( t_F \) denote a bounding box and the corresponding frame in tube \( T \).

In summary, we propose spatial-aware object embeddings for actions; tubes through videos by linking boxes based on the zero-shot likelihood from the presence of actors, the presence of relevant objects around the actors, and the expected spatial relations between objects and actors.

### 7.4 LOCAL AND GLOBAL OBJECT INTERACTION

To distinguish tubes from different videos in a collection, contextual awareness in the form of relevant global object classifiers is also a viable source of information. Here, we first outline how to obtain video-level scores based on object classifiers. Then, we show how to compute spatial- and global-aware embeddings for action localization, classification, and retrieval.

#### 7.4.1 SCORING VIDEOS WITH GLOBAL OBJECTS

Let \( \mathcal{G} = \{G_C, G_N\} \) denote the set of global objects with corresponding classifiers and names. Different from the local objects \( \mathcal{O} \), these objects provide classifier scores over a whole video. Given an action class name \( Z \), we again select the top relevant objects \( \mathcal{G}_Z \subset \mathcal{G} \) using the textual embedding. The score of a video \( V \) is then computed as a linear combination of the word2vec similarity and classifier probabilities over the top relevant objects:

\[
t_{\text{global}}(V, Z) = \sum_{g \in \mathcal{G}_Z} w(g, Z) \cdot p(g|V),
\]

(7.6)

where \( p(g|V) \) denotes the probability of global object \( g \) of being in video \( V \).

#### 7.4.2 SPATIAL- AND GLOBAL-AWARE EMBEDDING

The information from local and global objects is combined into a spatial- and global-aware embedding. Here, we show how this embedding is employed for spatial-aware action localization, classification, and retrieval.
**Action localization.** For localization, we combine the tube score from our spatial-aware embedding with the video score from the global objects into a score for each individual tube $T$ as:

$$t(T, V, Z) = t_{\text{emb}}(T, Z) + t_{\text{global}}(V, Z).$$

(7.7)

We note that incorporating scores from global objects does not distinguish tubes from the same video. The global scores are however discriminative for distinguishing tubes from different videos in a collection $\mathcal{V}$. We compute the final score for all tubes of all videos in $\mathcal{V}$ using Equation 7.7. We then select the top scoring tubes per video, and rank the tubes over all videos based on their scores for localization.

**Action classification.** For classification purposes, we are no longer concerned about the precise location of the tubes from the spatial-aware embeddings. Therefore, we compute the score of a video $V$ given an action class name $Z$ using a max-pooling operation over the scores from all tubes $T_V$ in the video. The max-pooled score is then combined with the video score from the global objects. The predicted class for video $V$ is determined as the class with the highest combined score:

$$c^*_V = \arg \max_{Z \in \mathcal{Z}} \left( \max_{T \in T_V} t_{\text{emb}}(T, Z) + t_{\text{global}}(V, Z) \right).$$

(7.8)

**Spatial-aware action retrieval.** Spatial-aware action retrieval from user queries resembles action localization, i.e. rank the most relevant tubes the highest. However, different from localization, we now have the opportunity to specify actor and object relations via the search query. Given the effectiveness of size in actor-object interactions [36], we can also allow users to specify a relative object size $r$. By altering the size of queries objects, different localizations can be retrieved of the same action. To facilitate spatial-aware action retrieval, we alter the spatial relation match of Equation 7.4 with a match for a specified relative object size:

$$q(o, A, b, f, r) = m(o, A, b, f) + \left(1 - \frac{s(b)}{s(f)} - r\right),$$

(7.9)

where $s(\cdot)$ denotes the size of a bounding box. Substituting the spatial relation match with Equation 7.9, we again rank top scoring tubes, but now by maximizing a match to user-specified objects, spatial relations, and relative size.

7.5 EXPERIMENTAL SETUP

7.5.1 DATASETS

**UCF Sports** consists of 150 videos from 10 sport action categories, such as *Skateboarding*, *Horse riding*, and *Walking* [135]. We employ the test split as suggested in [82].

**UCF 101** consists of 13,320 videos from 101 action categories, such as *Skiing*, *Basketball dunk*, and *Surfing* [159]. We use this dataset for classification and use the test splits as provided in [159], unless stated otherwise.
J-HMDB consists of 928 videos from 21 actions, such as Sitting, Laughing, and Dribbling [65]. We use the bounding box around the binary action masks as the spatio-temporal annotations for localization. We use the test split as suggested in [65].

Hollywood2Tubes consists of 1,707 videos from the Hollywood2 dataset [96], supplemented with spatio-temporal annotations for localization [106]. Actions include Fighting with a person, Eating, and Getting out of a car. We use the test split as suggested in [96].

7.5.2 IMPLEMENTATION DETAILS

Textual embedding. To map the semantics of actions to objects, we employ the skip-gram network of word2vec [108] trained on the metadata of the images and videos from the YFCC100M dataset [165]. This model outputs a 500-dimensional representation for each word. If an action or object consists of multiple words, we average the representations of the individual words [62].

Actor and object detection. For the detection of both the actors and the local objects, we use Faster R-CNN [134], pre-trained on the MS-COCO dataset [89]. This network consists of the actor class and 79 other objects, such as snowboard, horse, and toaster. After non-maximum suppression, we obtain roughly 50 detections for each object per frame. We apply the network to each frame (UCF Sports, J-HMDB), or each 5th frame (UCF 101, Hollywood2Tubes) followed by linear interpolation.

Spatial relations. The spatial relations between actors and objects are also estimated from the MS-COCO dataset. For each object instance, we examine the spatial relations with the closest actor (if any actor is close to the object). We average the relations over all instances for each object.

Object classification. For the global objects, we employ a GoogLeNet network [164], pre-trained on a 12,988-category shuffle [101] of ImageNet [26]. This network is applied to each 5th frame of each video. For each frame, we obtain the object probabilities at the softmax layer and average the probabilities over the entire video. Following [62], we select the top 100 most relevant objects per action.

Evaluation. For localization, we compute the spatio-temporal intersection-over-union between top ranked actor tubes and ground truth tubes. We report results using both the (mean) Average Precision and AUC metrics. For classification, we evaluate with mean class accuracy.

7.6 EXPERIMENTAL RESULTS

7.6.1 SPATIAL-AWARE EMBEDDING PROPERTIES

In the first experiment, we focus on the properties of our spatial-aware embedding, namely the number of local objects to select and the influence of the spatial relations. We also evaluate qualitatively the effect of selecting relevant objects per action. We evaluate these properties on the UCF Sports dataset for both localization and classification.

Influence of local objects. We evaluate the performance using three settings of our embeddings. The first setting is using solely the actor detections for scoring bounding
Table 10.: **Influence of spatial awareness.** On UCF Sports we compare our spatial-aware object embedding to two other embeddings; using only the actors and using actors with objects, while ignoring their spatial relations. Our spatial-aware embedding is preferred for both localization (one object per action) and classification (five objects per action).

<table>
<thead>
<tr>
<th></th>
<th>Localization (mAP @ 0.5)</th>
<th>Classification (mean accuracy)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># local objects</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Embedding I: <em>Actor-only</em></td>
<td>0.083</td>
<td>-</td>
</tr>
<tr>
<td>Embedding II: <em>Actors and objects</em></td>
<td>-</td>
<td>0.175</td>
</tr>
<tr>
<td>Embedding III: <em>Spatial-aware</em></td>
<td>-</td>
<td>0.221</td>
</tr>
</tbody>
</table>

In Table 10, we provide both the localization and classification results. For localization using only the actor results in tubes that might overlap well with the action of interest, but there is no direct means to separate tubes containing different actions. This results in low Average Precision scores. For classification, using only the actor results in weak accuracy scores. This is because there is again no mechanism to discriminate videos containing different actions.

The second row of Table 10 shows the result when incorporating local object detections. For both localization and classification, there is a considerable increase in performance, indicating the importance of detections of relevant objects for zero-shot action localization and classification.

In the third row of Table 10, we show the performance of our spatial-aware embedding. The embedding outperforms the other settings for both localization and classification. This result shows that gathering and capturing information about the relative spatial locations of objects and actors provides valuable information about actions in videos. The spatial-aware embedding is most beneficial for the action *Riding a horse* (from 0.03 to 0.75 mAP), due to the consistent co-occurrence of actors and horses. Contrarily, the performance for *Running* remains unaltered, which is because no object relevant to the action is amongst the available detectors.

We have additionally performed an experiment with finer grid sizes on UCF Sports. For localization with the top-5 objects, we reach an mAP of 0.170 (4x4 grid) and 0.171 (5x5 grid), compared to a score of 0.199 with the 3x3 grid. Overall, the scores decrease slightly with finer grid sizes, indicating that coarse spatial relations are preferred over fine spatial relations.

**How many local objects?** In Table 10 we also consider how many relevant local objects to maintain per action. For localization, we observe a peak in performance using the top-1 local object per action, with a mean Average Precision (mAP) of 0.221 at an overlap threshold of 0.5; a sharp increase in performance over the 0.083 mAP using only the actor. When more objects are used, the performance of our embeddings degrades...
spatial-aware object embeddings

Figure 43: **Qualitative action localization results.** For Skateboarding and Riding a horse, relevant objects (blue) aid our localization (red). For Swinging on a bar and Kicking, incorrectly selected objects result in incorrect localizations. We expect that including more object detectors into our embedding will further improve results.

Slightly, indicating that actors are more likely to interact with a single object than multiple objects on a local level. At least for the UCF Sports dataset.

For classification, we observe a reverse correlation; the more local objects in our embedding, the higher the classification accuracy. This result indicates that for classification, we want to aggregate more information about object presence in videos, rather than exploit the single most relevant object per action. This is because a precise overlap with the action in each video is no longer required for classification. We exploit this relaxation with the max-pooling operation in the video-level scoring of Equation 7.8.

**Selecting relevant objects.** In our zero-shot formulation, a correct action recognition depends on detecting objects relevant to the action. We highlight the effect of detecting relevant objects in Figure 43. For successful actions such as Skateboarding and Riding a horse, the detection of respectively skateboards and horses help to generate a desirable action localization. For the actions Swinging on a bar and Kicking, the top selected objects are however incorrect, either because no relevant object is available or because of ambiguity in the word2vec representations.

**Conclusions.** We conclude from this experiment that our spatial-aware embedding is preferred over only using the actor and using actors and objects without spatial relations. Throughout the rest of the experiments, we will employ the spatial-aware embedding, using the top-1 object for localization and the top-5 for classification.
7.6 EXPERIMENTAL RESULTS

In the second experiment, we focus on the localization and classification performance when incorporating contextual awareness from global object scores into the spatial-aware embedding. We perform the evaluation on the UCF Sports dataset.

**Effect on localization.** In Figure 44, we show the AUC scores across several overlap thresholds. We show the results using our spatial-aware embedding and the combined spatial- and global-aware embedding.

We observe that across all overlap thresholds, adding global object classifier scores to our spatial-aware embedding improves the localization performance. This result indicates the importance of global object classification scores for discriminating tubes from different videos. The increase in performance is most notable at lower overlap thresholds, which we attribute to the fact that no localization information is provided by the global objects. The higher the overlap threshold, the more important selecting the right tube in each video becomes, and consequently, the less important the global object scores become.

**Effect on classification.** In Table 11, we show the classification accuracies on the UCF Sports dataset. We first observe that our spatial-aware embedding yields results competitive to the global object approach of Jain et al. [62], who also report zero-shot classification on UCF Sports. We also observe a big leap in performance when using our spatial- and global-aware embedding, with an accuracy of 0.645. We note that the big improvement is partially due to our deep network for the global object classifiers,
Table 11.: Local and global object interaction effect on classification. Adding global object awareness improves our spatial-aware object embedding considerably on UCF Sports.

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.100</td>
</tr>
<tr>
<td>Jain et al. [62]</td>
<td>0.264</td>
</tr>
<tr>
<td>Spatial-aware embedding</td>
<td>0.255</td>
</tr>
<tr>
<td>Spatial- and global-aware embedding</td>
<td>0.645</td>
</tr>
</tbody>
</table>

Figure 45.: Spatial-aware action retrieval. Top retrieved results on J-HMDB given specified queries. Our retrieved localizations (red) reflect the prescribed object (blue), spatial relation, and object size.

namely a GoogleNet trained on 13k objects [101]. We have therefore also performed an experiment with our spatial- and global-aware embedding using the network of [62]. We achieved a classification accuracy of 0.374, still a considerable improvement over the accuracy of 0.264 reported in [62].

Conclusion. We conclude from this experiment that including global object classification scores into our spatial-aware embedding improves both the zero-shot localization and classification performance. We will use this embedding for our comparison to related zero-shot action works.

7.6.3 SPATIAL-AWARE ACTION RETRIEVAL

For the third experiment, we show qualitatively that our spatial-aware embedding is not restricted to specific action queries and spatial relations. We show that any object, any spatial relation, and any object size can be specified as a query for spatial-aware action retrieval. For this experiment, we rely on the test videos from J-HMDB. In Figure 45, we show three example queries and their top retrieved actions.

The example on the left shows how we can search for a specific combination of actor, object, and spatial relation. The examples in the middle and right show that specifying different sizes for the query object leads to different action localization. The middle example shows an interaction with a baseball, while the right example shows an interaction with a soccer ball, which matches with the desired object sizes in the queries.
7.6 Experimental Results

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Test</th>
<th>Splits</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jain et al. [62]</td>
<td>–</td>
<td>101</td>
<td>3</td>
<td>0.303 ± 0.00</td>
</tr>
<tr>
<td>Ours</td>
<td>–</td>
<td>101</td>
<td>3</td>
<td><strong>0.328 ± 0.00</strong></td>
</tr>
<tr>
<td>Kodirov et al. [75]</td>
<td>51</td>
<td>50</td>
<td>10</td>
<td>0.140 ± 0.02</td>
</tr>
<tr>
<td>Liu et al. [91]</td>
<td>51</td>
<td>50</td>
<td>5</td>
<td>0.149 ± 0.01</td>
</tr>
<tr>
<td>Xu et al. [196]</td>
<td>51</td>
<td>50</td>
<td>30</td>
<td>0.186 ± 0.02</td>
</tr>
<tr>
<td>Xu et al. [198]</td>
<td>51</td>
<td>50</td>
<td>50</td>
<td>0.222 ± 0.03</td>
</tr>
<tr>
<td>Xu et al. [197]</td>
<td>51</td>
<td>50</td>
<td>50</td>
<td>0.229 ± 0.03</td>
</tr>
<tr>
<td>Li et al. [86]</td>
<td>51</td>
<td>50</td>
<td>30</td>
<td>0.268 ± 0.04</td>
</tr>
<tr>
<td>Ours</td>
<td>–</td>
<td>50</td>
<td>10</td>
<td><strong>0.404 ± 0.01</strong></td>
</tr>
<tr>
<td>Kodirov et al. [75]</td>
<td>81</td>
<td>20</td>
<td>10</td>
<td>0.225 ± 0.04</td>
</tr>
<tr>
<td>Gan et al. [47]</td>
<td>81</td>
<td>20</td>
<td>10</td>
<td>0.311 ± 0.01</td>
</tr>
<tr>
<td>Ours</td>
<td>–</td>
<td>20</td>
<td>10</td>
<td><strong>0.512 ± 0.05</strong></td>
</tr>
</tbody>
</table>

Table 12: Comparison to state-of-the-art for zero-shot action classification on UCF101. For all protocols and test splits we outperform the state-of-the-art, even without us needing any training videos for action transfer.

We conclude from this experiment that our embedding is capable of providing spatio-temporal action retrieval results for arbitrarily specified objects, spatial relations, and object sizes.

7.6.4 Comparison to state-of-the-art

For the fourth experiment, we perform a comparative evaluation of our approach to the state-of-the-art in zero-shot action classification and localization. For localization, we also compare our results to supervised approaches, to highlight the effectiveness of our approach.

**Action classification.** In Table 12, we provide the zero-shot classification results on the UCF-101 dataset, which provides the most comparisons to related zero-shot approaches. Many different data splits and evaluation setups have been proposed, making a direct comparison difficult. We have therefore applied our approach to the three most common types of zero-shot setups, namely using the standard supervised test splits, using 50 randomly selected actions for testing, and using 20 selected actions for testing.

In Table 12, we first compare our approach to Jain et al. [62], who like us do not require training videos. With an accuracy of 0.328 we outperform their approach (0.303). We also compare to approaches that require training videos for their zero-shot transfer, using author suggested splits. For the (random) 51/50 splits for training and testing, we obtain an accuracy of 0.404. Outperform the next best zero-shot approach (0.268) considerably. We like to stress that all other approaches in this regime use the videos from the training split to guide their zero-shot transfer, while we ignore these videos. When using 20 actions for testing, the difference to other zero-shot approaches increases from 0.255 [75] and 0.311 [47] to 0.512. The lower the number of actions compared to the number of objects in our embedding, the more beneficial for our approach.
In Table 13, we provide the localization results on the UCF Sports, Hollywood2Tubes, and J-HMDB datasets. We first compare our result to Jain et al. [62] on UCF Sports in Table 13a, which is the only zero-shot action localization work in the literature we are aware of. Across all overlap thresholds, we clearly outperform their approach. At the challenging overlap threshold of 0.5, we obtain an AUC score of 0.311, compared to 0.071 for Jain et al. [62]; a considerable improvement.

Given the lack of comparison for zero-shot localization, we also compare our approach to several supervised localization approaches on UCF Sports (Table 13a) and Hollywood2Tubes (Table 13b). We observe that we can achieve results competitive to supervised approaches [22, 61, 166], especially at high overlaps. Naturally, the state-of-the-art supervised approach [49] performs better, but requires thousands of hard to obtain video tube annotations for training. Our achieved performance indicates the effectiveness.

### Table 13:

**Comparison to state-of-the-art** for zero-shot action localization on (a) UCF Sports, (b) Hollywood2Tubes, and (c) J-HMDB. The only other zero-shot action localization approach is [62], which we outperform considerably. We also compare with several supervised alternatives. We are competitive, especially at high overlaps thresholds.

<table>
<thead>
<tr>
<th></th>
<th>UCF Sports</th>
<th>Hollywood2Tubes</th>
<th>J-HMDB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.1 0.2 0.3 0.4 0.5</td>
<td>0.1 0.2 0.3 0.4 0.5</td>
<td>0.1 0.2 0.3 0.4 0.5</td>
</tr>
<tr>
<td><strong>Supervised</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gkioxari et al. [49]</td>
<td>0.560 0.560 0.560 0.520 0.495</td>
<td>0.345 0.240 0.154 0.092 0.048</td>
<td>0.346 0.333 0.305 0.268 0.230</td>
</tr>
<tr>
<td>Jain et al. [61]</td>
<td>0.550 0.525 0.490 0.370 0.270</td>
<td>0.121 0.051 0.020 0.007 0.001</td>
<td></td>
</tr>
<tr>
<td>Tian et al. [166]</td>
<td>0.455 0.425 0.315 0.265 0.240</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cinbis et al. [22]</td>
<td>0.292 0.169 0.128 0.102 0.049</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Zero-shot</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jain et al. [62]</td>
<td>0.288 0.232 0.162 0.099 0.072</td>
<td>0.210 0.138 0.086 0.047 0.020</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td><strong>0.435</strong> 0.393 0.371 0.357 0.311</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Action localization.** In Table 13, we provide the localization results on the UCF Sports, Hollywood2Tubes, and J-HMDB datasets. We first compare our result to Jain et al. [62] on UCF Sports in Table 13a, which is the only zero-shot action localization work in the literature we are aware of. Across all overlap thresholds, we clearly outperform their approach. At the challenging overlap threshold of 0.5, we obtain an AUC score of 0.311, compared to 0.071 for Jain et al. [62]; a considerable improvement.

Given the lack of comparison for zero-shot localization, we also compare our approach to several supervised localization approaches on UCF Sports (Table 13a) and Hollywood2Tubes (Table 13b). We observe that we can achieve results competitive to supervised approaches [22, 61, 166], especially at high overlaps. Naturally, the state-of-the-art supervised approach [49] performs better, but requires thousands of hard to obtain video tube annotations for training. Our achieved performance indicates the effectiveness.
of our approach, even though no training examples of action videos or bounding boxes are required. Finally, to highlight our performance across multiple datasets, we provide the first zero-shot localization results on J-HMDB in Table 13c.

**Conclusion.** For classification, we outperform other zero-shot approaches across all common evaluation setups. For localization, we outperform the zero-shot localization of [62], while even being competitive to several supervised action localization alternatives.

### 7.7 CONCLUSIONS

We introduce a spatial-aware embedding for localizing and classifying actions without using any action video during training. The embedding captures information from actors, relevant local objects, and their spatial relations. The embedding further profits from contextual awareness by global objects. Experiments show the benefit of our embeddings, resulting in state-of-the-art zero-shot action localization and classification. Finally, we demonstrate our embedding in a new spatio-temporal action retrieval scenario with queries containing object positions and sizes.
CONCLUSIONS

8.1 THESIS SUMMARY

This thesis investigates how objects help to recognize and localize activities in visual content. Specifically, this thesis focuses on the research question: *what do objects tell about the extent of activities in space and time?* We break this research question down into six chapters, starting by studying the spatial extent of activities, followed by the temporal extent and finally the full spatio-temporal extent of activities.

Chapter 2 first investigates activity recognition from a spatial perspective by learning a set of image parts that describe various activities. Different from most existing part-based methods, we argue that parts are naturally shared between categories and should be modeled as such. We motivate our approach with a quantitative and qualitative analysis by backtracking where selected parts come from. Our analysis shows that in addition to the category parts defining the category, the parts coming from the background context and parts from other image categories improve categorization performance. Part selection should not be done separately for each category, but instead be shared and optimized over all categories. To incorporate part sharing between categories, we present an algorithm based on AdaBoost to optimize part sharing and selection, as well as fusion with the global image representation. Our experimental results show the effectiveness of our approach for activities, as well as object and scene categories, further improving over deep convolutional neural networks and alternative part representations. We conclude that the spatial extent of activities goes beyond the activity itself to include its surrounding context and even parts of other activities.

Chapter 3 investigates activity recognition temporally in videos. Where current works encode videos by averaging object scores over all frames, we propose to encode videos using fragments that are discriminatively learned per activity. Our *bag-of-fragments* split a video into semantically coherent fragment proposals. From training video proposals we show how to select the most discriminative fragments for an activity. An encoding of a video is in turn generated by matching and pooling these discriminative fragments to the fragment proposals of the video. The bag-of-fragments forms an effective encoding for activity detection and is able to provide a precise temporally localized event recounting. Furthermore, we show how bag-of-fragments can be extended to deal with irrelevant objects in the activity recounting. As such, we conclude that fragments describing different object preferences matter for activity recognition and recounting in videos.

Chapter 4 investigates the object representations used to help recognize activities. The current standard for object-based activity representation is to learn from the 1,000 classes
defined in the ImageNet Large Scale Visual Recognition Challenge [26, 136]. Here, we investigate how to leverage the complete ImageNet hierarchy for pre-training deep networks. To deal with the problems of over-specific classes and classes with few images, we introduce a bottom-up and top-down approach for reorganization of the ImageNet hierarchy based on all its 21,814 classes and more than 14 million images. Experiments on challenging activity datasets show that video representations derived from the layers of a deep neural network pre-trained with our reorganized hierarchy yield state-of-the-art results, indicating the importance of employing the right set of objects and hierarchical level for representing activities.

Chapter 5 focuses on the full spatio-temporal extent of activities for localization in videos. The state-of-the-art relies on action proposals at test time and selects the best one with a classifier trained on carefully annotated box annotations [61,118,175]. Annotating activity boxes in video is cumbersome, tedious, and error prone. Rather than annotating boxes, we propose to annotate activities in video with points on a sparse subset of frames only. We introduce an overlap measure between action proposals and points and incorporate them all into the objective of a non-convex Multiple Instance Learning optimization. Experimental evaluation on activity localization datasets shows that using point annotations yield results comparable to box annotations while being significantly faster to annotate.

Chapter 6 continues the work of the previous chapter and proposes fully automatic visual cues that replace manual point annotations, dubbed pseudo-annotations. These pseudo-annotations include cues about objects [217] and actors [205], as well as actions [175], motion [61], and center bias [169]. Thorough evaluation on three challenging activity localization datasets shows that we reach results comparable to results with full box and point supervision. We also show that pseudo-annotations can be leveraged during testing to improve weakly- and strongly-supervised localizers. This indicates that auxiliary information about objects, actors, and motion can help to point out when and where certain activities occur in videos, resulting in activity localization using only video labels as activity annotations.

Finally, chapter 7 continues our goal from the last two chapters to reduce the annotation effort for activity localization by replacing it with object information. We aim to both localize and classify human activities in videos without any training example. Where traditional approaches rely on global attribute or object classification scores for their zero-shot knowledge transfer, our main contribution is a spatial-aware object embedding. To arrive at spatial awareness, we build our embedding on top of freely available actor and object detectors. Relevance of objects is determined in a word embedding space and further enforced with estimated spatial preferences. Besides local object awareness, we also embed global object awareness into our embedding to maximize actor and object interaction. Finally, we exploit the object positions and sizes in the spatial-aware embedding to demonstrate a new spatio-temporal action retrieval scenario with composite queries. Action localization and classification experiments on four contemporary action video datasets support our proposal. Apart from state-of-the-art results in the zero-shot localization and classification settings, our spatial-aware embedding is even competitive with recent supervised action localization alternatives. We conclude that objects and their spatial relations to actors provide a strong link for the spatio-temporal extent of activities, making localization without any training examples possible.
8.2 GENERAL CONCLUSIONS

This thesis has shown the potential of using objects to help learn *which* activities are present in videos and also *when* and *where* activities occur. Throughout this thesis, the focus has been on recognition in videos with predominantly a single dominant activity present. As such, a logical next step involves exploiting objects to recognize activities in complex scenes containing various other activities. Furthermore, the activities in this thesis mostly depict scenes from sports, movies, and consumer videos. Generalizing the scope to include activities from *e.g.* surveillance footage and ego-centric videos allows us to investigate the link between objects and activities on a more general level. Ultimately, the goal is to move beyond human-object interaction towards long-term reasoning about high-level activities as they happen in videos. Being able to recognize activities and the interacting objects provides a basis for high-level reasoning by machines.
SAMENVATTING

Objecten voor de Spatio-Temporele Herkenning van Activiteiten in Video’s

Dit proefschrift onderzoekt hoe objecten kunnen helpen om activiteiten te herkennen en localiseren in visuele data. Het proefschrift gaat specifiek in op de onderzoeksvraag: *wat vertellen objecten over de extent van activiteiten in tijd en ruimte?* We vallen deze onderzoeksvraag aan in zes hoofdstukken, beginnend by het studeren van de spatiële extent van activiteiten. Dit wordt gevolgd door het temporele extent van activiteiten en uiteindelijke de volledige spatio-temporele extent van activiteiten.

Hoofdstuk 2 onderzoekt eerst de herkenning van activiteiten van een spatieel perspectief door een collectie van deelplaatjes – locale vierkanten van gehele plaatjes – te leren die verschillende activiteiten beschrijven. Anders dan de meeste bestaande methoden, argumenteren wij dat deelplaatjes van nature gedeeld zijn tussen categorieën en dat ze ook zo gemodelleerd moeten worden. We motiveren deze argumentatie met een kwantitatieve en kwalitatieve analyse door te kijken waar geselecteerde deelplaatjes vandaan komen. Onze analyse laat zien dat deelplaatjes van sowel ieders eigen categorie, als van andere categorieën, en van de achtergrond helpen met de herkenning van activiteiten. Het selecteren van deelplaatjes moet dus niet gedaan worden voor elke categorie apart, maar moeten gedeeld en geoptimaliseerd worden over all categorieën. Om het delen van deelplaatjes op te nemen, presenteren wij een algoritme gebaseerd op AdaBoost om te optimaliseren voor het delen en selecteren van deelplaatjes. Verder optimaliseert ons algoritme ook de opname van globale representaties van plaatjes. Onze experimentele resultaten laten de effectiviteit zien van onze aanpak voor zowel activiteiten als objecten en scenes in het algemeen, met daarbij een verdere verbetering over diepe netwerken en andere methoden die deelplaatjes gebruiken. We concluderen dat de spatiële extent van activiteiten verder gaan dan de activiteit zelf om deelplaatjes van context om de activiteit en zelfs deelplaatjes van andere activiteiten op te nemen.

Hoofdstuk 3 onderzoekt de herkenning van activiteiten in video’s van een temporeel perspectief. Huidige werken representeren video’s door object scores te middelen over all frames, terwijl wij voorstellen om video’s te representeren met gebruik van fragmenten die discriminatief geleerd zijn per categorie. Onze *bag-of-fragments* splitsen een video in semantisch coherente fragment proposals. Op basis van proposals van training video’s laten we zien hoe de meest discriminatieve fragmenten geselecteerd worden voor een activiteit. Een video representatie wordt vervolgens gemaakt door die discriminatieve fragmenten te matchen met de fragment proposals van een nieuwe video. De bag-of-fragments vormen een effectieve representatie voor de herkenning van activiteiten en ze
kunnen ook een precieze semantische omschrijving geven van video's met temporele locaties. Verder laten we zien hoe de bag-of-fragments kunnen worden verbeterd om om te kunnen gaan met irrelevante objecten in de omschrijving. We concluderen daarom dat fragmenten die verschillende objecten beschrijven er toe doen voor de herkenning en omschrijving van activiteiten.

Hoofdstuk 4 onderzoekt de object representaties zelf die worden gebruikt om activiteiten te herkennen. De huidige standaard voor activiteitherkenning op basis van objecten is om te leren van de 1,000 categorieën die gedefinieerd zijn in de ImageNet Large Scale Visual Recognition Challenge [26, 136]. In dit hoofdstuk onderzoeken wij hoe de complete ImageNet hiërarchie gebruikt kan worden om diepe netwerken van te voren te trainen. Om om te kunnen met de problemen van te specifieke object categorieën en met categorieën met weinig voorbeelden, introduceren we een bottom-up en top-down aanpak voor de reorganisatie van de ImageNet hiërarchie met al zijn 21,814 objecten en meer dan 14 miljoen plaatjes. Experimenten op uitdagende activiteiten datasets laten zien dat onze video representaties die gebruik maken van de lagen van de deep netwerken van onze reorganisaties state-of-the-art resultaten gehalen. Dit laat het belang zien van het gebruik van de juiste collectie van objecten en het juiste hiërarchische niveau om activiteiten te representeren.

Hoofdstuk 5 focust zich op de gehele spatio-temporele extent van activiteiten voor localisatie in video’s. De state-of-the-art bouwt op actie proposals tijdens het testen en selecteert de beste proposals met een classifier die getraind is op zorgvuldig geannoteerde boxes [61, 118, 175]. Activiteiten boxes annoteren in video is langzaam, veel werk, en gevoelig voor fouten. In plaats van boxes te annoteren, stellen wij voor om activiteiten te annoteren in video met punten op een klein gedeelte van de video frames. We introduceren een overlap functie tussen actie proposals en punten en we voegen de functie toe in een Multiple Instance Learning optimalisatie. Experimentele evaluatie op activiteiten datasets laten zien dat het gebruik van punt annotaties vergelijkbaren resultaten behaalt met box annotaties, terwijl punten significant sneller zijn om te annoteren.

Hoofdstuk 6 gaat door op het werk van het vorige hoofdstuk en stelt volledig geautomatiseerde visuele clues voor om handmatige punt annotaties te vervangen. We noemen deze automatische cues pseudo-annotaties. Deze pseudo-annotaties bevatten clues over objecten [217] en acteurs [205], en verder over acties [175], beweging [61], en een voorkeur voor het midden van de frame [169]. Evaluatie op drie uitdagende activiteiten datasets laten zien dat we resultaten kunnen halen die vergelijkbaar zijn met resultaten van volledige box en punt supervisie. We laten ook zien dat pseudo-annotaties gebruikt kunnen worden tijdens het testen om elk localisatie model te verbeteren. Dit geeft aan dat externe informatie over objecten, acteurs, en beweging kan helpen om aan te wijzen waar en wanneer bepaalde activiteiten voorkomen in video’s, met een activiteitlen localisatie tot gevolg die alleen video labels nodig heeft als annotaties.

Hoofdstuk 7 gaat door met het doel van de laatste twee hoofdstukken om de noodzaak voor annotaties in activiteiten localisatie terug te nemen door het te vervangen met informatie over objecten. We hebben zowel de classificatie als localisatie van activiteiten tot doel zonder gebruik te maken van trainingsvoorbeelden. Waar traditionele aanpakken bouwen op globale attributen of object classificatie scores om hun kennis over te dragen, is onze grootste contributie een spatial-aware embedding van objecten. Om tot spatial-awareness te komen, bouwen wij onze embedding boven op vrij verkrijgbare acteur
en object detectoren. De relevantie van objecten wordt bepaald in een embedding ruimte van woorden en verder gestuurd met geschatte spatiale preferenties. Naast locale spatiale awareness, passen we ook globale object awareness in in onze embedding om de interactie tussen acteurs en objecten te maximalizeren. Uiteindelijk buiten we de object posities en groottes in de spatial-aware embedding uit om een nieuwe spatio-temporele actie retrieval scenario te demonstreren met uitgebreide queries. Actie localizatie en classificatie experimenten op vier actuele datasets bevestigen ons voorstel. Naast state-of-the-art resultaten in classificatie en localizatie zonden trainingsvoorbeelden, is onze embedding zelfs competitief met recente alternatieven die wel leren van voorbeelden. We concluderen dat objecten en hun spatiale relaties met acteurs een sterke link bevatten voor de spatio-temporele extent van activiteiten, wat het mogelijk maakt om activiteiten te localizeren zonder trainingsvoorbeelden.
ACKNOWLEDGEMENTS

This PhD could not have been completed without the help of many great people around me and I therefore wish to express my gratitude. First, I want to thank my supervisor Cees Snoek for all his help and guidance. Your structured approach to supervision is precisely what I needed. Whenever I had a new idea, new results, or was stuck, you always had the bigger picture in mind and were willing to guide me further. I also see now the lengths you have gone through to include me in all forms of research activities beyond writing papers. Thank you Cees for helping me get to where I am now. I have much more to learn from you and I hope to gather more knowledge in our future collaborations. I also want to thank my promotor Arnold Smeulders for his guidance and for sharing his life philosophy. Your insights provided during Soos talks, presentations, and meetings have taught me a lot about the life of a researcher, the importance of formality, and about cats in Nijmegen.

I also want to thank two people who started on the same research project; Jan van Gemert and Thomas Mensink. Jan, I realize how much time I stole from your busy schedule with our blue couch meetings. For me however, they were ever so important. Thank you for being patient with me and for teaching me to be a better researcher. Now that you have traded Amsterdam for Delft, there is no longer a dynamic duo on the blue couch, but I will do my best to honor your legacy by continuing such meetings. Thomas, thank you for all your insights into the pros and cons of our job and for helping me take the first teaching steps. Your critical mindset has created refreshing discussions that forced me to properly think about all the steps I want to make in research and beyond.

I am grateful to Shih-Fu Chang for welcoming me to his lab at Columbia University during the summer of 2016. The experience was a grand adventure that resulted in many new stories to share. I am happy that you made me feel so welcome in New York, as well as for our collaboration on Chapter 6.

Similar to any other thesis from our group, it could not have been done without the help of Dennis Koelma. It is no secret that Chapter 4 happened due to your notion and efforts to make it work. Thank you for all your help over the years and for letting me take the steering wheel during our trip through America. The formation of all our PhDs is equally dependent on the help of Virginie Mes. Thank you for keeping all of us in check and for always being available to help us. I also want to give special thanks to Morris Franken for the awesome cover rendered for this thesis. I owe you at least multiple beers (read: ice teas) for all your help.

I would like to thank all the members of the committee for taking the time to read my thesis and honouring my defense with their presence.

In case of a tough defense, I have two paranymphs at my side to see me through: Spencer Cappallo and Honza (Jan) Zahálka. Spencer, it is a shame we share an office now,
no longer is it possible to walk into your office to discuss every new idea. Thank you for not being a (loud) snorer during our conference stays and your Louis CK reenactments. To Honza, "Děkuji, že jsi mi ukázal Prahu, město vlka, a za vyděšení mého synovce, když navštívil naší kancelář.". Although not officially a paranymph, I would like to acknowledge Stevan Rudinac here as the honorary paranymph to my and any other PhD defense. Thanks for keeping the spirits up and for always being down to socialize with us in the office.

During the course of the PhD, I have had the pleasure to work and interact with many great colleagues. Thank you Masoud for sharing our kind heart and positive spirit. Silvia, you are one of the most kindhearted people I know, I am sure you will reach whatever bar you set for yourself. To Zhenyang, who will defend on the same day as me, I wish you good luck. Elise, thank you for (reluctantly) tolerating the field of Computer Vision and for sharing exotic Cuba Libres with me. Cheers to all the people from the office with whom I have had the pleasure to work and socialize with; Fares, Lou, Sezer, Svetlana, Agni, Ran, Jörn, Kandan, Stratis, Zeynep, Amir(hossein), Amir, Ivo, Mihir, Hamdi, Gjorgji, William, Devanshu Arya, Tom, Peter, Karen, (mini-)Max, Bas, Rianne, Jakub, Patrick, Sara, Matthias, Yash, Zeno, and Marcel. Also cheers to the people from the closely adjacent labs (the QUVA lab, AMLab, CV lab, and Delta lab) for sharing lunches, drinks, and frisbees.

Ik wil mijn vader, moeder, zus, Fernando, en de rest van mijn familie bedanken voor alle steun en interesse in het onderwerp. Zonder jullie steun had ik het nooit gehaald en dit zou voor mij niets betekenen als ik het niet met jullie kon delen. Verder wil ik ook Sandra, mijn vrienden, en voetbalgenoten bedanken voor de aanmoedigingen en belangstelling over de jaren.
<table>
<thead>
<tr>
<th>Title</th>
<th>Published in</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water Detection through Spatio-Temporal Invariant Descriptors</td>
<td>Computer Vision and Image Understanding, issue 154, 2016.</td>
<td>P. Mettes, R.T. Tan, R.C. Veltkamp</td>
</tr>
<tr>
<td>A bottom-up approach to class-dependent feature selection for material classification</td>
<td>Computer Vision Theory and Applications (VISAPP), 2014.</td>
<td>P. Mettes, R.T. Tan, R.C. Veltkamp</td>
</tr>
<tr>
<td>Title</td>
<td>MediaMill at TRECVID 2014: Searching concepts, objects, instances and events in video</td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Published in</td>
<td>NIST TRECVID workshop 2014.</td>
<td></td>
</tr>
</tbody>
</table>
A.1 ANNOTATION PROTOCOL

Below, we outline how each action is specifically annotated using a bounding box. The protocol is the same for the point annotations, but only the center of the box is annotated, rather than the complete box.

- **AnswerPhone**: A box is drawn around both the head of the person answering the phone and the hand holding the phone (including the phone itself), from the moment the phone is picked up.

- **DriveCar**: A box is drawn around the person in the driver seat, including the upper part of the stear itself. In case of a video clip with a driving car in the distance, rather than a close-up of the people in the car, the whole car is annotated as the driver can hardly be distinguished.

- **Eat**: A single box is drawn around the union of the people who are jointly eating.

- **FightPerson**: A box is drawn around both people fighting for the duration of the fight. If only a single person is visible, no annotation is made. In case of a chaotic brawl with more than two people, a single box is drawn around the union of the fight.

- **GotOutCar**: A box is drawn around the person starting from the moment that the first body parts exists the car until the person is standing complete outside the car, beyond the car door.

- **HandShake**: A box is drawn around the complete arms (the area between the union of the shoulders, ellbows, and hands) of the people shaking hands.

- **HugPerson**: A box is drawn around the heads and upper torso (until the waist, if visible) of both hugging people.

- **Kiss**: A box is drawn around the heads of both kissing people.

- **Run**: A box is drawn around the running person.

- **SitDown**: A box is drawn around the complete person from the moment the person starts moving down until the person is complete seated at rest.
Figure 46.: Annotation aggregations for the point and box annotations on Hollywood2Tubes. The annotations are overall center-oriented, but we do note a bias towards the rule-of-thirds principle, given the higher number of annotations on $\frac{2}{3}$-th the width of the frame.

<table>
<thead>
<tr>
<th></th>
<th>Training set</th>
<th>Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of videos</td>
<td>823</td>
<td>884</td>
</tr>
<tr>
<td>Number of action instances</td>
<td>1,026</td>
<td>1,086</td>
</tr>
<tr>
<td>Numbers of frames evaluated</td>
<td>29,802</td>
<td>31,295</td>
</tr>
<tr>
<td>Number of annotations</td>
<td>16,411</td>
<td>15,835</td>
</tr>
</tbody>
</table>

Table 14.: Annotation statistics for Hollywood2Tubes. The large difference between the number of frames evaluated and the number of annotations is because the actions in Hollywood2 are not trimmed.

- **SitUp:** A box is drawn around the complete person from the moment the person starts to move upwards from a laid down position until the person no longer moves upwards.

- **StandUp:** Vice versa to SitDown.

## A.2 Annotation Statistics

In Figure 46, we show the aggregated point annotations (training set) and box annotations (test set). The aggregation shows that the localization is center oriented. The heatmap for the box annotations do show the rule-of-thirds principle, given the the higher number of annotations on $\frac{2}{3}$-th the width of the frame.

In Table 14, we show a number of statistics on the annotations performed on the dataset.
A.3 ANNOTATION EXAMPLES

In Figure 47 we show an example frame of each of the 12 actions, showing the diversity and complexity of the videos for action localization.

Figure 47.: Example box annotations of test videos for Hollywood2Tubes.
BIBLIOGRAPHY


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


