UNIVERSITÀ
DEGLI STUDI
FIRENZE

UNIVERSITÀ DEGLI STUDI DI FIRENZE
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE (DINFO)
CORSO DI DOTTORATO IN INGEGNERIA DELL'INFORMAZIONE
CURRICULUM: INFORMATICA (SSD ING-INF/05)

OBJECT AND ACTION ANNOTATION IN VISUAL MEDIA BEYOND CATEGORIES

Candidate
Federico Becattini

Supervisors
Prof. Alberto Del Bimbo

Dr. Lorenzo Seidenari

Prof. Marco Bertini

PhD Coordinator
Prof. Luigi Chisci

CICLO XXX, 2014-2017

Università degli Studi di Firenze, Dipartimento di Ingegneria dell'Informazione (DINFO).

A Jacopo

Acknowledgments

First of all, I would like to thank Professor Alberto Del Bimbo for giving me the chance to work for three years under his supervision. He actually did more than that: he fed my interests and passions in the wonderful subject of Computer Vision, he found me a spot in a great lab among a bunch of amazing researchers and he trusted me enough to put me in front of a class full of students (I'm sure I am the one who learned the most in those occasions). Overall, the PhD has been an amazing experience that made me grow both from a personal and professional point of view, and it probably wouldn't have been possible without him.

Straight after, I need to deeply thank Lorenzo Seidenari. For one reason or another I have been working under his guidance since 2011, when I was still a bachelor student. Almost everything that I know regarding my job as an engineer, a computer scientist and a researcher can be traced back to him in some measure. His ideas and advices have been the foundation of my PhD and I'm grateful to have had him as a supervisor.

I would also like to thank everyone who actively worked with me during these past three years, starting with Lamberto Ballan, with whom I crossed paths only in the final stages of the PhD, yet who left a huge footprint on my work and my way of thinking. In the same way Tiberio Uricchio has always provided precious insights both on our research and on our personal lives, proving to be at the same time a great colleague and friend. With Claudio Baecchi I had the pleasure to rush through a series of "brave" experiments and a lot of others crazy "new ideas". He is the main source of general computer science knowledge in town, and the first place my compass turns to when I manage to mess things up. Which is quite often. Under our guidance, Giovanni Cuffaro, besides being an old friend, proved to be the student who gave me the greatest satisfactions up till now. Andrea Ferracani, Giuseppe Becchi and Simone Ercoli instead shared with me the more applied side of my PhD, visiting museums, companies, campuses and also restaurants from time to time. Daniele Pezzatini and Lea Landucci also took part in some of those visits, during the development of *Imaging900*. Simone and Andrea were also involved with me in the Multimedia Design and Production course and I thank them for their help and support. Thank you all, it has been a pleasure working with you.

The next thought goes to Leonardo Galteri and Claudio Ferrari, the other two "PhD noobs". We started this journey together, we went to lectures and courses together and went through a lot of amazing experiences side by side. Many of the best memories that I have from the last three years involve you two, it wouldn't have been the same if we weren't in this together.

Who's left from the lab to thank? First of all Francesco Turchini, the guy who sat at my right side for three years with all the pros and cons that this implies. We have laughed and worked together, we constantly made jokes at each other and we probably hated each other for at least ten minutes every month. But in the end we grew up to be great friends and I'm glad about it. You are probably the first one in the lab I bonded with when I first started the PhD, three years ago. Andrea Salvi, the last of a long list of people who sat at my left side. But, more than that, a friend who started learning this job from the very foundations with me, almost ten years ago. The first time I stepped into MICC was with you and we shared a countless number of days in there together. I clearly remember this unexpected wave of joy when you told me that you decided to come and work there with us. Similar things can be said for Vincenzo Varano, the third brother in arms that went through all University with us. It always makes me smile when I think about our time studying together and now working side by side, the three of us all sitting in the same room at the same table. Enrico Bondi and Dario Di Fina, the former lab chiefs before me. Your absence in the lab is almost tangible, and not because there will never be a better lab chief than Dario, but because we miss both of you a lot. Federico Bartoli, my Maker companion. Being squashed on that train twice a day, talking all day to the weirdest audience and those fantastic dinners in Trastevere made those days in Rome memorable. Matteo Bruni and Riccardo Del Chiaro, you have just joined the lab but you already proved to be great additions to our group. In particular thanks to Matteo we will all be "Never hungry again". I would also like to thank all the remaining great "seniors" in the lab: Marco Bertini, Andrew Bagdanov, Pietro Pala, Stefano Berretti, Federico Pernici and Giuseppe Lisanti. You are all a reassuring presence we can count on and look up to as role models for our job. Thanks also to Maxime Devanne, Suman Ghosh, Mahshid Majd and Dena Bazazian who enriched our lab with their presence during their visiting period. In particular thanks to Mahshid who helped opening my mind towards different cultures and different ways of seeing the world, teaching me not to take everything for granted. Your enthusiasm and persistence to convince me to visit your home town are remarkable. Who knows, maybe I'll be able to come there eventually. And thanks to Dena and her contagious liveliness, which turned our lab into a much more pleasant environment. Teaching you to talk as a true Florentine and having you around for the three months you stayed here, made it much easier to deal with the final stressful rush of this PhD and turned you into a priceless and unique friend.

Stepping outside the lab, I would also like to mention all the people that I had the pleasure to meet thanks to the PhD. Being able to attend to interesting conferences

and amazing summer schools has been a chance to connect with a huge network of brilliant people scattered all around the world. Thanks to Macarena (Spain), the only one I was able to meet every year across three different cities. I hope we'll keep this trend up in the future and enjoy your great company again. Thanks to all the other ICVSS 2015 and 2016 guys and especially to Estefania (Spain), Lubos (Slovakia), Agnese (Italy), Fabio (Bari), Dan (Hong Kong), Pascal (Netherlands), Felipe (Brasil), Laura (Spain), Eric (Sweeden), Simon (Sweeden), Giulia (Italy), Alessia (Italy), Valentina (Italy) and the drunk guy at the party (USA). Thanks to all the people I spent an amazing time with in Bucharest: Miguel Angel (Spain), Jordi (Spain), Gabriel (Brasil), Souad (Tunisia), Alexandros (Greece), Ionut (Romania), Lukas (Austria), Roxane (Austria). Thanks to Colin (USA), who shared a lunch with me during my brief visit in Amsterdam. And thanks to all the other amazing guys I just had the pleasure to meet in Venice: Marc (Spain), Luis (Spain), Xialei (China), Lu (China), Lichao (China), Aitor (Spain), Pravin (India), Maryam (Iran) and Necati (Turkey). I probably forgot to mention someone, but thank you all. Getting to know every single one of you turned me into a much more complete and open minded person than I was three years ago. Doing a PhD makes you experience the whole spectrum of human emotions (only who went through it will understand the sense of despair of figuring out that you tested on the training set) and you played a big part in most of the positive ones and probably helped to deal with the negatives ones.

A mention is also due to everyone who has been besides me since way before I started this journey. Thanks to Claudia and my whole family, who always encouraged me and never questioned any of my decisions. Thanks to Pelli, Paolo, Paci, Ventu, Guido, Mure and Cicco for sharing the path that led to this point during University. Thanks to Simone, Giacomo, Giada, Sara, Giulia, Giulia and Giulia for being the great friends that you are. Thanks to Duccio, Vanni, Alessia, Giulia, Jessica and Noemi for all the good time we have spent together in the past years, hoping we will share more in the future.

I most certainly missed a lot of people who deserved to be mentioned. It's quite likely that all your names will come to my mind the second this thesis gets printed. And to conclude, thanks to you who are reading this, spending some of your time on it.

Every single word in this thesis is for everyone of you who taught me something, inspired me or looked up at me knowing that I could have made it this far. Never underestimate the power of your encouragement.

Abstract

The constant growth of applications involving artificial intelligence and machine learning is an important cue for an imminent large scale diffusion of intelligent agents in our society, intended both as robotic and as software modules. In particular, the developments in computer vision are now more than ever of primary importance in order to provide a certain degree of awareness to these agents with respect to the environment they act in. In this thesis we reach out to this goal, tackling at first the need to detect objects in images at a fine-grained instance level. Moving then to the video domain we learn to discover unknown entities and to model the behavior of an important subset of those: humans. The first part of this thesis is dedicated to the image domain, for which we propose a taxonomy based technique to speed up an ensemble of instance based Exemplar-SVM classifiers. Exemplar-SVMs have been used in literature to tackle object detection tasks, while transferring at the same time semantic labels to the detections at a linear cost respect to the number of training samples. Our proposed method allows us to employ these classifiers achieving a sub-logarithmic dependence and resulting in speed gains up to 100x for large ensembles. We also demonstrate the application of similar techniques for image analysis in a real case scenario: the development of an Android App for the Museo Novecento in Florence, Italy, which is able to recognize paintings in the museum and transfer their artistic styles to personal photos. Transitioning to videos, we then propose an approach aimed at discovering objects in an unsupervised fashion by exploiting the temporal consistency of a frame-wise object proposal. Almost without relying on the visual content of the frames we are able to generate spatio-temporal tracks that contain generic objects and that can be used as a preliminary step to process a video sequence. Lastly, driven by the intuition that humans should be the focus of attention in video understanding, we introduce the problem of modeling the progress of human actions with a frame-level granularity. Besides knowing when someone is performing an action and where in every frame this person is, we believe that predicting how far an ongoing action has progressed

will provide important benefits for an intelligent agent in order to interact with the surrounding environment and with the human performing the action. To this end we propose ProgressNet, a Recurrent Neural Network based model to jointly predict the spatio-temporal extent of an action and how far it has progressed during its execution. Experiments on the challenging UCF101 and J-HMDB datasets demonstrate the effectiveness of our method.

Contents

Chapter 1

Introduction

Technological advancements have always dictated the pace of how people live their lives, how they interact with each other and how they structure society. Throughout history, scientific findings have become milestones to outline the evolution of mankind, both concerning men in their personal life and the overall social-political context in which they live.

More than ever, the last century has impacted the underlining texture of society, with technologies capable of shortening geographical distances and creating a whole interconnected world, where everyone interacts with an increasing amount of people at an accelerating rhythm. And these changes in society are not just the evolution of a path that has been disentangling throughout the past centuries, it is instead an exponential trend that is destined to change frantically in the near future.

Starting from mechanical vehicles, passing through computers and culminating in the diffusion of the internet, this fast spread of technology has become the key to trigger an even faster development and understanding of technology.

This continuous flow of knowledge and scientific findings is now expected to converge into an era in which the attention shifts, or at least expands, from humans to intelligent agents that interact with men themselves. Tracing back its roots to the first statistical methods in the early 1950s, the next technological milestone is in fact the one of artificial intelligence in everyday life. Thanks to the advancements in machine learning and the ever growing capabilities of computer hardware, we are on the verge of witnessing the diffusion of artificial agents, which can be embodied, among others, by autonomous driving cars, robotic aid systems for disabled people, smart cities or even simple assistants running on a smartphone.

All these pieces of technology, in order to be perfected or to be of any utility for

the final user, are forced to take into account a common trait that resides in actively and autonomously understanding the environment in which they operate. To do so, an intelligent agent can rely on different sources of data, which can generically be identified as sensors. Whether it is a tactile sensor, an ambient sensor, a microphone or a camera, the data will have to be gathered from the environment and analysed through an algorithm in order to provide a prompt and suited response to the user.

The employed algorithms must be capable of determining, in the most accurate way possible, the state of its surroundings, which normally are composed of objects, people and other intelligent agents. Furthermore, this state in which objects and people may be found, is a function of time and is in constant evolution, posing strict time constraints on the actions to be performed. Actions which could themselves be time dependent and therefore might need to continuously adapt to new inputs.

1.1 The objective

By all means the task of realizing an intelligent agent is a complex task to be handled and its facets cover a broad spectrum of research fields. Of particular interest is the capability of the intelligent agent to gather data by actually seeing the environment that surrounds it. In fact, sight could be considered the most important of the human senses for comprehending the context in which an individual, human or artificial, is acting.

With this in mind, in the past decades the field of computer vision has moved huge steps forward towards fully understanding a scene and its content. The vast variability of scenes and possible contents has posed numerous challenges to the computer vision community. In particular the big picture of scene understanding can be sliced in a multiplicity of different semantic layers. At a high level, one could be interested in simply recognizing a setting in a broad sense: am I outdoor or indoor? Is this a beach or an office? Is this a jungle or a soccer field? This is of course a first step to understand the environment but it has a profound importance since context poses a set of implicit constraints about what it is expected to see in it.

Going more in detail, it is important to understand what there is beyond a simple context. Knowing if a particular object is present or not will allow an agent to obtain a better understanding of what is happening in a particular scene. Furthermore, knowing exactly where the objects are, will allow it to interact with them in some way.

The level of detail needed for a computer vision algorithm could be even more

precise, since one could need to know the difference between two similar objects or even to distinguish between different instances of the same kind of object. Fine grained recognition and instance recognition aim at doing this: there is a dog, but is it Fido or Pluto? Is it a Terrier or a Corgi?

This level of understanding and reasoning can be extremely complex, and this is only assuming that the scene is static, i.e. representing it through a single image. Bringing time into the loop adds a whole new layer of complexity, since these objects have to be identified during their evolution in order to be able to interact with them. Differently from before, the data that needs to be processed is a stream of images with a temporal coherence, as in a video.

At first, objects have to be found and recognized in each frame as in the static case, but their identity has to be maintained during the span of their life in the line of sight of the agent (i.e. until the end of the video it is processing). Moreover, the knowledge of how they are evolving will be necessary to interact with them as humans do.

Whereas every type of object in time could go through some kind of transformation, of large interest is the ability to model the behaviour of a particular type of entity: humans. Since most applications are developed as human-centered, understanding human actions assumes a critical importance. Human actions could just be considered as an extension of an object detection task, where a temporal extent is also required in addition to a categorical label and a spatial localization. Yet some applications may need to precisely understand how an action evolves through time, for instance being able to recognize and classify the phases of an action as they develop. This could allow an intelligent agent to efficiently prepare its response or even anticipate what is going to happen in the surrounding environment.

1.2 Contributions

This thesis focuses on algorithms capable of predicting which entities are present in a scene, understand where they are and possibly add some kind of additional semantic information about which state they are in. In particular we are interested in providing a fine-grained labeling to the detected entities, which translates in reasoning about specific object instances instead of generic object classes or give a detailed level of annotation about how they are interacting with the environment.

Overall, we are interested in how the surrounding environment is perceived by a machine, seeking a prompt understanding of entities and how they are modifying or being modified by the environment they live in. First of all it is important to

identify which objects are present and to understand their properties. Only when a detailed level of understanding is obtained, an agent can properly interact with other entities. But how to interact with them is something that can only be learned by observing other humans and how they behave in the environment. Objects and human actions are therefore tightly connected, since the modalities of interaction are defined both by object properties and human behaviors. In particular the entity an agent may interact with could be a human, leading to the need of an even more detailed comprehension of their actions. Therefore, to achieve full understanding of how the environment is structured and how it evolves through time, it is important for an agent to model in a fine-grained way both object appearances and human actions.

We break down this complex problem into simpler modules by separating reasoning about objects and actions. At first we look at objects at an instance level, predicting detailed annotations about them. We deal with objects both in images and in videos, modeling how they evolve in time by exploiting their visual temporal coherence. Videos are also a key component for dealing with humans and their actions, since their comprehension from single images can be an extremely hard task even for a person. We therefore exploit video sequences to provide a detailed annotation of how humans act in the scene while the action is still taking place. In the following we provide a brief overview of these subproblems that we dealt with in this thesis.

The first problem that we address is object detection, i.e. locating objects of interest in images by providing a categorical label and their spatial coordinates within the frame reference system. We consider this task to be the starting point to build more complete algorithms, both for image and video analysis. We adopted the Exemplar-SVM framework [91], a technique that exploits an ensemble of instance classifiers to capture intra-class variations. Exemplar-SVMs are in fact instance based classifiers, which can also be used to transfer semantic informations from known samples onto detections. Our work focuses on making this technique feasible for large ensembles, since its computational footprint has a linear dependence with reference to the number of training samples. By building an exemplar taxonomy and combining it with other efficiency strategies, we are able to obtain up to a two orders of magnitude speed-up in the evaluation of large ensembles.

We exploited similar techniques in a real case scenario, i.e. the development of an Android App for the Museo Novecento in Florence, Italy. Through the App, visitors can recognize some of the paintings in the museum by framing them with the camera. By doing so, they will get additional information about the artworks

and transfer the style of the painting onto a personal photo.

Transitioning towards the video domain, a technique to discover objects in videos in an unsupervised fashion is presented. To find instances of unknown objects we match object proposal boxes [67, 165] across adjacent frames, under the assumption that proposals covering an object will exhibit a spatio-temporal consistency. Through this matching process we are able to create a set of tracks that represent meaningful objects in a video. The whole process does not rely on any kind of prior knowledge about the object that may be present in the video and does not require any form of supervision.

Lastly, motivated by the intuition that understanding humans and their actions is the key to develop effective interaction applications, we introduce the new task of action progress prediction. What we want to achieve is a fine grained comprehension of how an action is evolving during its execution. In particular we predict for each frame in a video how far an action has progressed by indicating the percentage of its completion. At the same time we provide spatio-temporal localization as in classical action detection approaches. To the best of our knowledge we are the first to deal with the concept of action progress in videos.

This thesis is organized as follows. In Chapter 2 an overview of the state of the art on object detection and action recognition is given. In Chapter 3 we outline our technique to speed-up an ensemble of Exemplar-SVMs and tackle an object detection task. In Chapter 4, we describe our experience with the Museo Novecento, where we applied similar image recognition techniques to detect different paintings in the museum. Chapter 5 goes into detail about our unsupervised object discovery method in the video domain, whereas in Chapter **??** we explain our work with action progress prediction, the new task that we introduced for a detailed human action understanding. In Chapter 7 we draw our conclusions, reviewing the most relevant contributions of this work.

Chapter 2

Literature review

This chapter provides a brief overview of related work on object and action recognition. The first part of the chapter focuses on object detection in still images and videos, reviewing both classical methods with handcrafted features and the new techniques that rely on deep learning. The second part deals instead with the problem of localizing and recognizing humans performing specific actions in videos.

This thesis is composed of elements drawn from different fields of research in computer vision and machine learning, focusing on two main areas: object detection, i.e. localizing and recognizing objects in images; and action recognition, i.e. classifying actions in videos, possibly predicting when it takes place (action detection) and where the actor is within each frame (action localization). In the following section an overview of the most used state of the art techniques is given for both topics.

2.1 Object Detection

Object detectors in today's state of the art are based on three different aspects: good features, powerful classifiers and a large amount of data. Traditionally, features have been hand-crafted and engineered to meet design requirements that seemed suitable to tackle the problem at hand and then used to train an appropriate classifier. With the advent of deep learning [81], this pipeline blended into an end-to-end model where features and classifiers are jointly learned directly from raw pixels in order to succeed in specific tasks.

One of the most commonly used hand-crafted feature for object detection is Dalal and Triggs' *Histogram of oriented gradients* (HOG) [21], which characterizes object appearance using edge distribution. These features are usually employed to train models that are evaluated on images in a sliding window fashion at different scales. Since computing (and evaluating) HOG features over image pyramids can be very expensive, Dollár *et al.* [28] have proposed a method to create sparse pyramids approximating the remaining scales with pyramidal classifiers at each level. In [12] the multi-scale sampling problem has been moved from testing time to training time. Fast commonly used classification methods are given by cascade classifiers [141], an extension of the AdaBoost algorithm which uses the intuition according to which a series of week classifiers could outperform a single strong classifier. Cascade architecture found its evolution in soft cascade [14] where the trade-off between speed and accuracy can be weighed exploring ROC surfaces. HOG features are usually exploited in combination with classifiers such as Support Vector Machines (SVM) [18]. A variant of SVM where classifiers are trained with only one positive example, called Exemplar SVM [91], has also gained popularity for the ability to transfer meta-data from training to test samples.

One limitation of holistic sliding window approaches is the lack of robustness to intra-class variations of appearance when aspect ratios are fixed. This problem is encountered when objects of some categories can exhibit a high variability of poses or shapes such as people or animals. Felzenszwalb *et al.* [39] proposed a method to overcome this problem with the Deformable Parts Model (DPM). DPM employs a set of filters with HOG features modelling each category with a root filter and a fixed number of part filters. This kind of classifier is very powerful but at the same time very slow to evaluate, even though fast evaluation methods have been proposed recently using cascade classifiers [40], evaluating templates at all locations simultaneously exploiting the properties of Fast Fourier Transform [29], using vector quantization [120] or combining various strategies to deal with different bottlenecks [121].
In addition to DPM, bag of visual words models ([140], [60]) are also common for object detection. Optimized data structures, such as hash tables, can also be used in object detection. Dean *et al.* [24] exploit local sensitive hashing to replace the dot product operations and effectively detect up to 100.000 classes in less than 20 seconds. Alternatively, Lampert *et al.* [85] have proposed an Efficient Subwindow Search (ESS) using a branch and bound approach.

Another way to speed-up a sliding window detector is to avoid an exhaustive search through window proposal techniques. Object proposals [67] provide a rela-

tively small set of bounding boxes likely to contain salient regions in images, based on some *objectness* measure. In this way only promising areas of the image are tested, reducing considerably the computational burden. In [2] an objectness measure is proposed to select candidate regions that are likely to contain an object and Cheng *et al.* [17] use binarized normed gradients (BING) for efficient objectness estimation. Different proposals, such as EdgeBoxes [165] and Selective Search [139], are commonly used in image related tasks to reduce the number of candidate regions to evaluate.

Proposal methods have also become a key component in modern object detectors based on convolutional neural networks [53]. The R-CNN [52] architecture opened the way to a series of groundbreaking models that exploit a restricted set of proposal regions to accurately localize objects in the scene. R-CNN has been refined in multiple iterations introducing at first the concept of ROI pooling to share part of the computation for each proposal [51] and then replacing the proposal module with a Region Proposal Network (RPN) [115] to learn how to generate candidate boxes. Evolving from this paradigm, CNN-based detectors such as YOLO [112, 113] and SSD [89] have started to look at the whole image to generate predictions based on fixed anchor boxes in the image.

2.2 Action recognition

Human action understanding has been traditionally framed as a classification task. It has been addressed with a plethora of methods using global features, local features and representation learning [1]. Motion based features have played a key role in dealing with action classification tasks: Histogram of Optical Flow (HOF) and Motion Boundary Histogram (MBH) [86] have been largely used, often in combination with powerful encoding schemes such as VLAD [71] and Fisher Vectors [108, 109]. These descriptors have been used in the form of Dense Trajectories [86] to track sampled feature points through frames and follow them according to optical flow. An improved version which separates relevant motion from camera motion has also proved to be successful for action recognition tasks [146]. Methods that adopt Convolutional Neural Networks have also started to emerge, often adapting the model to deal with the temporal component, such as 3D convolutions (C3D) [138], two stream networks [38, 127] and Temporal Segment Networks (TSN) [147].

Recently, helped by the fact that datasets have grown in size and complexity thus providing a larger number of classes and more detailed annotations [69], sev-

eral tasks have emerged aiming at a more precise semantic annotation of videos, namely action detection and action localization. Whereas the goal of action detection is to determine when an action is taking place, action localization also aims at determining where the actor is in every interested frame.

Early methods to tackle the task of temporal action detection have been Gaidon *et al.* [45, 46] proposing an Actom Sequence Model that focuses on atomic actions and Niebles *et al.* [102] who used latent variables to model complex actions. Fernando *et al.* [42] adapted the Bag of Visual Words representation to time in order to deal with action recognition.

More recent approaches mostly rely on deep learning, making a large use of recurrent neural network models: in [32] Long Short Term Memory (LSTM) networks are exploited to gather action proposals from untrimmed videos; in [155] LSTMs are used to deal with multiple inputs and multiple outputs in case of dense action labelling; in [159] a pyramidal descriptor is proposed and combined with recurrent neural networks. Heilbron *et al.* [65] have recently proposed a very fast approach to generate temporal action proposals based on sparse dictionary learning. Yeung *et al.* [156] looked at the problem of temporal action detection as joint action prediction and iterative boundary refinement by training a RNN agent with reinforcement learning. Shou *et al.* [123, 125] instead have recently proposed action localization approaches based on multi-stage CNNs and Convolution-Deconvolution networks.

Frame level action localization instead has been tackled extending state-of-the art object detection approaches [115] to the spatio-temporal domain. A common strategy is to start from object proposals and then perform object detection over RGB and optical flow features using convolutional neural networks [54, 107, 122]. Gkioxari *et al.* generate action proposals by filtering Selective Search boxes with motion saliency, and fuse motion and temporal decision using an SVM [54]. More recent approaches, devised end-to-end tunable architectures integrating region proposal networks in their model [107, 122]. As discussed in [122], most action detection works do not deal with untrimmed sequences and do not generate action tubes. To overcome this limitation, Saha *et al.* [122] propose an energy maximisation algorithm to link detections obtained with their framewise detection pipeline. Another way of exploiting the temporal constraint is to address action detection in videos as a tracking problem, learning action trackers from data [149].

A closely related line of work addresses the specific case of online action detection. In [66], a method based on Structured Output SVM is proposed to perform early detection of video events. To this end, they introduce a score function of class

confidences that has an higher value on partially observed actions. The structure of sequential actions is used in Soral *et al.* [132], in an ego-vision scenario, to predict future actions given the current status. Their purpose is to timely remind the user of actions missing in the sequence.

Going beyond action recognition and detection/localization, more complex kind of tasks have been presented in literature. The very recent direction of predictive vision [144, 148] is for example tries to obtain some kind of prediction of its near future given an observed video. Vondrick *et al.* predict a learned representation and a semantic interpretation [144] while subsequent works predict the entire video frame [94, 145]. All these tasks however do not focus on the progress of an action, they focus on predicting the aftermath of an action based on some preliminary observations, leaving progress prediction an open issue.

Chapter 3

Indexing ensembles of Exemplar-SVMs with rejecting taxonomies

Ensembles of Exemplar-SVMs have been introduced as a framework for object detection but have found a large interest in a wide variety of task. What makes this technique so attractive is the possibility of associating to instance specific classifiers one or more semantic labels that can be transferred at test time. To guarantee its effectiveness though, a large collection of classifiers has to be used. This directly translates in a high computational footprint, which could make the evaluation step prohibitive. To overcome this issue we organize Exemplar-SVMs into a taxonomy, exploiting the joint distribution of Exemplar scores. This permits to index the classifiers at a logarithmic cost. We propose a highly efficient Vector Quantized Rejecting Taxonomy to discard unpromising image regions during evaluation. This allows us to obtain ramarkable speed gains, with an improvement up to more than two orders of magnitude. To verify the robustness of our indexing data structure with reference to a standard Exemplar-SVM ensemble, we experiment with the Pascal VOC 2007 benchmark on the Object Detection competition and on a simple segmentation task. [1]

[1] Parts of this chapter have been published as "Indexing quantized ensembles of exemplar-SVMs with rejecting taxonomies" in *Multimedia Tools and Applications*, vol. in press, 2017. (Special Issue: Content Based Multimedia Indexing) [10] and "Indexing ensembles of exemplar-svms with rejecting

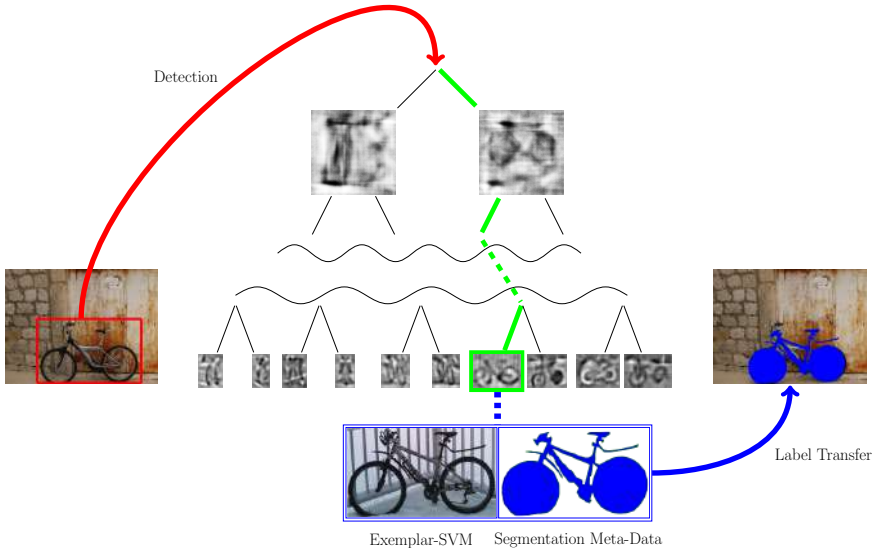

Figure 3.1: Overview of our approach. Vector Quantized Exemplars of each class are stored in a learned taxonomy which is traversed to locate the best matching one enabling detection and label transfer at a logarithmic cost.

3.1 Introduction

The exponential growth of user produced media has made large scale indexing a central task in computer vision and multimedia analysis. This huge amount of data can be leveraged in training complex computer vision algorithms in order to boost performance in tasks such as image classification, object detection or, more broadly, scene understanding.

Indexing algorithms are not just meant to store and retrieve data, but can be a key component in the process of scaling a computer vision technique to data sizes of today's scenarios. Data therefore requires specific indexing structures in order to be handled properly and be accessed as fast as possible. Several approaches have been developed, mostly based on hashing and taxonomy learning techniques. Such structures are built with a process that is guided by similarity in some visual feature space, possibly exploiting partial matching of images to improve robustness. The task at hand, namely image retrieval, classification or object detection, usually

taxonomies", in *Proc. of Content-Based Multimedia Indexing (CBMI)*, Bucharest (Romania), 2016. [9].

determines the choice of structure and similarity function.

Visual taxonomies are often exploited to stress both computational efficiency and to impose a structure on the data. Regarding efficiency, a logarithmic dependency is created on the number of elements that have to be accessed. On the other hand, the topology that this kind of data structure imposes on the data is useful for classification tasks, since samples are organized ranging from coarse to fine as the tree is traversed from the root to a leaf. Both these aspects, which are directly tied to effectiveness and efficiency, depend of how well data is distributed and how it has been clustered together. In fact the branching factor and the overall balance have an impact on both elements. With this in mind, efforts to learn an optimal taxonomy have been done.

Gao and Koller [49] propose to learn a relaxed hierarchy in which a subset of confusing class labels can be ignored in the upper levels of the tree. This method is derived from [153] where a set of binary classifiers is organized in a tree or a DAG structure. A tree data structure where each node groups a set of category labels more and more specific as one gets closer to the leaves has been proposed in [93].

In [13] spectral clustering is used recursively to optimize an overall tree loss, extending the work of Deng *et al.* [27] who jointly learn a hierarchical structure and a set of classifiers used at each node. An optimization problem is designed to maximize efficiency given a constraint on accuracy. A similar approach is used in [56] where the confusion matrix of class similarities is exploited to build a label hierarchy.

Liu *et al.* [88] instead have proposed a probabilistic approach for learning the label tree parameters using maximum likelihood estimation. Similarly, random forests have been exploited to build fast taxonomies for classification [116] and fine-grained categorization [154].

Apart from speed issues, taxonomies are also useful to organize data semantically, since objects are often naturally organizable in hierarchies. In [25] a hierarchy is built to represent composition and exclusion relations among classes. Instead of learning a taxonomy in [82] and [58] the ImageNet [26] hierarchy is exploited to transfer annotations between similar classes. Similarly in [117] WordNet [96] is used to propagate knowledge from known to novel categories. In [99] the WordNet taxonomy is used in combination with spatial information to establish scene configurations. Coarse-to-fine taxonomies have been used to classify plants by Fan *et al.* [36] using a hierarchical multi-task structural learning algorithm.

In this chapter we propose to build a taxonomy through an object based indexing strategy, which adopts a metric learning approach to improve with respect to nearest

neighbor methods.

Our method, instead of indexing image patches, organizes a set of instance specific classifiers into a taxonomy, exploiting as a key component the Exemplar-SVM framework (ESVM) [91]. Exemplar-SVM learns a single classifier for each available training object, meaning that the positive set will contain a single point, opposite to a wide collection of negative patches. ESVM is considered a semi-parametric approach, where object similarity metrics are trained discriminatively. Highly specific templates are learned in this process improving the quality of nearest neighbours. What makes the Exemplar framework even more appealing is the possibility of maintaining the properties of a nearest neighbour technique, namely the ability of establishing correspondences between train and test samples.

All the annotation available for training instances, such as segmentation masks, 3D information or user tags can be transferred to unlabelled data thanks to this direct correspondence that can only be established with non-parametric approaches. This property has found strong interest in the computer vision community in a broad range of tasks, thanks to being simple and intuitive.

The main drawback of Exemplar-SVMs, which limits the practical use of the method, is related to computational efficiency since thousands of classifiers have to be evaluated. This both affects training and testing time complexity. Training is a less relevant issue since it is usually performed once and for all. Moreover it has been shown that given enough background patches linear classifiers can be trained efficiently [59]. In this work we present a novel method to learn a taxonomy on exemplars, that can be exploited as an indexing structure. Furthermore we also show how taxonomy nodes can be processed using Vector Quantization obtaining even higher performance.

The goal is to introduce a logarithmic factor instead of a linear one regarding the number of classifiers that need to be evaluated at test time. To further reduce the amount of classifier evaluations we learn node specific thresholds in order to reject patches that are unlikely to be classified as positive by the subsequent nodes. The choice of exploiting a taxonomy is also justified by the fact that it allows us to cluster similar Exemplars, adding implicit information about visual interclass similarity, which is useful for tasks as detection or fine grained classification.

We quantize clusters of Exemplar-SVMs and treat them as separate and highly efficient classifiers. This approach differs from Fast Template Vector Quantization (FTVQ) [120] since we quantize cluster centroids instead of proper ESVM classifiers. This permits to evaluate the ensemble in the quantized domain, using these centroids to guide the taxonomy traversal with a reduced computational footprint.

As for [97], which introduces Context Forests, our method is significantly different since we are able to speed-up the evaluation with a taxonomy using the whole ensemble, whereas they use forests to obtain a retrieval set of relevant exemplars. Moreover they base their forests on global image features while we directly index the Exemplar classifiers.

We find that exploiting a combination of our speed-up strategies we are able to obtain an improvement of a couple of orders of magnitude in speed. An in depth qualitative and quantitative evaluation of our method is reported, showing the impact of the indexing data structure on an object detection task and a segmentation task.

The chapter is organized as follows. After providing background theory notions about Exemplar-SVM ensembles in Section 3.2, the proposed method is explained in Section 3.3 and Section 3.4, focusing on the learning and evaluation parts. Experimental results are discussed in Section 3.5 and conclusions are drawn in Section 3.6.

3.2 Exemplar-SVM ensembles

Recently, Exemplar-SVMs [91] have been proposed to perform label transfer among objects in images. Labels can be tags or categorical variables that identify the object (as in an object detection task), or more structured meta-data such as segmentation masks or 3D models. Ensembles of Exemplar-SVMs leverage the intuition according to which a pool of simple classifiers, one for each training sample, can outperform a single and complex one. Moreover, behind this choice lays the desire to be able to exploit in object detection tasks the explicit correspondence typical of a nearest-neighbor method inside a discriminative learning framework, such as Support Vector Machines.

An optimization problem is defined for each exemplar $\mathbf{x}_E$, separating it from the negative windows by learning a weight vector $\mathbf{w}_E$ and a bias b_E which identify the optimal separating hyperplane $\mathbf{w}_E^T\mathbf{x} + b_E = 0$. The optimization problem has the following convex objective function

$$\Omega_E(\mathbf{w}, b) = \|\mathbf{w}\|^2 + C_1 h(\mathbf{w}^T\mathbf{x}_E + b) + C_2 \sum_{\mathbf{x}\in\mathcal{N}_E} (-\mathbf{w}^T\mathbf{x} - b) \qquad (3.1)$$

where h is the hinge loss function $h(x) = \max(0, 1 - x)$ and C_1 and C_2 are set separately to moderate the unbalance in the data set. We follow the approach of Malisiewicz *et al.* [91] which report $C_1 = 0.5$ and $C_2 = .01$.

New formulations of the framework have also been proposed. A joint calibration algorithm for optimizing the ensemble in its entirety is used in [98] to learn specific per-exemplar thresholds; recursive E-SVM is defined by [161], where exemplars are used as visual features encoders and in [79] three different viewpoints for interpreting E-SVM are proposed. All this interest towards Exemplar-SVMs is justified by the wide range of possible label transfer applications that can be paired with object detection: segmentation [137], 3D model and viewpoint estimation [5], part level regularization [6], GPS transfer [57], scene classification [130] amongst others.

The evaluation phase of the ESVM framework requires that each classifier is independently tested in a sliding window fashion, generating different detections according to the classifier scores. To perform the step faster one must either reduce the amount of windows to be evaluated or the amount of exemplars. In the following we show how to deal with both issues. First we show how to learn a taxonomy of exemplars per class in order to reduce the number of classifiers. Then we introduce different strategies for reducing the number of test windows such as a rejecting taxonomies and flipped Exemplars. Moreover the use of Vector Quantization tackles the overall speed of the method.

3.3 Learning an ESVM taxonomy

We propose to build a highly balanced tree using spectral clustering hierarchically. Different clustering techniques for learning the taxonomy (such as hierarchical k-means and agglomerative clustering with different cluster aggregation criteria) have been tested in a set of preliminary experiments. All but the divisive spectral clustering resulted in poor taxonomies with unbalanced trees which are detrimental to performance; moreover, as also shown in [70, 130], methods based on Euclidean metrics do not perform well in high-dimensional spaces, due to the well known curse of dimensionality. This is due to the fact that spectral clustering can be seen as a relaxation of a graph partitioning problem [101, 133, 142], which directly optimizes cluster balance. As a quantitative confirmation of this good behavior we perform an experiment to measure the depth and balance of our trees. We use as baseline K-means and compare it to our approach based on spectral clustering.

Table 3.1 reports a comparative analysis between spectral clustering and K-means, highlighting the benefits of the former in creating a balanced tree. We show statistics for different trees, created using the 20 classes of the Pascal VOC 2007 dataset [33]. Balanced trees have a depth equal to the $log_2(X)$, where X is the

										
log_2(Num)	8.26	8.46	8.92	8.18	8.98	7.84	10.29	8.55	9.64	8.02
Depth SC	13	13	13	12	13	12	14	12	14	11
Depth KM	25	37	34	39	47	32	46	31	45	27
Balance SC	0.76	0.77	0.75	0.77	0.76	0.78	0.76	0.76	0.76	0.77
Balance KM	0.49	0.49	0.45	0.42	0.47	0.47	0.47	0.46	0.46	0.46
										
log_2(Num)	7.75	8.99	8.5	8.41	12.2	9.01	8.01	7.95	8.21	8.34
Depth SC	11	13	13	12	17	13	12	13	13	12
Depth KM	37	30	39	28	59	43	31	31	27	36
Balance SC	0.74	0.75	0.74	0.76	0.76	0.76	0.77	0.73	0.77	0.76
Balance KM	0.43	0.47	0.47	0.51	0.47	0.44	0.48	0.49	0.5	0.47

Table 3.1: **Spectral Clustering (SC) vs. K-means Clustering (KM)**. Statistics on the trees created for each class of the Pascal VOC 2007 dataset [33] are shown. `Num` is the number of Exemplars for each class and its binary logarithm provides a lower bound on the tree depth (perfectly balanced tree). `Depth SC` and `Depth KM` are the depths obtained with Spectral Clustering and K-means, respectively (lower is better). `Balance` indicates the average ratio of cluster sizes at each split in the tree (higher is better, perfect balance is obtained with $\texttt{Balance} = 1$).

data stored in the tree. Therefore comparing tree depth with data cardinality is a good cue of tree balancing. Moreover tree depth also affects efficiency, since in the worst case the amount of comparison can not be larger than the tree depth. Spectral clustering highly outperforms K-means, building trees with depths close to the optimal value. To stress this we also report a balance value, calculated as the average ratio of cluster cardinalities at each split in the tree

$$\texttt{Balance}(N) = \frac{1}{|N|} \sum_{i \in N} \frac{\min(|L_i|, |R_i|)}{\max(L_i|, |R_i)}, \tag{3.2}$$

where L_i and R_i are the sets of exemplar obtained splitting node i. A perfectly balanced tree would have `Balance`=1, which is obtained when all node splits have the same cardinality. Considering these results we adopted Spectral Clustering for building our taxonomies.

Spectral clustering is a technique that reformulates clustering as a graph partitioning problem, where connected components of the graph are associated with different clusters. In spectral clustering the attention is focused on a tool called graph Laplacian matrix, which establishes the similarity of nearby vertices. In literature various formulations of the Laplacian matrix have been proposed. In this

work we follow the approach that refers to the normalized version of [101]:

$$\mathbf{L}_n = \mathbf{D}^{-1/2}\mathbf{S}\mathbf{D}^{-1/2} \tag{3.3}$$

where $\mathbf{S}$ is the affinity matrix and $\mathbf{D}$ is a diagonal matrix where each element $D_{ii} = \sum_{j=1}^{N} S_{ij}$.

We first define the matrix $\mathbf{A}$ that captures the likelihood of two exemplars $\mathbf{w}_i$ and $\mathbf{w}_j$ firing together on the same sample. This matrix represents the compatibility of exemplars. So given the matrix obtained by concatenating all the raw features $\mathbf{H} = [\mathbf{h_1} \dots \mathbf{h_N}]$ and the matrix of the respective learnt exemplar hyperplanes $\mathbf{W} = [\mathbf{w_1} \dots \mathbf{w_N}]$ our affinity matrix is defined as:

$$\mathbf{A} = \mathbf{W}^T\mathbf{H}. \tag{3.4}$$

Therefore element A_{ij} represents the score of exemplar $\mathbf{w}_i$ on the feature $\mathbf{h}_j$ on which exemplar $\mathbf{w}_j$ has been trained and vice versa.

Since the matrix $\mathbf{A}$ is not guaranteed to be symmetric we apply the same strategy as in [13] and define

$$\mathbf{S} = \frac{1}{2}\left(\mathbf{A}^T + \mathbf{A}\right) \tag{3.5}$$

that is symmetric.

Recursively applying spectral clustering we create our taxonomy as shown in Algorithm 1. Note that for each split we set the representative of each node as

$$\hat{\mathbf{w}}_N = \frac{1}{|\mathcal{W}_N|}\sum_{i \in \mathcal{W}_N} \mathbf{w}_i. \tag{3.6}$$

An example of the representatives for the first split of the bicycle and bus classes is shown in Figure 3.2, highlighting how the dominant views of the objects, frontal and sideways, are captured. Even though the proposed approach is feature independent in our experiments we employed Histogram of Oriented Gradients features (HOG) [21], as in the original Exemplar-SVM formulation. The representations of the centroids in Figure 3.2 are done showing the HOG glyphs and the inverse Hoggle representation of [143].

3.4 Accelerated ESVM evaluation

An ensemble of exemplars can be easily associated with a set $\mathcal{W}$ of exemplar hyperplanes $\mathbf{w}_i$ of the same class. Each image window feature vector $\mathbf{h}$ can be evaluated

FUNCTION splitSet($\mathcal{W}$)
Data: $\mathcal{W} = \{\mathbf{w}_i \dots \mathbf{w}_M\}; S$
Result: subsets $\mathcal{W}_L, \mathcal{W}_R : \mathcal{W}_L \cap \mathcal{W}_R = \emptyset$;
$\mathcal{W} = \mathcal{W}_L \cup \mathcal{W}_R$;
subset representatives: $\hat{\mathbf{w}}_L$, $\hat{\mathbf{w}}_R$
if $|\mathcal{W}| > 1$ **then**
  $[\mathcal{W}_\mathcal{L}, \mathcal{W}_\mathcal{R}] \leftarrow$ `spectralClustering`$(\mathbf{S}, 2)$;
  $\hat{\mathbf{w}}_\mathbf{L} \leftarrow \frac{1}{|\mathcal{W}_L|} \sum_{i \in \mathcal{W}_L} \mathbf{w}_i$;
  $\hat{\mathbf{w}}_\mathbf{R} \leftarrow \frac{1}{|\mathcal{W}_R|} \sum_{i \in \mathcal{W}_R} \mathbf{w}_i$;
  splitSet($\mathcal{W}_L$);
  splitSet($\mathcal{W}_R$);
else
  leaf node reached;
end

Algorithm 1: Taxonomy Learning. We recursively apply spectral clustering obtaining a binary tree and keeping as representative for node n the mean hyperplane $\hat{\mathbf{w}}_n$.

selecting the best exemplar using:

$$\arg\max_{i \in \mathcal{W}} \mathbf{w}_i^T \mathbf{h} \tag{3.7}$$

In order to do this each exemplar has to be tested against every test window. Thanks to our learned hierarchy we can reduce this operation to a tree traversal. The computation is performed by iteratively selecting from the current node, the child with the highest scoring representative:

$$\texttt{next\_node} = \arg\max_{i \in \{L,R\}} \hat{\mathbf{w}}_i^T \mathbf{h} \tag{3.8}$$

where L and R are the left and right children of the current node, respectively. The scores

$$\hat{\mathbf{w}}_i^T \mathbf{h} = \frac{1}{|\mathcal{W}_N|} \sum_{i \in \mathcal{W}_N} \mathbf{w}_i^T \mathbf{h} \tag{3.9}$$

represent a lower bound on the score obtained by the best exemplar present in each sub-tree therefore we greedily pursue the path that maximizes this bound. This of course does not guarantee to select the leaf with the actual maximum. A schematic pipeline of our system is shown in Figure 3.1.

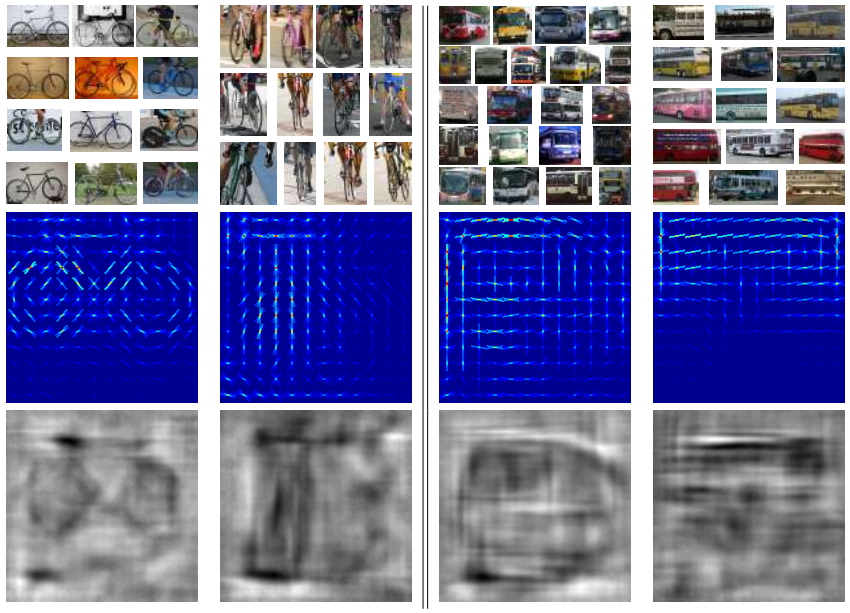

Figure 3.2: Visualization of samples from the first two splits for the classes `bicycle` and `bus` of Pascal VOC 2007 [34] (top). From the HOG (center) and inverted HOG [143] (bottom) representations of their centroids can be clearly seen how the exemplars are indexed based on their viewpoint, clustering frontal and lateral views of the objects. Better viewed on computer.

3.4.1 Flipped exemplars

To improve the efficiency of the taxonomy, we also propose a small variant adding flipped copies of the exemplars to the ensemble. In fact a common trick to enhance detection accuracy, which is also used in the Exemplar-SVM formulation, is to evaluate both test images and their horizontally flipped copies. In this way the classifiers become more expressive, gaining a certain degree of invariance to viewpoint. This, in the case of a sliding window based detector, comes at the cost of doubling the number of windows that have to be evaluated.

With this in mind we evaluate windows only from the unflipped image and instead we double the number the classifiers, adding flipped exemplars. In the case of the standard ensemble there is no substantial difference between testing flipped windows or evaluating flipped models, but thanks to the logarithmic factor introduced by the tree, we are able to halve the number of windows without almost

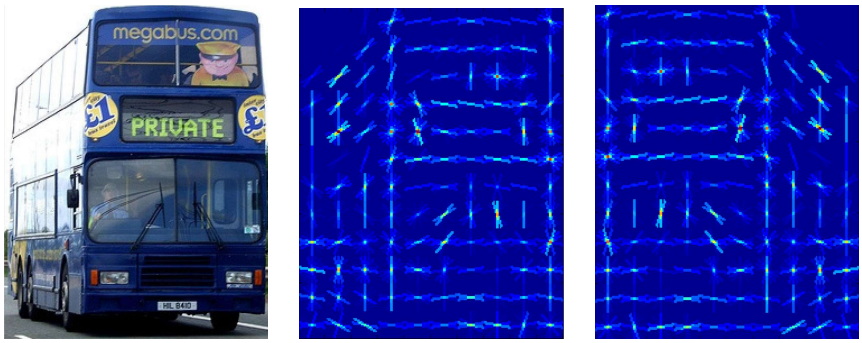

Figure 3.3: **Horizontal flipped HOG**: From the original image (left) an exemplar HOG template is learnt (center) and then the flipped copy is obtained reorganizing the bins of the histogram (right).

any additional cost. In fact if the tree is well balanced, doubling the number of exemplars only increases the depth of the tree by one.

To obtain the flipped version of each classifier we simply swap its components according to the orientations of the gradients in the correspondent HOG feature vector without any additional training. Figure 3.3 shows an example of flipped HOG template.

3.4.2 Early rejection

For each query image tens of thousands of windows have to be evaluated, making the evaluation phase very expensive. Inspired by the similarities with cascade classifiers by Bourdev and Brandt [14], we inserted in our framework a rejecting strategy by learning early rejection thresholds. The aim is to discard most of the windows at the higher levels of the tree if they do not look promising. This is done learning a threshold on the output score of each node, which makes the evaluation faster by drastically reducing the number of comparisons for each image. As shown in Figure 3.4, after a few nodes most of the background windows are removed while windows that reach the leaves are densely clustered around the object to be detected.

We introduce an approach which is close to the philosophy of soft-cascades proposed in [14] to learn the threshold values. Soft-cascades are based on the usage of a rejection distribution vector $\mathbf{v} = (v_1, ...v_T)$ where $v_t \geq 0$ is the minimum fraction of objects that we are allowed to miss at the t-th stage of the cascade. In

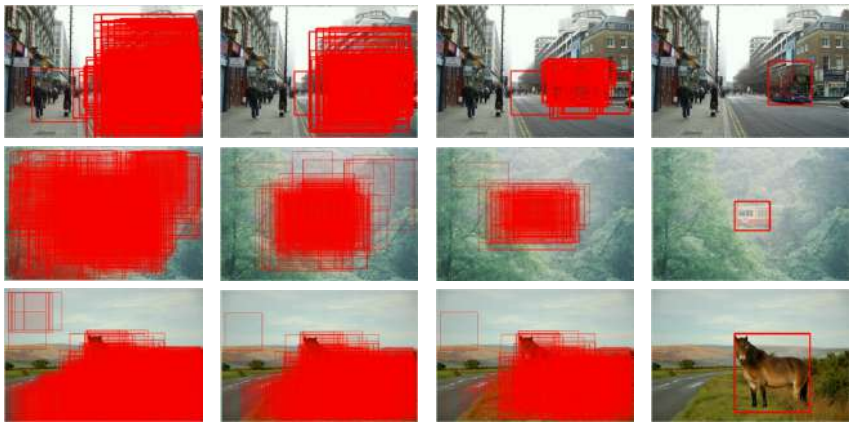

Figure 3.4: Windows retained during tree traversal using rejection thresholds. The amount of windows decreases significantly from the first tree node (leftmost) to the last (rightmost).

our setting, instead of a cascade with T classifiers we have a tree with M leaves, i.e. $2M - 1$ classifiers. The aim is to learn a rejection threshold for each node and to do so we group together nodes at the same depth, in order to establish the rejection distribution vector. Each path down the tree is treated as a soft-cascade. For a tree of depth D we employ a rejection distribution vector $\mathbf{v}$ with D values, one for each tree level, defined as

$$v_d = \exp \frac{1}{2} \left(\frac{d}{D} - 1 \right). \tag{3.10}$$

In our implementation this function defines the percentage of windows we want to keep at each level of our tree and has the property to saturate towards 1 descending towards a leaf. With this strategy we reject most of the least promising windows in the first levels, increasing the amount of kept windows at each stage.

Thresholds t_n for each node n are learned by evaluating approximately 2M windows drawn randomly from a validation set. Considering a set of windows H that reach node n we select the value that at this node allows $v_t \cdot |H|$ windows to be retained.

3.4.3 Quantized Nodes

In literature the task of speeding-up the evaluation of an ensemble of Exemplar-SVMs has been tackled by Sadeghi *et al.* [120] exploiting Vector Quantization. We

show that their approach is orthogonal to ours and we propose a vector quantized indexing taxonomy.

In our framework, image patches are represented as HOG feature vectors [21]. The feature vector is built by dividing the patch in $N_w \times N_h$ cells. For each cell a histogram of oriented gradients is computed over N_b bins. The resulting feature vector is composed by $N_w \times N_h$ arrays of N_b bins (we use $N_b = 31$ as in [39], [91], [120]). When searching for an object in an image we compare a template of such object category which in our case is a learned exemplar-SVM. Therefore, a comparison between a learned HOG template (e.g. an Exemplar-SVM) and a HOG feature vector will require $N_w \times N_h \times N_b$ comparisons.

To approximate this computation, Vector Quantization requires to learn a dictionary in order to represent HOG feature vectors. This can be done simply by clustering random HOG cells and collecting the resulting centroids. This allows us to represent each cell as a scalar instead of a 31-dimensional array by taking the index of the closest centroid.

A convenient and efficient strategy to accelerate computation is to pre-compute values and store them in a lookup table. For each classifier a lookup table is then built by computing partial scores between the cells of the template and all the centroids. These lookup tables are data structures which can be indexed with a centroid index and return the precomputed partial score with a given template cell. In this way, at test time the full dot product between a template and a detection window is approximated as a sum of partial scores by looking them up in the table.

We can integrate this technique in our framework by replacing the representative of each node in the taxonomy with a lookup table. In this way we are able to exploit the fast evaluation of single templates offered by Vector Quantization and to maintain the speed-up offered by the taxonomy.

Representing the tree as a lookup table $\mathbf{T}^{C \times M \times N}$, where C is the number of cells in a template, M the number of centroids used to quantize the models and N the number of nodes in the taxonomy, we can reformulate the node selection for traversing the tree from Equation 3.8 as:

$$\texttt{next\_node} = \arg \max_{i \in \{L,R\}} \sum_{c=1}^{C} \mathbf{T}(c, \mathbf{h}(c), i) \tag{3.11}$$

where $\mathbf{h}(c)$ is the c–th cell of the current window $\mathbf{h}$ and L and R are the left and right children of the current node, respectively.

To ensure a fast evaluation of the lookup table we employ fixed point arithmetic for computing scores, as suggested in [120]. To do so we map the values in the

lookup table as 16 bit integers in the range $\left[-\frac{(2^{15}-1)}{C}; \frac{(2^{15}-1)}{C}\right]$ in order to be able to sum up to C signed values without overflow or underflow. To reverse to the correct score after evaluation, we simply rescale the value to the original range.

The complexity of evaluating a template on a single window hence amounts just to C table lookups and C sums between 16 bit integers instead of a dot product between two high dimensional floating point vectors. As stated in [120], the dot product could be faster than a table lookup, depending on the CPU architecture and on how data is stored in memory. We therefore ensure that at training time the lookup table is packed in a contiguous block of memory, permitting the usage of fast SIMD instructions sets such as SSE (Streaming SIMD Extensions) and AVX (Advanced Vector Extension). Moreover, a feature quantization step has to be added to the evaluation pipeline. In fact each HOG cell in the feature vectors extracted from the test windows have to be associated with a centroid. However the cost of this operation is negligible compared to the time needed to evaluate the ensemble.

All the proposed variants of our method are compatible with the Vector Quantized Taxonomy. In particular, for the rejecting taxonomy of Section 3.4.2 we simply compute the rejection thresholds directly in the quantized domain.

3.4.4 Rescoring

After evaluating the hierachy we rescore the best 5% of the test windows with the whole ensemble, in order to gather more detections. This step is required since the ESVM framework boosts detected bounding boxes using an exemplar co-occurrence matrix. This boosting procedure is less effective with a reduced number of detections, as in the case of our taxonomy which associates only one detection for bounding box. The rescoring step allows us to overcome this problem by evaluating the entire ensemble only on a restricted subset of windows, which requires a negligible additional cost. Furthermore this helps to mitigate the approximation effect of Vector Quantization. In fact a similar approach is used in [120].

3.4.5 Complexity analysis of ESVM ensembles

To evaluate and ensemble of ESVMs, a set of N windows taken from an image has to be tested against a pool of M Exemplar classifiers. Therefore, the complexity of evaluating an ensemble is $O(NM)$. Our proposed algorithm instead, in the case of a balanced binary tree has a complexity of $O(N \log_2(M))$ since each window only has to traverse it from the root to a leaf.

The rescoring step (Section 3.4.4) has an impact on this cost which depends on the windows to be rescored in an image and the number of exemplars, so it is $O(N'M)$ where N' is the amount of windows to be rescored. The cost will increase linearly in the amount of rescored windows, since the number of exemplar remains fixed. Our method has a trade-off which is found when $\log_2(M)N + N'M \geq NM$, which is not dependent on N since $N' = pN$, with p the fraction of windows to rescore. Hence, we obtain that the rescoring becomes expensive when $\log_2(M) \geq (1-p)M$. In case window rejection is applied a factor furtherly reducing the rescoring cost has to be considered. The above trade-off depends both on the amount of exemplars and on the percentage of rescored windows. It easily seen that for $p < 1$ there is always a value of M for which our method is faster than the ensemble. In practice our approach becomes inconvenient for small M (e.g less than 50) and high p (more than .15). Moreover the use of Vector Quantized Exemplars, effectively reduces the computational time required for each comparison.

3.5 Experimental results

We evaluate the proposed algorithm on the Pascal VOC 2007 object detection benchmark [34], comparing the results to the baselines given by the original ESVM framework [91] and its Vector Quantized version [120]. Besides accuracy we focus on the speed-up gain in the evaluation step. We also present results for a segmentation task based on label transfer, showing how our approach is able to provide comparable high quality masks with the monolithic ensemble formulation.

3.5.1 Object Detection

Object detection accuracy is important in order to establish how the taxonomy is able to approximate the capabilities of a standard ESVM ensemble. To build the hierarchical ensembles we use the pre-trained exemplars provided by [91] for each of the Pascal `trainval` objects (20 categories, 12608 exemplars). Each class is evaluated using a different taxonomy, specific to its class.

We report the results obtained using four different approaches: the standard hierarchy built through plain spectral clustering (`Tree`), an augmented version created using both exemplars and their horizontally flipped copies (`Tree-flip`, see Section 3.4.1) and the rejecting version of the two previous strategies (`Tree-rej` and `Tree-flip-rej`, respectively. See Section 3.4.2).

Method											
ESVM [91]	**19.0**	**47.0**	3.0	**11.0**	9.0	39.0	40.0	2.0	6.0	15.0	**7.0**
ELDA [59]	18.4	39.9	**9.6**	10.0	**11.3**	39.6	**42.1**	**10.7**	6.1	12.1	3.0
FTVQ [120]	18.0	**47.0**	3.0	**11.0**	9.0	39.0	40.0	2.0	6.0	15.0	**7.0**
Tree-flip-rej	18.0	45.5	2.2	9.0	9.4	39.0	37.4	1.6	**6.2**	13.5	6.1
Tree-rej	13.1	46.6	1.9	10.0	8.0	39.7	37.8	1.2	5.9	**15.5**	5.1
Tree-flip	17.6	45.5	2.3	9.2	7.7	39.0	38.0	1.5	**6.2**	13.6	6.1
Tree	13.0	46.4	2.1	10.5	7.2	**40.0**	38.5	1.3	5.9	**15.5**	**7.0**
										mAP	**Time**
ESVM [91]	2.0	**44.0**	**38.0**	13.0	**5.0**	**20.0**	**12.0**	**36.0**	28.0	**19.8**	6.48 ms
ELDA [59]	**10.6**	38.1	30.7	**18.2**	1.4	12.2	11.1	27.6	**30.2**	19.1	6.48 ms
FTVQ [120]	2.0	**44.0**	**38.0**	13.0	**5.0**	**20.0**	**12.0**	**36.0**	28.0	19.7	1.06 ms
Tree-flip-rej	1.2	41.9	37.8	8.8	3.0	17.1	10.5	31.1	27.2	18.3	**0.17** ms
Tree-rej	1.6	41.6	36.0	9.7	3.4	16.8	11.2	35.0	26.8	18.4	0.26 ms
Tree-flip	1.2	42.0	37.6	10.6	3.0	17.2	10.8	30.4	27.1	18.3	0.35 ms
Tree	1.5	42.0	37.5	11.7	3.3	17.7	10.9	34.0	27.0	18.7	0.59 ms

Table 3.2: **Results on the Pascal VOC 2007 dataset**. Comparison of our method with the Exemplar-SVM baseline [91]. The results obtained using Fast Template Vector Quantization [120] are also given as a term of comparison. The four variants of our approach (standard tree, tree with flip augmentation, rejecting tree, rejecting tree with flip augmentation) are reported, showing detection accuracy and timings. Running times refer to the mean time required for evaluating each exemplar on an image.

The same experiments are then repeated combining these techniques with the Fast Template Vector Quantization method of [120], which to the best of our knowledge is the fastest Exemplar-SVM formulation in literature (see Section 3.4.3).

Table 3.2 summarizes the results for all of the proposed methods, along with the ESVM ensemble baseline (`ESVM`) and the Fast Template Vector Quantization [120] (`FTVQ`) method applied to Exemplar-SVM. To provide these baselines we used the Exemplar-SVM framework of [91] available online and our implementation of FTVQ. All experiments have been performed on an Intel Core i7-2600K, 4 x 3.40Ghz to provide a fair comparison of timings.

Detection accuracy is reported in terms of mean Average Precision (mAP) and the evaluation time represents the average time to evaluate an image with a single exemplar. This is a cost which is normalized both with the number of images to evaluate and with the number of classifiers in the ensemble. Average Precision is calculated by numerically integrating the area under the Precision-Recall curve for each class. This is the standard evaluation metric for object detection since

Pascal VOC 2010 [33], which differs from the 11-point AP used previously and in particular in [91].

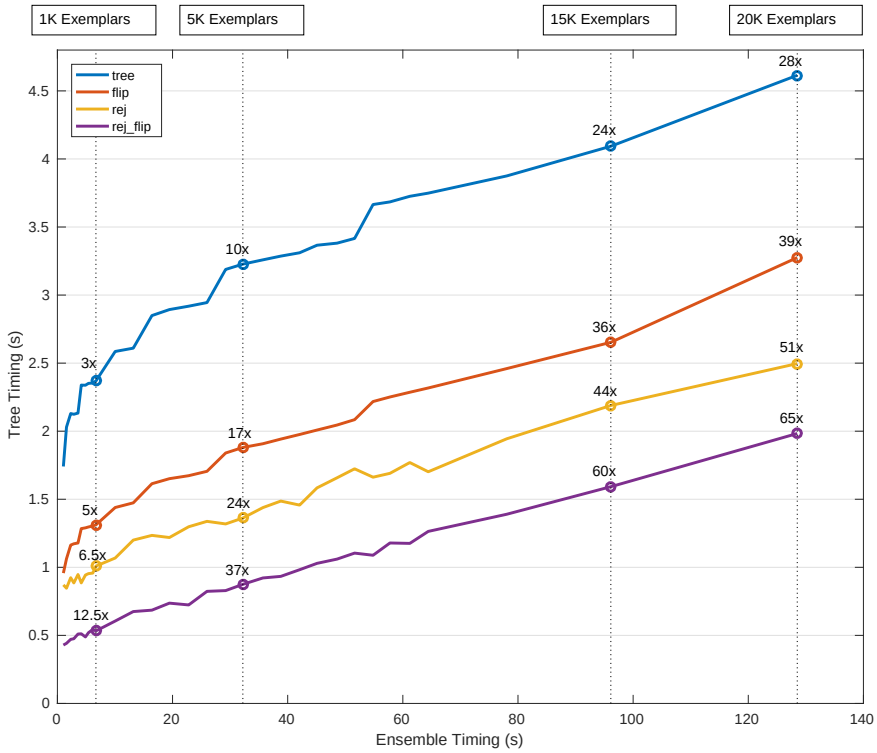

Figure 3.5: Plot of timings varying the number of exemplars. The average execution time for evaluating an image of the four variants of our method (*Tree Timing*) is compared with the time required by the monolithic ensemble [91] (*Ensemble Timing*). A logarithmic trend is observed in relation to the number of exemplars. Speed-up values are reported, at the correspondent mark in the plot, for 1k, 5k, 15k and 20k exemplars.

The `Tree` method obtains a mAP of 18.65, which is comparable to the ESVM ensemble baseline and is on average 10 times faster. The use of the flipped exemplars strategy instead, at almost no additional cost in mAP, lowers by 60% the overall evaluation time since we have to evaluate only the original unflipped windows, obtaining a $18\times$ speed-up with respect to the ESVM baseline.

Employing rejecting thresholds we are able to reach a $25\times$ and $38\times$ speed up

for the `Tree-rej` and the `Tree-rej-flip` methods, respectively. These two methods leave the mAP of the system almost unchanged with respect to the standard `Tree` algorithm, meaning that we are able to prune windows that do not contain any detectable object. All of the proposed variants of the method are faster than FTVQ.

To analyse the benefit of our approach in relation with the ensemble size, in Figure 3.5 we report a comparison of timings between our methods and the `ESVM` baseline. Each point in the plot compares a timing obtained with the same number of exemplars with both methods. To perform these experiments, large ensemble have been syntetically generated in order to gather timings. The plot shows a clear logarithmic relationship between the timings of the ensemble and the `Tree` and `Tree-flip` methods. Using the rejecting taxonomy, the logarithmic trend is less evident since the number of comparisons is reduced, depending on the image content.

It is important to note that the speed-up increases as more models are used. For example, considering the `Tree` method, we have a $10\times$ speed-up with 5k exemplars but we are able to obtain a $28\times$ speed-up when 20k exemplars must be evaluated. The same behaviour is registered for the other variants reaching a $65\times$ gain with `Tree-rej-flip` at 20k exemplars, going from more than 2 minutes to less than 2 seconds. In the experiments reported in Table 3.2 this speed-up is not appreciable since most of the classes in Pascal have a small number of exemplars (200-300 on average) and we are averaging timings on all classes.

We perform the same analysis on our Vector Quantized variants of the taxonomies, comparing them to the fairer baseline of FTVQ [120]. In Table 3.3 the Object Detection results on Pascal 2007 are shown. Overall, the mAP of the method is slightly lower than the respective unquantized versions but we are able to improve the mean execution time to less than 0.1ms per exemplar per image for each variant. Note that on average all quantized tree methods have almost the same speed. This is caused by the feature quantization step which dominates over the evaluation phase. This reflects on the trends of the plots in Figure 3.6. Here the speed of the vector quantized hierarchies is compared to the speed of FTVQ. Whereas all these techniques are much faster than `ESVM`, it can be seen how the benefit brought by the taxonomy is even bigger. With 20k exemplars we obtain from 39x speed-up with `VQ-Tree` to 95.5x with `VQ-Tree-rej-flip`. As stated before, the time required by the rejecting and flipped versions of the quantized tree are almost the same for a relatively small amount of exemplars due to a fixed time needed to quantize the features from the test windows.

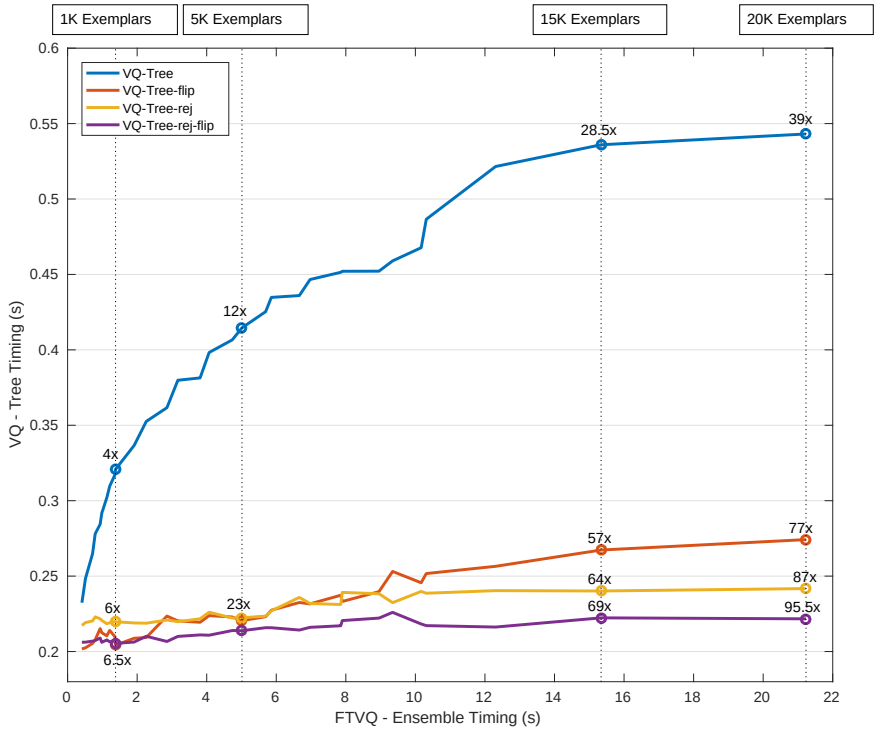

Figure 3.6: Plot of timings varying the number of exemplars applying Vector Quantization. The execution time of the four variants of our Vector Quantized Taxonomies are compared with the time for evaluating the monolithic FTVQ ensemble [120]. Speed-up values, at the correspondent mark in the plot, are reported for 1k, 5k, 15k and 20k exemplars (overlapping points are omitted). A logarithmic trend is observed in relation to the number of exemplars with the `VQ-Tree` variant while the others are domniated by the linear quantization cost.

Method											
ESVM [91]	**19.0**	**47.0**	3.0	**11.0**	9.0	39.0	40.0	2.0	6.0	15.0	**7.0**
ELDA [59]	18.4	39.9	**9.6**	10.0	**11.3**	39.6	**42.1**	**10.7**	6.1	12.1	3.0
FTVQ [120]	18.0	**47.0**	3.0	**11.0**	9.0	39.0	40.0	2.0	6.0	15.0	**7.0**
VQ-Flip-rej	17.7	45.0	2.2	9.0	7.3	34.9	36.7	1.2	5.3	14.8	4.6
VQ-Tree-Rej	14.8	46.3	1.8	10.0	7.9	**39.2**	37.5	1.6	5.5	15.3	3.6
VQ-Tree-Flip	17.1	44.8	2.2	9.1	7.1	34.7	37.0	1.3	5.2	14.5	4.4
VQ-Tree	15.0	46.0	2.0	9.7	8.0	**39.2**	37.4	1.7	5.6	**15.4**	4.7
										mAP	**Time**
ESVM [91]	2.0	**44.0**	**38.0**	13.0	**5.0**	**20.0**	**12.0**	**36.0**	28.0	**19.8**	6.48 ms
ELDA [59]	**10.6**	38.1	30.7	**18.2**	1.4	12.2	11.1	27.6	**30.2**	19.1	6.48 ms
FTVQ [120]	2.0	**44.0**	**38.0**	13.0	**5.0**	**20.0**	**12.0**	**36.0**	28.0	19.7	1.06 ms
VQ-Flip-rej	1.2	40.8	36.8	10.5	2.7	16.8	10.9	30.0	26.8	17.8	**0.04** ms
VQ-Tree-Rej	1.5	41.5	36.9	10.6	3.4	16.6	11.2	34.5	27.8	18.4	0.05 ms
VQ-Tree-Flip	1.3	41.1	36.6	10.4	2.7	16.7	10.2	29.9	26.9	17.7	0.05 ms
VQ-Tree	1.6	41.3	36.6	11.0	3.3	16.5	11.5	34.9	**28.0**	18.5	0.08 ms

Table 3.3: **Results on the Pascal VOC 2007 dataset (quantized version).** Comparison of the quantized version of our method with the Exemplar-SVM baseline [91]. The results obtained using the Fast Template Vector Quantization [120] are also given as a term of comparison. The four variants of our Vector Quantized Taxonomies (standard tree, tree with flip augmentation, rejecting tree, rejecting tree with flip augmentation) are reported, showing detection accuracy and timings. Running times refer to the mean time required for evaluating each exemplar on an image.

To provide a final overview, in Figure 3.7 we show the execution time varying the number of exemplars for all the proposed methods. As expected, the vector quantized taxonomies perform much better than their unquantized counterparts and the usage of rejection thresholds and flipped exemplars is crucial in ensuring an extremely fast evaluation of the method. On the plot are underlined the speed-ups obtained by the most performing variant of the algorithm: `VQ-Tree-rej-flip`. Evaluation speed with a large number of classifiers is a couple of orders of magnitude faster than `ESVM`, lowering the required time from approximately 2 minutes to half a second for 20k exemplars with more than a 500-fold gain.

We analyze the rescoring effect for the `Tree` method varying the percentage of rescored windows. As can be seen in Figure 3.8 using less than 5 % of the windows affects the mAP by almost 2 points. Increasing the fraction of windows to be rescored with the whole ensemble the mAP obviously saturates to the baseline value. Note that rescoring the 5% of the windows has a cost of 0.1 ms per exemplar.

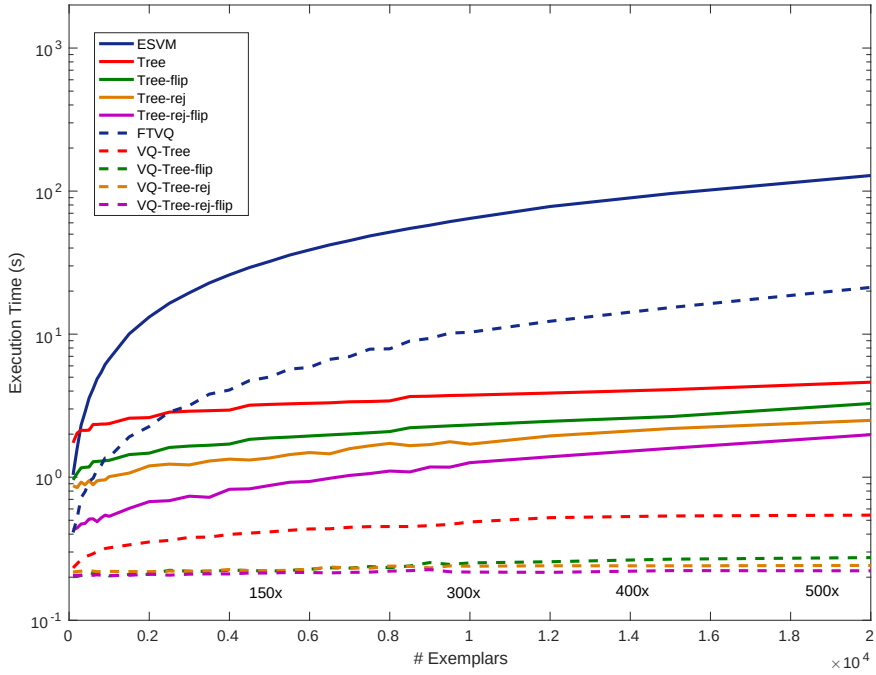

Figure 3.7: Comparison of execution times for all the presented methods against the ESVM [91] and FTVQ [120] baselines varying the number of exemplars in the ensemble. Vector Quantized taxonomies are reported as dashed lines while continuous lines are the respective unquantized formulations. Speed-ups are reported for our fastest method (`VQ-Tree-rej-flip`).

This operation becomes expensive above 15% of the windows.

3.5.2 Label Transfer Segmentation

Along with object detection accuracy we evaluated the label transfer capabilities of the system performing a segmentation experiment. In order to do so, we manually annotated the Pascal VOC *Bus* class (229 exemplars and 213 test objects for a total of 442 buses) with segmentation masks. Given a set of detection stemmed from a bank of exemplars the following procedure can be applied to generate segmentation masks. First we weight all transferred masks according to the exemplar detection confidence and accumulate such weighted masks into a segmentation map. To re-

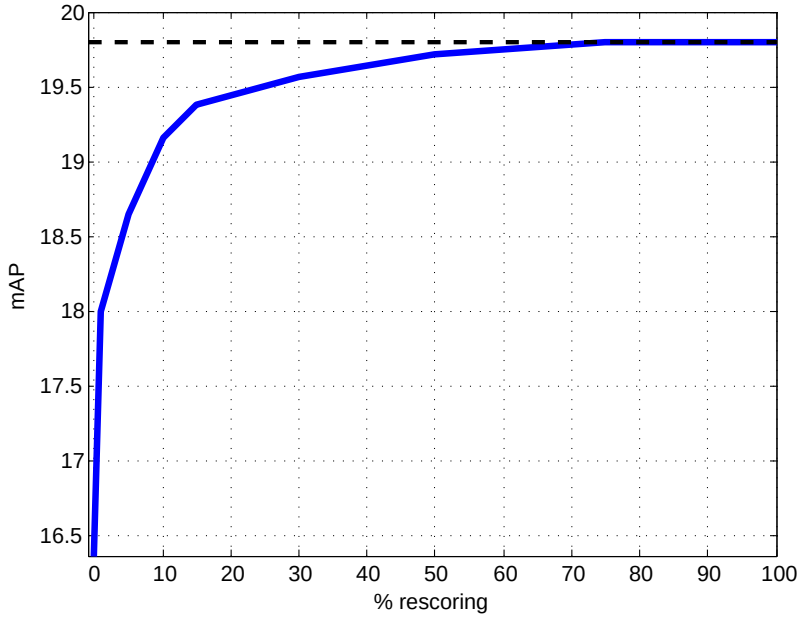

Figure 3.8: Mean average precision varying the percentage of rescored windows. Rescoring the whole window set is equivalent to evalute the full Ensemble. We report mAP without rescoring as 0%.

move noise, we rescale the mask into $[-1, 1]$ and threshold it. Mask rescaling is necessary on one hand to avoid misses in case few exemplars have positive score, on the other hand to avoid over segmentations when many exemplars are matched. Threshold is cross-validated on the validation set. As baseline we use the standard monolithic ensemble [91] and compare it to `VQ-Tree-rej-flip`. The segmentation masks have been evaluated for both the *Bus* class and the *Background* class, measuring a pixel-wise accuracy

$$Acc = \frac{tp}{tp + fp + fn} \tag{3.12}$$

as specified in the Pascal guidelines for segmentation. Using the hierarchical structure we obtained a 48.07% accuracy for the *Bus* class and 76.30% for the *Background* against a baseline of 49.59% and 77.14%, respectively. Some qualitative results are reported in Figure 3.9 where the segmentation masks generated by our method and by the baseline are compared. These results show that with our index-

Figure 3.9: Segmentation masks produced by Exemplar-SVM label transfer. Green masks in the upper row are produced using the standard ensemble [91], red masks in the lower row are generated using our hierarchical ensemble.

ing strategy we are still able to retrieve objects which maintain a good alignment with test objects, leaving the label transfer capabilities of the framework almost unaffected.

3.6 Conclusion

In this work we have proposed a technique to overcome the main drawback of the Exemplar-SVM framework, i.e. the high computational burdain at test time. In fact, even if it has found large interest for many computer vision tasks, the real applicability of this technique is limited due to the linear dependency in the number of examples. We solve this issue by building a quantized indexing taxonomy to access data at a logarithmic cost and just a small loss in detection performance. We have shown how combining different strategies to speed-up the evaluation we are able to provide a reliable approximation of a ESVM ensemble with a very small

computational footprint, which is orders of magnitude faster than the original formulation. Moreover we have shown how our indexing impacts the label transfer property of the method, applying the technique to a segmentation task. Confirming the previous experiments, the results are satisfactory providing high quality masks compared to the baseline.

Chapter 4

Imaging Novecento. A Mobile App for Automatic Recognition of Artworks and Transfer of Artistic Styles

Imaging Novecento *is a native mobile application that can be used to get insights on artworks in the "Museo Novecento" in Florence, IT. The App provides smart paradigms of interaction to ease the learning of the Italian art history of the* 20^{th} *century.* Imaging Novecento *exploits automatic approaches and gamification techniques with recreational and educational purposes. Its main goal is to reduce the cognitive effort of users* versus *the complexity and the numerosity of artworks present in the museum. To achieve this the App provides automatic artwork recognition. It also uses gaming, in terms of a playful user interface which features state-of-the-art algorithms for artistic style transfer. Automated processes are exploited as a mean to attract visitors, approaching them to even lesser known aspects of the history of art.* [1]

[1] Parts of this chapter have been published as "Imaging Novecento. A Mobile App for Automatic Recognition of Artworks and Transfer of Artistic Styles" in *Proc. of International Euro-Mediterranean Conference (EUROMED)*, Nicosia (Cyprus), 2016. [9].

4.1 Introduction

Modern museums can provide new paradigms for experiencing artworks. Thanks to the technological development, novel initiatives include pervasive uses of tech to create interactive experiences for visitors throughout a museum. However, making content relevant and appealing through these modern technologies is a difficult problem, requiring more and more interactivity as the audience is shifting towards a 'multimedia point of view'. Moreover, while the massive amount of available artworks constitutes a huge resource for education and recreation purposes, it can also be a cognitive burden for visitors.

The cognitive process related to learning has been an active subject of study in recent decades. According to cognitive load theory, learners must cope with a certain level of cognitive effort to process new information [105]. In this regard, multimedia education, defined as "presenting words and pictures that are intended to foster learning" [95], can be an effective remedy because it facilitates the activation of sensory and cognitive perceptions (e.g. visual and notional memory), avoiding visitors from information overloading. This can also be reinforced by gamification, that is the use of playful experience to help a user find personal motivations and engagement with serious content [114]. This combination can enhance the visitor's involvement and further lower its cognitive effort. Using gamified applications, museum visitors have the opportunity to feel the emotion of a game, share results with friends on social networks or become part of a game community [100]. This aspect of learning through gaming is even more valuable in the context of the "Bring Your Own Device" (BYOD) approach [7] that allows on demand access to digital content on personal devices. The BYOD approach and gamification have been identified in the NMC Horizon Report 2015 to be increasingly adopted by museums in one year's time or less for mobile and online engagement [75].

In this chapter we report our experience in embedding these concepts into *Imaging Novecento*, a system built around a mobile application developed for the museum "Museo Novecento" in Florence, IT. We aimed at improving the learning process of the visitors by exploiting a simple gamification paradigm, and at reducing visitors cognitive load. To this end, we also developed a state-of-the-art computer vision system that is able to 1) recognize artworks from photos; 2) apply their style to user photos. These technologies are well suited for integrating the BYOD and gamification paradigms. The chapter is organized as follows: in Sec. 4.1.1 we briefly describe the context of the project, in Sec. 4.1.2 we establish the specifications of the project, in Sec. 4.2 we describe the proposed system and its components. Finally in Sec. 4.3 we report the conclusions of our experience.

4.1.1 The Museo Novecento in Florence and Innovecento

The "Museo Novecento" in Florence, IT, is a museum opened on June 24, 2014. The museum is dedicated to the Italian art of the 20^{th} century and offers a selection of about 300 artworks distributed in fifteen exhibition halls on two levels. The venue is located in the former hospital of the "Leopoldine" in Piazza Santa Maria Novella. The museum has been an example of innovation since its genesis, thanks to the prompt adoption of the latest multimedia technologies.

In March 2015, in order to improve the visitor experience, the Municipality of Florence has published an open call "INNOVecento - Novecento Museum Innovation Lab" inviting companies and professionals to propose ideas and solutions based on ICT. Five companies specialized in technologies applied to cultural heritage have already responded to the call which, at the time of writing, is still open. As NEMECH, centre of competence of the Tuscany region in Italy, we proposed *Imaging Novecento*. The App features automatic recognition of artworks through the visitor's smartphone and automatic transfer of artistic styles from artworks. These styles can be applied to user images.

4.1.2 Motivations and design

The target of the App is rather wide. Although *Imaging Novecento* can be used by anyone (e.g. tourists and residents), during the design process we identified a specific audience. We mainly target the App towards people in a relatively young age (between 14 and 30 years old), more accustomed to digital technologies, open to technological innovation and to gamification.

One of the main ideas of the App is to exploit the pervasiveness of mobile cameras in modern smartphones to reduce the cognitive effort required to museum visitors. In fact, despite themed rooms and the ubiquitous explanatory cards, users can still be overwhelmed by the great number of artworks present in the museum. Labels in museums can be very concise or, on the contrary, can be filled with lots of explanation, often generic, not highlighting salient features of individual paintings. By using *Imaging Novecento*, the visitor can take a picture of the artwork he is interested in. The App will automatically recognize the painting and provide related information. Another reason for the adoption of this automatic process is the resistance of museums' curators to place or attach additional materials, such as QR codes or BLE iBeacon [63], next to artworks.

Furthermore, tourists and school groups are usually 'hit-and-run' visitors who tend to rapidly forget or do not have the time to process the overload of infor-

mation. To solve this issue, *Imaging Novecento* leverages a playful feature that employs state-of-the-art algorithms for transferring artistic styles from recognized artworks to user images. This is done using a gamification paradigm at the interface level. Gamification techniques have been proved to be useful in engaging students in the learning process, improving their skills and maximizing their long-term memory [23].

4.1.3 Previous work

Several previous works have addressed the problem of providing an engaging experience to museum visitors. Rapid technological development has led to the implementation of a lot of applications. There are several active trends for virtual museums: immersive reality [43,92], natural interaction installations [8,37], mixed reality, mobile applications [16, 157]. While they all offer increasing engagement of visitors, only recently studies on the effects of audience have been carried out [75, 104]. In particular, a recent audience study has been conducted on the case of the "Keys to Rome" international exhibition, hosted at the "Imperial Fora Museum" in Rome in 2015, to assess the impact of these technologies on cultural heritage. The exhibition was made up of 11 digital installations and applications, installed in the museum [104]. The study highlights some fundamental aspects that must be taken into account when designing applications for virtual museums: 1) the majority of museum visitors are tourists and school groups; 2) visitors generally require applications with an high level of interactivity, particularly on their mobile devices; 3) it is essential for the UX design to use metaphors of informal learning capable to stimulate attention, memory and engagement (e.g. through gamification) in visitors.

Automatic artwork recognition Automatic artwork recognition is a long standing problem in applications for cultural heritage. Descriptors such as SIFT and SURF have been used for years in order to address this task [118, 135] due to their accuracy in recognizing paintings. Crowley and Zisserman [19] retrieve artworks finding object correspondences between photos and paintings by using a deformable part based method. More recent approaches for artwork recognition adopt Convolutional Neural Networks (CNN) as in [4], where a holistic and a part based representation are combined. Peng and Chen [106] exploit CNNs to extract cross-layer features for artist and artistic style classification tasks. Artistic style recognition is also performed in [77] on two novel large scale datasets. Similarly to these works, we explore the use of CNNs features but we aim to obtain a global rep-

resentation that is semantically meaningful and also capable of retaining low level visual content information. Artwork recognition has also been used with wearable devices, as in [8] where the user's position is jointly estimated with what he is looking at.

Artistic style transfer Regarding the application of artistic style to photos, a lot of research has been done in the past. The problem of rendering a given photo in the style of a particular artwork is known in literature as a branch of non photorealistic rendering [84]. This class of works use texture transfer [30, 151] to achieve style transfer. These techniques are non-parametric and directly alter image pixels of the content image into pre-defined styles. Another direction of work focuses on the idea of separating style and content in order to 'remix' them together in different configurations. First works were evaluated on much simpler images such as characters in different handwritings [136] or images representing human body configurations [31]. Only recently, the breakthrough paper from Gatys *et al.* [50] showed the possibility of disentangling the content from the style of natural images by using a convolutional neural network based representation. The advantage of this approach is the capability of performing style transfer from any painting to any kind of content images. The approach was recently extended with a more advanced perceptual loss [74] and also applied to movies [3] by considering the optical flow.

4.2 The System

The system is composed by two main components: a mobile App and a computer vision system responsible to address the two tasks of automatically recognize artworks and apply artwork styles to user photos. The mobile App is used by the visitor in the museum and is the *fulcrum* of the user interaction. Once installed by the user in his mobile phone, it allows to take pictures, deliver artwork information and request the style transfer to new photos. Due to the limited amount of computational power available on most mobile devices, the computer vision system is deployed on a scalable web server system that processes requests from the mobile App. Since the two tasks use quite different technologies, we discuss them separately in the following sections.

4.2.1 The Mobile App

Imaging Novecento has been developed as an Android application using Ionic [2]. Ionic is a framework, based on Sass and AngularJS, for building highly interactive native web apps through mobile-optimized HTML, CSS and JS components and tools. *Imaging Novecento* is a contextual App that can be used exclusively inside the Museo Novecento in Florence. An information flyer of the App is delivered to the visitor at the ticket office. In the flyer there are a QR code, through which the visitor can download the App from the Google Play store, and the list of the artworks on which the App can perform the automatic recognition and style transfer processes. The list comprises a selection of twenty artworks for which the museum's curators have provided multimedia materials. The App interface (Fig. 4.1) is quite simple and is organized in two main views: 1) the Camera View and 2) the Artwork Details view.

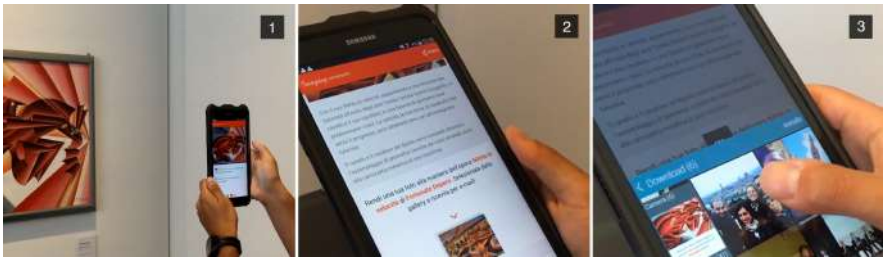

Figure 4.1: *Imaging Novecento* in action: 1) the user takes a picture of an artwork; 2) the artwork is recognized and insights are shown; 3) the user selects a photo from his own gallery in order to apply that artwork style and to share the results on social networks.

The Camera View allows the visitor to frame one of the artworks on the list in order to have it immediately recognized by the automatic system. Proper feedback is given in case the recognition is not successful. Once the artwork is recognized, the Artwork Details view is activated. In this view, exhaustive but concise information about the author, the history of the artwork and its artistic style are given. An infographic is presented to the user. It works as a "call to action" for enabling the transfer of the recognized painting style to a photo from the user's device gallery. The infographic provides an animated preview that shows the result of the artistic style transfer on a predefined picture. After the image has been successfully uploaded, the remote process for style transfer is performed. The result of the elab-

[2] `http://ionicframework.com/`

oration is then sent to the user in a few minutes by email. The image has a resolution of 900px wide preserving the original image aspect ratio and can be shared on the most popular social networks (e.g. Facebook).

4.2.2 Automatic Artwork Recognition

Artwork recognition is performed through a Python web server with a REST interface. The server processes the image and returns the ID of the recognized painting. The recognition step combines modern deep features with classical Support Vector Machines (SVM) in order to classify photos of paintings. Image features are extracted using a deep convolutional neural network (CNN), and are then evaluated using a set of classifiers, one for each recognizable artwork. The neural network we adopted is the Caffe reference model [73], fine-tuned for style recognition using the FlickrStyle dataset [77][3].

To obtain a representation which is at the same time semantically meaningful and capable of retaining low level visual content information, we extract image features from an intermediate level of the network. In particular, we adopt the *pool5* feature map, the latest one before the fully connected (FC) layers of the CNN. In fact, FC layers trade spatial information for a more semantic representation, which is highly coupled with the task and with the visual domain on which the network has been trained. This choice is therefore motivated by the fact that our visual domain, while being quite close, is different from the one of FlickrStyle. Moreover,

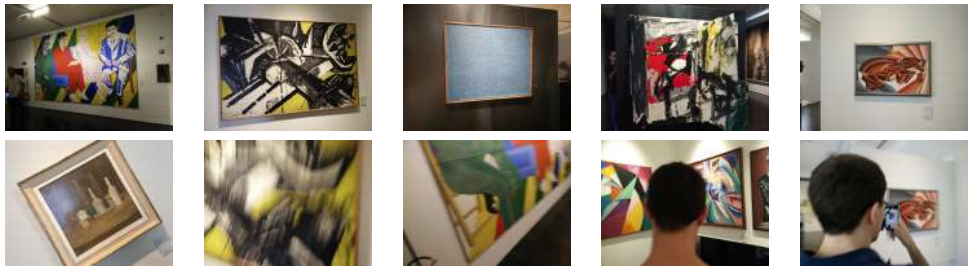

Figure 4.2: Samples from the dataset collected at the museum. On the first row standard pictures are shown, depicting the painting in their entirety. On the second row instead, are reported more challenging photos, due to blur, occlusion or rotation.

[3]the network is available online at `http://caffe.berkeleyvision.org/gathered/examples/finetune_flickr_style.html`

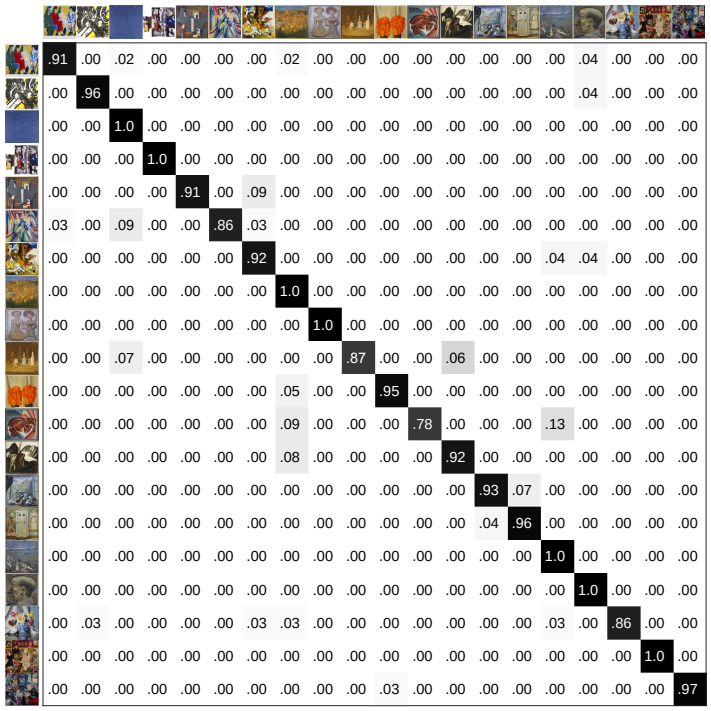

Figure 4.3: Confusion matrix for the artwork recognition module. Each row indicates the percentages of correct and incorrect classifications for a given artwork.

since a sufficiently large dataset was not available to perform a further fine-tuning step, SVM classifiers have been trained to adapt the framework to the App's domain and be able to classify artworks correctly. For training the classifiers we used approximately 1,800 images, gathered at the museum using different smartphones and tablets, namely Galaxy S4, Galaxy Tab, iPhone 6, iPad Mini and OnePlus One. These images represent all of the twenty artworks plus a 'negative' set of images containing other scenes and paintings inside of the museum. They are used to reduce the false positive rate when the user accidentally attempts to recognize other paintings. All the classifiers are One vs All SVMs. During the evaluation phase, the ID of the highest scoring one is returned to the mobile App, if it scores above a cross validated threshold. Details about the recognized artwork are then provided to the user, who can upload a personal photo to get the style of the painting transferred on to it. Calls to the webserver are handled asynchronously and each request

takes approximately 300ms on a CPU.

In order to test the recognition accuracy we collected an additional set of photos which were not used for training. For each one of the twenty artworks in our system, we collected approximately 30 photos taken from different viewpoints, with different scales and degrees of occlusion. Fig. 4.2 shows some of the photos from the test set. Some of them are "difficult" in a sense that might be blurred or taken from challenging viewpoints and artworks may be partially occluded by other visitors. Despite these difficulties our system achieves an overall good performance with a mean accuracy of 94.01%. In detail, in Fig. 4.3 we report the confusion matrix for the twenty artworks in the test set, showing how often each painting is correctly classified or confused with other artworks. As can be seen, the majority of artworks are perfectly recognized. Only four artworks have performance slightly inferior to 0.9, due to the difficult lightening conditions present in their specific locations at the museum.

4.2.3 Artistic style transfer

From the Artwork Detail view of the mobile App, the user has the possibility to upload a personal image on which the style of the artwork will be applied. In this way, entertaining personal pictures that share similarities with the artworks can be obtained and shared on social networks. As a result, a visit at the museum can become a playful experience, combining gaming and learning aspects for young visitors. We base our approach on that of Gatys *et al.* [50], that is capable of freely mixing style and content of two different photos. The main advantage of this approach is its broad applicability to different styles, in contrast to fixed handcrafted styles [30, 151]. This allows a museum curator to easily add new artworks in the system without requiring the development of a new transfer style algorithm. Following [50], our approach uses a CNN to derive a neural representation of content and style. The feature responses of a pre-trained network on object recognition (VGG-19 [128]) are used to capture the appearance of an artwork image and the content of a user photo under the form of texture information. We start from a blank novel image that is altered with back-propagation until its neural representation is similar in terms of euclidean distance to the style and content representations.

Unfortunately, the generation of the image is quite computational intensive. For an image of 900 pixel large, it takes about $\sim$90 seconds on a K80 NVIDIA GPU. As a result, the requests have to be handled offline since it is not possibile to obtain the output image in few seconds. Considering also that multiple requests can be made at the same time from multiple users, we implemented a scalable web server

Figure 4.4: Two examples of image stylization: 1) Baccio Maria Bacci, "Il tram di Fiesole", applied to a picture of the Battistero in Florence, IT; 2) Alberto Moretti, "Malcom X ed altri", applied to a picture of Piazza della Repubblica, also in Florence.

that is able to be easily deployed on several interconnected nodes. Web requests are handled in Python and enqueued to a distributed queue run by a Celery[4] server. By treating each request as a single unit task, it allows to process the images in a distributed batch fashion on several GPUs and several servers if available. After completing the computation, each output image is sent to the user via email, together with a description of the artwork. We also include links to share the image to several social media, with the aim of enabling viral publicity of the museum.

4.3 Conclusion

We presented the *Imaging Novecento* App, recently developed for the "Museo Novecento" in Florence, IT. Following previous studies on cultural heritage audience and applications, the App aims at enhancing the experience in the museum reducing cognitive load and exploiting gamification. The App automatically recognizes a selection of paintings and provides insights on artworks and their authors. The user can upload a personal picture with his smartphone to get it stylized with the recognized artwork style. He also has the possibility of sharing it on social networks. In this work we show how computer vision technologies can be exploited to

[4] `http://www.celeryproject.org`

increase interactivity and reduce cognitive load. This can attract the targeted audience to the museum and further engage people with content. A system evaluation and a user study of the system are being conducted and will be the subject of future work.

Acknowledgments.

We acknowledge the support of the “Museo Novecento” in Florence, IT, and the Municipality of Florence, IT.

Chapter 5

Segmentation free object discovery in videos

In this chapter we present a simple yet effective approach to extend without supervision any object proposal from static images to videos. Unlike previous methods, these spatio-temporal proposals, to which we refer as "tracks", are generated relying on little or no visual content by only exploiting bounding boxes spatial correlations through time. The tracks that we obtain are likely to represent objects and are a general-purpose tool to represent meaningful video content for a wide variety of tasks. For unannotated videos, tracks can be used to discover content without any supervision. As further contribution we also propose a novel and dataset-independent method to evaluate a generic object proposal based on the entropy of a classifier output response. We experiment on two competitive datasets, namely YouTube Objects [111] and ILSVRC-2015 VID [119]. [1]

5.1 Introduction

Image and video analysis can be considered similar on many levels, but whereas new algorithms are continuously raising the bar for static image tasks, advancements on videos seem to be slower and hard going. What makes video comprehen-

[1] Parts of this chapter have been published as "Segmentation Free Object Discovery in Video" in *Proc. of European Computer Vision Conference (ECCV), Workshop on Brave new ideas for motion representations in videos*, Amsterdam (Netherlands), 2016. [20].

sion more difficult is mainly the huge amount of data that has to be processed and the need to model an additional dimension: time.

We believe that focusing on relevant regions of videos, such as objects, will reduce the complexity of the problem and ease learning for models like Deep Networks. The same concept has been successfully applied to images using object proposals [67], which analyse low level properties, such as edges, to find regions that are likely to contain salient objects. Advantages are twofold, first the search space is considerably reduced, second, as a consequence, the number of false positives generated by classifiers is lowered.

Object proposals provide a relatively small set of bounding boxes likely to contain salient regions in images, based on some *objectness* measure. Different proposals, such as EdgeBoxes [165], are commonly used in image related tasks to reduce the number of candidate regions to evaluate. Recently, there have been some attempts to adapt the paradigm of object proposals to videos to solve specific tasks, by generating consistent spatio-temporal volumes. In [111] motion segmentation is exploited to extract a single spatio-temporal tube for video, in order to perform video classification. The task of object discovery is tackled in [134] by generating a set of boxes using a foreground estimation method and matching them across frames using both geometric and appearance terms. Kwak *et al.* [83] combine a discovery step matching similar regions in different frames and a tracking step to obtain temporal proposals. In [103] a classifier is learnt to guide a super-voxel merging process for obtaining object proposals. Temporal proposals have been exploited to segment objects in videos in [150] by discovering easy instances and propagating the tube to adjacent frames. Other methods to generate salient tubes have been proposed for action localization in [158] using human and motion detection.

Differently from the above approaches we do not rely on segmentation, which is a time-consuming task especially for videos. Our method is simply based on the response of a frame-wise proposal method. The weak supervision obtained from the temporal consistency of the video is exploited to generate tracks. Our method aims at generating few, highly precise, tracks containing objects in the video.

In this work we propose a technique to include time into a generic object proposal, by exploiting the weak supervision provided by time itself to match spatial proposals between adjacent frames. This results in spatio-temporal tracks that represent salient objects in the video and can therefore be used instead of the whole sequence. To the best of our knowledge we are the first to adopt a fully unsupervised matching strategy that only relies on bounding box coordinates without any

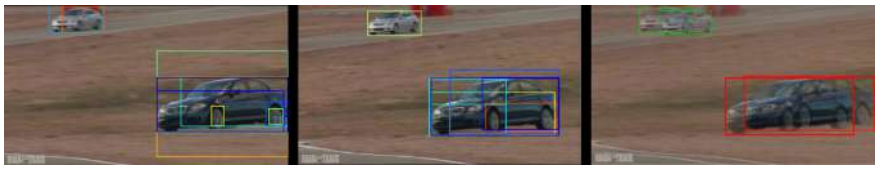

Figure 5.1: Example of frame matching; matched boxes are inserted into their respective track. *(left)* reference frame where the top 10 proposals extracted with EdgeBoxes are shown; *(center)* following frame with top 10 EdgeBoxes proposals; *(right)* 2 matched proposals between the two frames; these will be part of two different tracks

semantic content or visual descriptor apart from optical flow. We also introduce a novel dataset-independent proposal evaluation method based on the entropy of classifier scores and an optical flow based correction and filtering approach to get more accurate spatio-temporal proposals.

The article is organized as follows: in Section 5.2 we explain our method in details, introducing "tracks" and how we compose them from spatial proposals. In Section 5.3 we present our novel entropy based evaluation and in Section 5.4 we show some experimental results on two different datasets. Finally, in Section 5.5 we give the conclusions of our work.

5.2 Video Temporal Proposals

In this section we introduce the concept of "track", describing in details how these are generated from a set of bounding boxes extracted by an object proposal in the video.

Given a video V, for each frame f_i we extract a set B_i of bounding boxes b_i^k using an object proposal. We propose a method to match boxes that exhibit a temporal consistency in consecutive frames through the video, yielding to a set T of tracks t_j. A track is defined as a succession of bounding boxes b_i^k for which the intersection over union (IoU) between two boxes b_i^m (belonging to frame f_i) and b_{i+1}^n (belonging to frame f_{i+1}) is above a defined threshold θ_τ.

Starting from the first frame, each time a match is found, the corresponding bounding box is added to the end of the track and becomes the reference box for the following frame. If no match is found the last box of the track is compared with the following frames until a good match is obtained. An example of matching is

shown in Fig. 5.1.

When one or more consecutive matches are not found, tracks become fragmented, i.e. there are frames for which a track is active but there is no bounding box. This is usually due to a lack of good bounding boxes for that frame, occlusion or appearance changes of the object. It is thus necessary to avoid matching boxes in frames too far apart that therefore do not represent the same content, but at the same time we want to be able to tolerate some missing boxes without prematurely terminating the track.

To this end we introduce a Time to Live counter (TTL) τ for each track. We define $\tau_i(t_j)$ as the number of frames, at frame i, that the method can still wait before considering the track t_j terminated. TTL starts from an initial value γ; each time a box can not be matched in a consecutive frame the TTL is decremented, otherwise is incremented (up to γ). More formally, given a track t_j and its last bounding box b_i^m we increment or decrement its TTL as follows:

$$\tau_{i+1}(t_j) = \begin{cases} \tau_i(t_j) + 1, & \text{if } \exists\, n : \text{IoU}(b_i^m, b_{i+1}^n) > \theta_\tau \\ \tau_i(t_j) - 1, & \text{otherwise} \end{cases} \tag{5.1}$$

When the TTL for a track reaches 0, the track is considered terminated. Missing frames caused by track fragmentation are linearly interpolated using the positions of the previous and following bounding boxes in the track.

Proposal Motion Compensation Proposals around objects in consecutive frames are usually unaligned due to movements of the object or the camera. This causes the IoU score to decrease even if the matching is good. We work around this problem by registering the boxes with optical flow before computing the IoU. The registration is performed on the last box of each track, by computing the mean offsets along the x and y axes inside the boxes. Shifted boxes are only used for matching and tracks consist only of unaltered boxes.

Temporal NMS As in the spatial case, temporal proposals also suffer of high redundancy. To reduce this effect we extend spatial non-maximal suppression to time, defining a temporal NMS where instead of computing IoU on areas it is computed over volumes (vIoU). If α_j^k is the area of the k-th bounding box in track t_j, then the volume v_j of the track is calculated as $v_j = \sum_{k\,=\,0}^{K} \alpha_j^k$ where K is the length of the track. Then, vIoU is defined as:

$$\text{vIoU}(t_j, t_k) = \frac{v_j \;\cap\; v_k}{v_j \;\cup\; v_k} \tag{5.2}$$

Using vIoU we apply the standard NMS.

Proposal Suppression Once all the tracks are computed for a given video, we apply a post-processing to remove the ones that are unlikely to represent an object. To this end we remove those tracks which have a length smaller than a value l. In this way we exclude very short tracks that are likely to be composed by background boxes that happen to have a high IoU.

Another problem is posed by logos and writings impressed on the video. In fact both of these are very well located by an object proposal but are usually of no interest. To prevent such objects to be considered as valid tracks, we take the mean optical flow magnitude in all the boxes of the track and we discard it if under a threshold s.

Track Ranking It is important to compute a score for temporal proposals, in order to account for the likelihood of objects in such proposal. To this end, we propose to consider two factors: the object proposal score used to generate the bounding boxes at each frame and the values given by the IoUs between frames of the tracks. For the former we define E_t as the mean of the scores given by EdgeBoxes, for the latter we define I_t as the mean of all the IoUs of the frames in the track. Using these two figures we define a track score as:

$$S_t = \lambda E_t \ + \ (1 - \lambda) I_t \tag{5.3}$$

where $\lambda \in [0, 1]$ is a weighting factor used to balance the contributions of the two scores.

5.3 Method Evaluation

Object proposals are usually evaluated measuring how well objects are covered by the generated boxes. These kind of evaluation does not take into account unannotated objects, and therefore provide a benchmark not reflecting the real capabilities of the proposal method.

The method presented in this work is a general framework for discovering salient spatio-temporal tracks in videos, which is built upon a generic bounding box oracle. To evaluate it, we introduce a novel method to establish the effectiveness of a generic video proposal, which is also dataset-independent since it does not rely on annotations. We evaluate whether a proposal effectively represents an instance of some object, since the goal of an object proposal is to locate good candidates and

not to produce the candidate of a given class (i.e. the one of the ground truth). To this end we propose an entropy based evaluation which indicates how the proposal is likely to be recognized as an object. Given a classifier capable of providing for an image a probability distribution $X = \{x_1, \ldots, x_N\}$ over N classes, we compute the Shannon entropy H for the probability vector X, $H(X) = -\sum_{i=1}^{N} x_i \log(x_i)$.

The rationale behind this choice is that, given a good classifier, for a known object the output probability distribution will be high for the relative class and near zero for the others, thus producing a small entropy. On the contrary, for inputs that the classifier is unsure of, e.g. background patches, the output probability will be distributed non-uniformly among all the possible classes, resulting in a higher entropy. Therefore, if the classifier is able to cover effectively a sufficiently large number of classes, then the entropy can be interpreted as a measure of *objectness* for the given proposal.

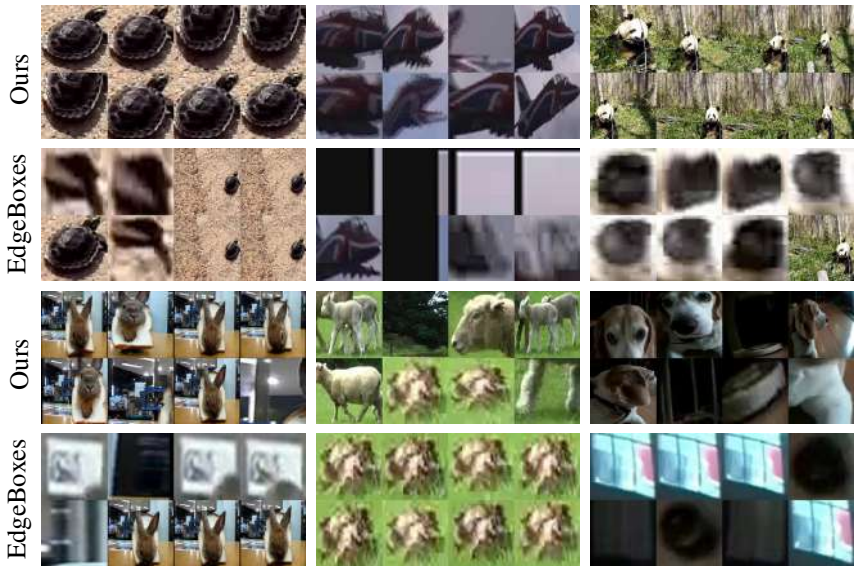

Figure 5.2: Keyframes of the top 8 tracks in the VID dataset, compared with the top 8 EdgeBoxes proposals. Our method has less redundancy and frames objects more clearly.

Method											Mean
EB [165]	5.02	5.19	5.48	4.52	5.92	6.27	6.16	6.54	5.68	5.19	5.60
Ours	3.58	3.25	3.10	2.45	4.02	3.00	3.58	3.25	3.10	2.45	3.18
GT	0.58	1.33	1.03	1.83	2.31	2.41	2.28	2.58	2.57	2.41	1.93

Table 5.1: Entropy comparison (lower is better) between the proposals provided by EdgeBoxes (EB) and our method (Ours) and Ground Truth boxes (GT).

5.4 Experiments

We experiment on the YouTube-Objects (YTO) [111] and on the ILSVRC2015-VID (VID) datasets [119], which both provide a per-frame annotation of the objects. YTO is composed by 10 classes and most videos contain a single object per video. VID instead is a more challenging dataset with 30 classes with multiple objects per video.

Here we evaluate our method using the entropy measure introduced in Section 5.3. In all experiments we use EdgeBoxes [165] as object proposal to generate bounding boxes and as baseline. For the entropy-based proposal scoring we chose the VGG-16 [128] network, trained on the ImageNet [119] dataset as image classifier, yielding a 1000-dimensional output probability vector.

For each video we classify 25 proposals and compute the entropy score. For our method we select the best 25 tracks of each video, according to Eq.5.3, and for each of them we classify, as representative, the box with the best EdgeBoxes score. We compare the entropy scores against the best 25 boxes given by EdgeBoxes for the whole video. As a lower-bound reference value we run the classifier on the dataset ground truth. This value is what can be expected to be obtained when proposing only meaningful objects.

Results for YTO are shown in detail in Table 5.1; it can be seen that our method yields a much lower entropy than EdgeBoxes, also it is close to the ground truth reference. The same trend can be observed on the VID dataset where we measured an average Entropy of 4.73, 3.96 and 3.71 for EdgeBoxes, Our method and the ground truth respectively.

High precision proposals As a further evaluation, we treated our proposal as an object detector measuring the mean Average Precision (mAP) for the YTO dataset. This aims at measuring the precision of a proposal method. Since the class set of YTO is a subset of the one of Pascal VOC [34], for this evaluation we used Fast-

Method											Mean
EB [165]	0.94	0.40	0.49	1.80	10.96	0.57	0.56	0.61	0.26	2.95	0.98
Ours	9.15	7.16	5.98	14.94	10.95	8.43	6.10	2.26	3.42	14.91	8.33

Table 5.2: AP comparison (higher is better) for object detection between EdgeBoxes (EB) and Our method.

RCNN [51], restricted to the ten common classes.

Table 5.2 shows a comparison between our proposed tracks and EdgeBoxes. In order to make the comparison fair, we evaluated the best 25 boxes proposed by both methods for each video, similarly to the entropy evaluation in Section 5.4. The mAP of our tracks is 8.5 times higher than EdgeBoxes, proving that our proposal is much more precise.

Qualitative results We report some qualitative results, showing a comparison of content extracted by our proposal with respect to EdgeBoxes. We compare the best boxes and tracks in a given video. In Fig. 5.2 it can be seen how our proposals are more diverse and frame an object correctly with respect to the top proposal chosen from EdgeBoxes.

In Fig. 5.3 we show an example of high and low entropy proposals. For our method and EdgeBoxes we report the 10 boxes with the lowest and highest entropies among the first best 25 proposals. As reference we also report high and low entropy boxes from the ground truth. It can be seen that our method is more focused on objects even in its highest entropy proposals.

5.5 Conclusions

We proposed a novel and unsupervised method to extract from videos, tracks containing meaningful objects. Our track proposal can build on any object bounding box proposal method. The matching process only relies on bounding box geometry and optical flow, resulting in a simple and effective method for high precision video object proposals. We also introduce a dataset independent method to evaluate the effectiveness of an object proposal, not relying on dataset annotations. The proposal has been evaluated on the YouTube Objects and ILSVRC-2015 VID datasets, showing a high precision and providing meaningful object proposals that can be used for any video analysis task without looking at the whole sequence.

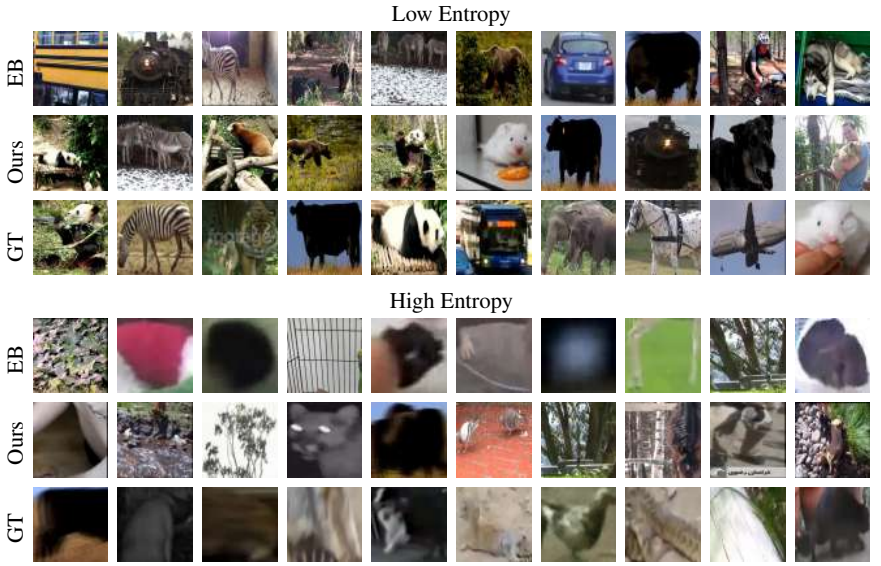

Figure 5.3: Highest and lowest entropy proposals for our method (Ours), Edge-Boxes (EB) and ground truth boxes (GT).

Chapter 6

Predicting action progress in videos

In this chapter we introduce the problem of predicting action progress in videos. We argue that this is an extremely important task because, on the one hand, it can be valuable for a wide range of applications and, on the other hand, it facilitates better action detection results. To solve this problem we introduce a novel approach, named ProgressNet, capable of predicting when an action takes place in a video, where it is located within the frames, and how far it has progressed during its execution. Motivated by the recent success obtained from the interaction of Convolutional and Recurrent Neural Networks, our model is based on a combination of the Faster R-CNN framework, to make framewise predictions, and LSTM networks, to estimate action progress through time. After introducing two evaluation protocols for the task at hand, we demonstrate the capability of our model to effectively predict action progress on the UCF-101 and J-HMDB datasets. Additionally, we show that exploiting action progress it is also possible to improve spatio-temporal localization. [1]

6.1 Introduction

Many human activities and behaviors rely on understanding what actions are taking place in the surrounding environment, to what point they have advanced, and even

[1] Parts of this chapter appear in "Am I Done? Predicting Action Progress in Videos" in *arXiv preprint arXiv:1705.01781*, 2017. [11].

Figure 6.1: Why is action progress prediction important? This example sequence from "Young Frankenstein" movie shows that in most interactive scenarios it is really necessary to be able to accurately predict action progress and react accordingly. *Note: you can watch the sequence on* `https://goo.gl/muYbCb`.

when they might be completed. From simple choices, like crossing the street when cars have passed, to more complex activities like intercepting the ball in a basketball game, an agent has to recognize and understand how far an action has advanced at an early stage, based only on what it has seen so far (Fig. 6.1). If an agent has to act to assist humans, it needs to understand the progress of their intended actions and plan its own actions in real time. It can not wait for the end of the action to perform the visual processing. Therefore, the ultimate goal of action understanding should be the development of an agent equipped with a fully functional perception action loop, from predicting an action before it happens, to following its progress until it ends. This is supported also by experiments showing that humans continuously understand the actions of others in order to plan goals accordingly [44]. Consequently, a model that is able to forecast action progress would enable new applications in robotics (e.g. human-robot interaction, realtime goal definitions) and autonomous driving (e.g. avoid road accidents).

Fully solving the task of action understanding is very hard. The most related literature is that of *predictive vision*, an emerging area which is gaining much interest in the recent years. Approaches have been proposed to perform prediction of the

near future, be that of a learned representation [87, 144], a video frame [94, 145] or directly the action that is going to happen [41]. We believe that fully solving action understanding requires not only to predict the future outcome of an action, but also to fully understand what is observed so far in the progress of an action. As a result, in this chapter we introduce the novel task of predicting *action progress*, *i.e.* the prediction of how far an action has advanced during its execution. In other words, considering a partial observation of some human action, in addition to understanding *what* action and *where* it is happening, we want to infer *how long* this action has been executed for with respect to its expected duration. As a simple example of an application let us consider the use case of a robot trained to interact socially with humans. The correct behaviour to respond to a handshake would be to anticipate, with the right timing, the arm motion, so as to avoid a socially awkward moment in which the person is left hanging. This kind of task cannot be solved unless the progress of the action is known.

Similar tasks have been addressed in the literature. First of all, predicting action progress is conceptually different from action recognition and detection [54, 156, 162], where the focus is on finding *where* the action occurred in time and space. Action completion [64, 90, 152, 160, 163] is a closely related task, where the goal is to predict when an action can be classified as complete, in order to improve temporal boundaries and the classifier accuracy on incomplete sequences. However, this is easier than predicting action progress because it does not require to predict the partial progress of an action.

Action progress prediction is an extremely challenging task since, to be of maximum utility, the prediction should be made while observing the video. While a thick crop of literature addresses action detection and spatio-temporal localization [46, 48, 68, 76, 129, 149, 156, 164], predicting action progress is more related to the variant of online action detection [22, 107, 115]. Here the goal is to accurately detect, as soon as possible, when an action has started and when it has finished but they do not have a model to estimate the progress. The additional information provided by action progress may help in a better definition of temporal boundaries. While advancements in deep learning [62, 81] have largely improved performance for action classification and localization, these are still unsolved problems. We hypothesize that a model for action progress would be forced to embed further knowledge of the rich dynamics of videos, thus improving the temporal understanding of actions.

With this work we propose the first method able to predict action progress using a supervised recurrent neural network fed with convolutional features, and make

three primary contributions:

- We define the new task of action progress prediction, which we believe is useful in developing intelligent planning agents and an experimental protocol to assess performance.
- Our approach is holistic and it is capable of predicting action progress while performing spatio-temporal action detection.
- Interestingly enough, there is a direct benefit in being able to correctly define action boundaries and predicting action progress. In fact, we show that this leads also to improvements in spatio-temporal action detection on untrimmed video benchmarks such as the UCF-101 dataset.

6.2 Predicting Action Progress

In addition to categorizing actions (*action classification*), identifying their boundaries spanning through the video (*action detection*) and localizing the area where they are taking place within the frames (*action localization*), our idea is to learn to predict the progress of an ongoing action. We refer to this task as *action progress prediction*. Additionally, being able to predict to what point the action has advanced, can then help to improve detected spatio-temporal tubes by refining their temporal boundaries.

6.2.1 Problem definition

Given a video $v = \{f_1 \dots f_N\}$ composed of N frames f_i, an action can be represented as a sequence of bounding boxes spanning from a starting frame f_S to an ending frame f_E, and enclosing the subject performing the action. This forms what in literature is usually referred to as *action tube* [54, 122]. For each box in a tube at time t_i, we define the action progress as:

$$p_i = \frac{t_i - t_S}{t_E - t_S} \in [0, 1]. \tag{6.1}$$

Therefore, given a frame f_i within a tube in $[t_S, t_E]$, the action progress can be interpreted as the fraction of the action that has already passed. This definition models the framewise action progress as a linearly and monotonically increasing value in the range $[0, 1]$.

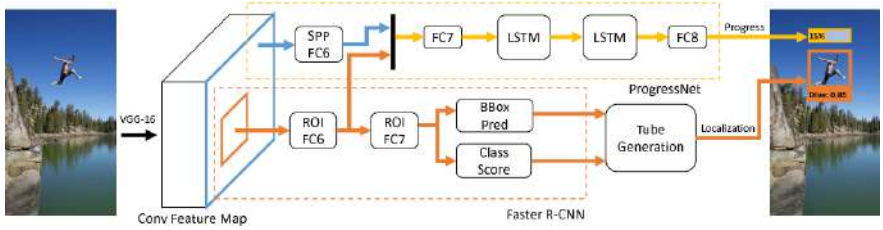

Figure 6.2: Proposed Architecture. On the bottom (highlighted in orange), we show the classification and localization data flows for tube generation. On the top (in yellow), our ProgressNet. Region (ROI FC6) and Contextual features (SPP FC6) from the last convolutional map are concatenated and then fed to a Fully Connected layer (FC7). Two cascaded LSTMs perform action progress prediction.

Although the definition of progress in Eq. 6.1 may be seen as simplistic, a major issue has to be considered in the definition of our task. Formalizing the progress as a more structured prediction task, *i.e.* localizing all action phases, would require a correct and unambiguos definition and annotation of such relevant short-term action boundaries (which is an extremely hard annotation task [126]). On the positive end, a simple linear and monotonic definition of action progress allows to learn predictive models from any dataset containing spatio-temporal boundary annotations. This means that our task does not require to collect any additional annotation to existing datasets, since the action progress values can be directly inferred from the existing temporal annotations.

6.2.2 Model Architecture

The whole architecture of our method is shown in Fig. 6.2, highlighting the first branch dedicated to action classification and localization, and the second branch which predicts action progress. We believe that sequence modelling can have a huge impact on solving the task at hand, since time is a signal that carries a highly informative content. Therefore, we treat videos as ordered sequences and propose a temporal model that encodes the action progress with a Recurrent Neural Network. In particular we use a model with two stacked Long Short-Term Memory layers (LSTM), with 64 and 32 hidden units respectively, plus a final fully connected layer with a sigmoid activation to predict action progress. We named this model *ProgressNet*.

Our model emits a prediction $p_i \in [0, 1]$ at each time step i in an online fash-

ion, with the LSTMs attention windows that keep track of the whole past history, *i.e.* from the beginning of the tube of interest until the current time step. Since actions can be also seen as transformations on the environment [148], we feed the LSTMs with a feature representing regions and their context. We concatenate a contextual feature, computed by spatial pyramid pooling (SPP) of the whole frame [61], with a region feature extracted with ROI Pooling [115]. The two representations are blended with a fully connected layer (FC7). The usage of a SPP layer allows us to encode context information for arbitrarily-sized images. To extract region features and perform detection we build upon the framework recently proposed by Saha *et al.* [122], which is an extension of the popular object detection model Faster R-CNN [115], fine-tuned for action recognition. This model adopts a trained Region Proposal Network (RPN) to generate candidate regions (ROI) that are likely to contain known classes within an image. Using a ROI Pooling layer, these regions are projected onto an intermediate feature map generated by the model and separate predictions are made for each ROI. We use ReLU non linearities after every Fully Connected layer, and dropout to moderate overfitting. In order to be able to feed action tubes to ProgressNet we use the tube generation module from [122], which links bounding boxes across adjacent frames using dynamic programming. Compared to [122], ProgressNet adds a negligible computational footprint of about 1ms per frame, while the whole processing of a single frame takes approximately 150ms on a TITAN X Maxwell GPU.

Tube Refinement. Given an action tube, composed of boxes B_i we predict the sequence of progress values $\widehat{p}_i$ for each box and compute its first order temporal derivative $\widehat{p}_i'$. We trim the tube if $\widehat{p}_i' \leq \delta$ and $\widehat{p}_i \leq \mu_s$ or $\widehat{p}_i' \leq \delta$ and $\widehat{p}_i \geq \mu_e$ where δ is used to find when the derivative comes close to zero (*i.e.* the action is not progressing) and μ_s and μ_e ensure that the action is about to begin or has reached a sufficiently far point in its execution. We use $\delta = 0.05$, $\mu_s = 0.1$ and $\mu_e = 0.8$, which have proven to be suitable by cross validation.

6.2.3 Learning

Our spatio-temporal localization network is initialized with the pretrained network from [122], which is based on the VGG-16 architecture [128]. To train our ProgressNet we use positive ground truth tubes as training samples. To avoid overfitting the network, we apply the two following augmentation strategies. First, for every tube we randomly pick a starting point and a duration, so as to generate a shorter or equal tube (keeping the same ground truth progress values in the chosen

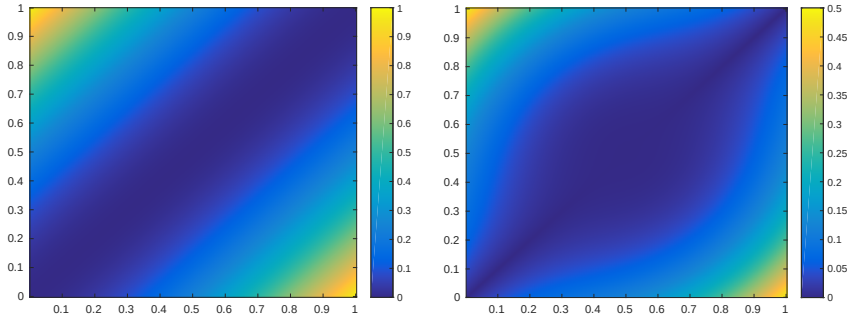

Figure 6.3: Comparison between the $L2$ (left) and the Boundary Observant (right) loss functions. Predicted values and ground truth targets are on the two axes. It can be seen that the Boundary Observant loss is stricter against errors on the action boundaries.

interval). For instance, if the picked starting point is at the middle of the video and the duration is half the video length, the corresponding ground truth progress targets would be the values in the interval $[0.5, 1.0]$. This also force the model to not assume 0 as starting progress value. Second, for every tube we generate a sub-sampled version, reducing the frame rate by a random factor uniformly distributed in $[1, 10]$. This second strategy helps in generalizing with respect to the speed of execution of different instances of the same action class.

Action progress is defined using Eq. 6.1. Since one of our goals is to improve action boundaries, we encourage the network to be more precise on the temporal boundaries of an action by using our Boundary Observant loss:

$$L = \frac{1}{N} \sum_{i}^{N} \left\{ \left[(p_i - 0.5)^2 + (\widehat{p}_i - 0.5)^2 \right] |p_i - \widehat{p}_i| \right\} \tag{6.2}$$

where $\widehat{p}_i$ and p_i are the predicted progress and the correspondent ground truth value, respectively. Compared to a standard $L2$ loss for regression, the Boundary Observant loss penalizes errors on the action boundaries more than in intermediate parts, since we want to precisely identify when the action starts and ends. At the same time, it avoids the trivial solution of always predicting the intermediate value $\widehat{p} = 0.5$. Fig. 6.3 shows the difference between the two loss functions, where predicted values are on the x axis and desired values on the y axis. Note from Eq. 6.1 that in action progress prediction only values in $[0, 1]$ can be expected

We initialize all layers of ProgressNet with the Xavier [55] method and employ

the Adam [78] optimizer with a learning rate of 10^{-4}. We use dropout with a probability of 0.5 on the fully connected layers.

6.3 Experiments

In this section we show results on action progress prediction for spatio-temporal tubes and propose two evaluation protocols. Since this is a novel task we introduce some simple baselines and show the benefits of our approach. To further underline the importance of predicting action progress, we exploit our predictions to refine precomputed action tubes and improve their temporal localization.

We experiment on the J-HMDB [72] and UCF-101 [131] datasets. J-HMDB consists of 21 action classes and 928 videos, annotated with body joints from which spatial boxes can be inferred. All the videos are temporally trimmed and contain only one action. We use this dataset to benchmark action progress prediction. UCF-101 contains 24 classes annotated for spatio-temporal action localization. It is a more challenging dataset because actions are temporally untrimmed and there can be more than one action of the same class per video. Moreover, it contains video sequences with large variation in appearance, scale and illumination. We use this dataset to predict action progress but also to measure the improvement in action localization. In order to be comparable with previous spatio-temporal action localization works, we adopt the same split of UCF-101 used in [107, 122, 149, 158]. Note that larger datasets such as THUMOS [69] and ActivityNet [35] do not provide bounding box annotations, and therefore can not be used in our setting.

6.3.1 Evaluation protocol and Metrics

In order to evaluate the task of action progress prediction, we introduce two evaluation protocols along with standard Video-AP for spatio-temporal localization.

Framewise Mean Squared Error. This metric tells how well the model behaves at predicting action progress when the spatio-temporal coordinates of the actions are known. Test data is evaluated frame by frame by taking the predictions $\widehat{p}_i$ on the ground truth boxes B_i and comparing them with action progress targets p_i. We compute mean squared error $MSE = ||\widehat{p}_i - p_i||^2$ across each class. Being computed on ground truth boxes, this metric assumes perfect detections and thus disregards the action detection task, only evaluating how well progress prediction works.

Average Progress Precision. Average Progress Precision (APP) is identical to framewise Average Precision (Frame-AP) [54] with the difference that true positives must have a progress that lays within a margin from the ground truth target. Frame-AP measures the area under the precision-recall curve for the detections in each frame. A detection is considered a hit if its Intersection over Union (IoU) with the ground truth is bigger than a threshold τ and the class label is correct. In our case, we fix $\tau = 0.5$ and evaluate the results at different progress margins m in $[0, 1]$. A predicted bounding box $\widehat{B}_i$ is matched with a ground truth box B_i and considered a true positive when B_i has not been already matched and the following conditions are met:

$$IoU(\widehat{B}_i, B_i) \geq \tau, \quad |\widehat{p}_i - p_i| \leq m \tag{6.3}$$

where $\widehat{p}_i$ is the predicted progress, p_i is the ground truth progress and m is the progress margin. We compute Average Progress Precision for each class and report a mean (mAPP) value for a set of m values.

Video-AP. Video Average Precision (Video-AP) [54] is used in order to compare with previous methods and show how predicting action progress can impact on spatio-temporal action localization. We report results at varying IoU thresholds, ranging from 0.05 to 0.6. Note that IoU is computed over spatio-temporal tubes, so detected tubes must be precise both at locating the action within the frame and at finding the correct temporal boundaries. Therefore, differently from detection in static images, IoUs higher than 0.4 are very strict.

6.3.2 Implementation Details

In practice, we observed that on some classes of the UCF-101 dataset it is hard to learn accurate progress prediction models. These are action classes like *Biking* or *WalkingWithDog* that exhibit a cyclic behavior, where even for a human observer is hard to guess how far the action has progressed. Therefore we extended our framework by adopting a curriculum learning strategy. First, the model is trained as described in Sect. 6.2.3 on classes that have a clear non-cyclic behavior[2]. Then, we fix all convolutional, FC and LSTM layers and fine-tune the FC8 layer that is used to perform progress prediction from the last LSTM output on the whole UCF-101 dataset. This strategy improves the convergence of the model.

[2]This subset consists of the following classes: *Basketball*, *BasketballDunk*, *CliffDiving*, *CricketBowling*, *Diving*, *FloorGymnastics*, *GolfSwing*, *LongJump*, *PoleVault*, *TennisSwing*, *VolleyballSpiking*.

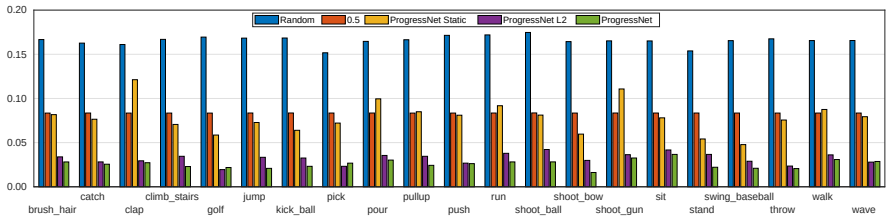

Figure 6.4: Mean Squared Error obtained by three formulations of our model (ProgressNet Static, ProgressNet L2 and ProgressNet) on the J-HMDB dataset. Random and constant 0.5 predictions are reported as reference.

6.3.3 Action Progress Prediction

In this section we report the experimental results on the task of predicting action progress. Since ProgressNet is a multitask approach, we first measure action progress performance on perfectly localized actions. We then test the method on real detected tubes with the full model and finally report a qualitative analysis which shows some success and failure cases.

Action progress on correct localized actions. In this first experiment we evaluate the ability of our method to predict action progress on correctly localized actions in both time and space. We take the ground truth tubes of actions on the test set and compare the MSE of three variants of our method: the full architecture trained with our Boundary Observant loss (*ProgressNet*), the same model trained with L2 loss (*ProgressNet L2*) and a reduced memoryless variant (*ProgressNet Static*).

The comparison of our full model against ProgressNet L2 is useful to understand the contribution of the Boundary Observant loss with respect to a simpler L2 loss. To underline the importance of using recurrent networks in action progress prediction, in the variant ProgressNet Static we substitute the two LSTMs with two fully connected layers that predict progress framewise. In addition, we provide two baselines: random prediction and a constant prediction of the progress expectation. The random prediction provides us with a higher bound on the MSE values. The latter, with a prediction of $\widehat{p} = 0.5$ for every frame, is a trivial solution that obtains good MSE results. Both are clearly far from being informative for the task.

We report the results obtained in this experiment in Table 6.1. We first observe that the MSE values are consistent among the two datasets, with ProgressNet models ahead of the baselines. ProgressNet and ProgressNet L2 obtain a much lower

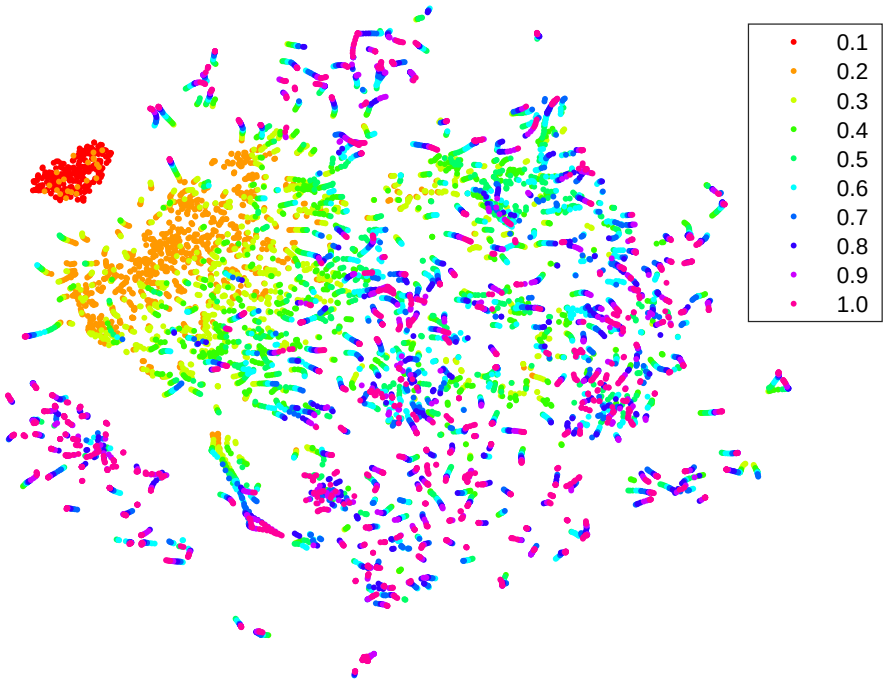

Figure 6.5: t-SNE visualization of ProgressNet's hidden states of the second LSTM layer on the J-HMDB test set. Each point corresponds to a frame and its color represents the ground truth progress quantized with a 0.1 granularity.

	J-HMDB	UCF-101
Random	0.166	0.166
0.5	0.084	0.083
ProgressNet Static	0.079	0.104
ProgressNet L2	0.032	0.052
ProgressNet	**0.026**	**0.050**

Table 6.1: Mean Square Error values for action progress prediction on the UCF-101 and J-HMDB datasets. Results are averaged among all classes.

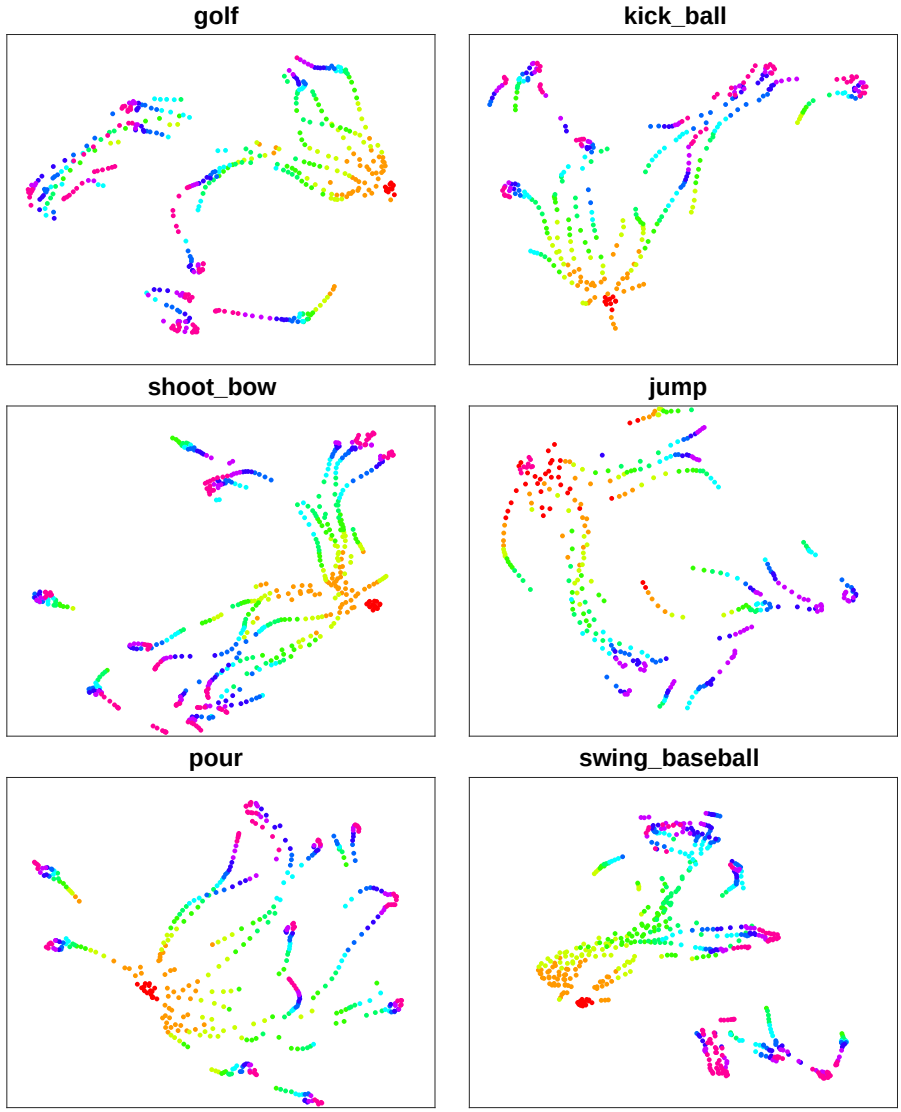

Figure 6.6: t-SNE visualization of ProgressNet's hidden states from the second LSTM layer of 6 classes from J-HMDB test set. Each point corresponds to a frame and its color represents the ground truth progress quantized with a 0.1 granularity (as in Fig. 6.5).

error than ProgressNet Static and the baselines. This confirms the ability of our model to understand action progress. In particular, the best result is obtained with ProgressNet, proving that our Boundary Observant loss plays an important role in training the network effectively. ProgressNet Static has an inferior MSE than the variants with memory, suggesting that single frames for some classes can be ambiguous and a temporal context can help to accurately predict action progress. In particular, observing the class breakdown for J-HMDB in Fig. 6.4, we note that the static model gives better MSE values for some actions such as *Swing Baseball* and *Stand*. This is due to the fact that such actions have clearly identifiable states, which help to recognize the development of the action. On the other hand, classes such as *Clap* and *Shoot Gun* are hardly addressed with models without memory because they exhibit only few key poses that can reliably establish the progress.

We also report in Fig. 6.5 and 6.6 embeddings of the hidden states of the second LSTM layer. The former contains the whole test set while the latter focuses on four classes. Each point is a frame of the test set of J-HMDB and is colored according to its true action progress. In all figures we note that progress increases radially along trajectories from points labeled with 0.1. This suggests that ProgressNet has learned directions in the hidden state space that follow action progress.

Action progress with the full pipeline. In this second experiment, we evaluate the performance of action progress while also performing the spatio-temporal action detection with the entire pipeline. Differently from the previous experiment, we test the full approach where action tubes are generated by the detector. In Fig. 6.7, we report the mAPP of ProgressNet (trained with BO loss), ProgressNet Static and the two baselines Random and 0.5 on both the UCF-101 and J-HMDB benchmarks. Note that the mAPP upper bound is given by standard mean Frame-AP [54], which is equal to mAPP with margin $m = 1$. In the $\widehat{p} = 0.5$ baseline, this upper bound is reached with $m = 0.5$.

It can be seen that ProgressNet has mAPP higher than the baselines for stricter progress margin. This confirms that our approach is able to predict action progress correctly even when tubes are noisy such as those generated by a detector. ProgressNet Static exhibits a lower performance than ProgressNet, confirming again that the memory is helpful to model action progress.

Qualitative analysis. Qualitative results taken from the UCF-101 dataset, obtained by our ProgressNet, are shown in Fig. 6.8. It is interesting to notice how in some of the examples the predicted progress does not have a decise linear trend,

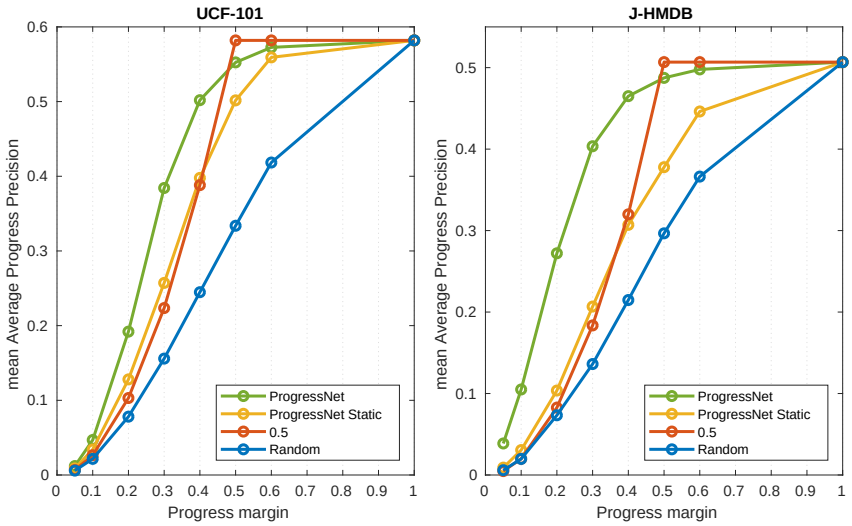

Figure 6.7: mAPP on the UCF-101 and J-HMDB datasets.

but follows the visual appearance of the action. In particular in the second row (*LongJump*), while the athlete is running, the output values tend to fluctuate, but as soon as he jumps the predicted progress grows firmly towards one. Insights on the model behavior are also given by the failure cases: in the *GolfSwing* clip, the actor hesitates before hitting the golf ball and the progress is therefore late respect to the ground truth. In the *FloorGymnastics* case instead, the first and the last pose are almost identical and the predicted progress grows straight towards completion after a few frames.

6.3.4 Spatio-temporal action localization

Besides having an intrinsic value in the understanding of human behavior in videos, action progress is also a useful tool for obtaining more precise action tubes. The reason why action tubes generated by machine learning methods are often imprecise can be traced back to two main causes: inaccurate frame level detections and difficulties in properly identifying the temporal action boundaries, *i.e.* the first and the last frame in which the action is present. Since most methods rely on powerful and precise detectors [107, 122] to generate candidate boxes, we argue that the primary cause of defects concerns temporal boundaries.

In this experiment we perform action localization of tubes trimmed with the

IoU	0.05	0.1	0.2	0.3	0.4	0.5	0.6
Ours	**79.9**	**77.0**	**67.0**	**55.6**	**46.4**	**35.7**	**26.9**
Saha *et al.* [122][3]	78.9	76.1	66.4	54.9	45.2	34.8	25.9
Yu *et al.* [158]	42.8	-	-	-	-	-	-
Weinzaepfel *et al.* [149]	54.3	**51.7**	46.8	**37.8**	-	-	-
Peng *et al.* [107]	**54.5**	50.4	42.3	32.7	-	-	-
Singh *et al.* [129]	-	-	73.5	-	-	46.3	
Kalogeiton *et al.* [76]	-	-	**77.2**	-	-	**51.4**	

Table 6.2: Mean Video-AP on UCF-101 (*split1*) for spatio-temporal action localization.

strategy described in Sect. 6.2.2. We follow previous work [76, 122, 129] and use UCF-101 (*split1*), reporting performance in terms of mean Video-AP. We apply our trimming strategy to tubes generated with [122] [3], but the proposed approach is applicable to tubes generated by any generic method[4]. We report the results for this experiment in Table 6.2. Comparing our approach to the baseline tubes generated with [122], we obtain higher mean Video-AP for every IoU threshold. This confirms that action progress can be useful to better localize actions in time.

6.4 Conclusion

In this work we defined the novel task of action progress prediction. Our method is the first that can predict spatio-temporal localization of actions and at the same time understand the evolution of such activities by predicting their stage on-line. This approach opens new scenarios for any goal planning intelligent agent in interaction applications. In addition to our model, we propose a boundary observant loss which helps to avoid trivial solutions. Moreover, we show that action progress can be used to improve action temporal boundaries.

[3] Please note that these numbers are slightly different with respect to the results reported in the original paper [122] (apparently there was a bug in the code). We report in Table 6.2 the updated results by the authors, as shown in `https://bitbucket.org/sahasuman/bmvc2016_code`.

[4] State of the art methods like Singh *et al.* [129] and Kalogeiton *et al.* [76], only recently appeared, are likely to generate better tubes.

Figure 6.8: Qualitative results (success cases in the green frame, failure cases in the red one). Each row represents the progression of an action. Progress values are plotted inside the detection box with time on the x axis and progress values on the y axis. Progress targets are shown in green and predicted progresses in blue.

Chapter 7

Conclusion

7.1 Summary of contribution

In this thesis we outlined different contributions to the topic of scene understanding, with a particular attention to object and action detection beyond simple categorical annotations. The first part of the thesis focused on a taxonomy learning algorithm to index ensembles of Exemplar-SVMs and permit an extremely fast evaluation. This results in the practical applicability of the method when using large ensembles. This is of particular interest for the ability to perform label transfer operations and augment detections with meta-data inferred from the training samples even for time constrained applications. Combining our hierarchical data structure with speed-up strategy such as early rejection an vector quantization we are able to provide up to 100 fold speed gains for large ensembles with only a small loss in accuracy.

Moving to the video domain, we presented two different techniques to tackle two kinds of tasks. At first we presented an unsupervised algorithm for object discovery. This method exploits the temporal consistency of object proposals to generate spatio-temporal tracks with objects of interest. Since the method does not rely on any kind of label, the algorithm could even discover objects that are not annotated in the dataset. To evaluate it we proposed an entropy based criterion that proved to be suitable to establish whether the proposed tracks are likely to contain objects or not.

Finally, we presented a technique for spatio-temporal action localization in videos. Unlike current methods in the state of the art, which classify and localize the action in time and space, we also predict how far the action has progressed. We refer to this task as *Action Progress Prediction*. We argue that this task will

have a large applicability spectrum: accurate action localization, action prediction, interaction with intelligent agents and planning, among others. Our preliminary experiments in this novel research direction proved to be effective and helpful for obtaining accurate spatio-temporal detections.

In addition, we also reported our experience with the development of the Android App *Imaging 900*, a computer vision application for the Museo Novecento in Florence, Italy. This App allows users to scan paintings with the phone and transfer its artistic style onto a personal picture, resulting in a higher engagement from the visitor.

7.2 Directions for future work

As future work we plan to extend some of the presented methods in the video domain, particularly for action recognition. At first we will apply our object discovery technique specifically to the action domain, with the aim of generating action tubes in an unsupervised fashion. This can be used as a preliminary step to help action detection algorithms with unsupervised action proposals. Given the effectiveness of the method for discovering objects, we believe that the technique could also be exploited for discovering actions.

The other part we intend to explore in more detail is action progress prediction. What emerged from our research is that in certain cases the formalization of a concept as action progress may not be straightforward and could require a careful study. Certain actions are in fact characterized by a cyclic or an erratic behaviour and are not representable as a simple linear progression. Different speeds of subactions could require to split the action into simpler atomic phases to capture the execution variability in different sequences. In order to follow this idea though, a big amount of fine-grained annotations would be needed, which are hard to obtain. A solution to this issue could be to apply an unsupervised strategy to align actions and automatically discover their atomic components.

Appendix A

Publications

This research activity has led to several publications in international journals and conferences. These are summarized below.[1]

International Journals

1. **F. Becattini**, L. Seidenari, A. Del Bimbo. "Indexing quantized ensembles of exemplar-SVMs with rejecting taxonomies", in *Multimedia Tools and Applications*, vol. in press, 2017. (Special Issue: Content Based Multimedia Indexing) [10.1007/s11042-017-4794-7]

International Conferences and Workshops

1. **F. Becattini**, L. Seidenari, A. Del Bimbo. "Indexing ensembles of exemplar-svms with rejecting taxonomies", in *Proc. of Content-Based Multimedia Indexing (CBMI)*, Bucharest (Romania), 2016.
2. G. Cuffaro, **F. Becattini**, C. Baecchi, L. Seidenari, A. Del Bimbo. "Segmentation Free Object Discovery in Video", in *Proc. of European Computer Vision Conference (ECCV), Workshop on Brave new ideas for motion representations in videos*, Amsterdam (Netherlands), 2016.
3. **F. Becattini**, A. Ferracani, L. Landucci, D. Pezzatini, T. Uricchio, A. Del Bimbo. "Imaging Novecento. A Mobile App for Automatic Recognition of Artworks and Transfer of Artistic Styles", in *Proc. of International Euro-Mediterranean Conference (EUROMED)*, Nicosia (Cyprus), 2016. **(Best paper award)**

[1] The author's bibliometric indices are the following: H-index = 1, total number of citations = 4 (source: Google Scholar on September 26, 2017).

Under submission

1. **F. Becattini**, T. Uricchio, L. Ballan, L. Seidenari, A. Del Bimbo. “Am I Done? Predicting Action Progress in Videos”, *Computer Vision and Patter Recognition*, 2018, Salt Lake City (Utah).

Bibliography

[1] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: a review," *ACM Computing Surveys*, vol. 43, no. 3, pp. 16:1–16:43, 2011.

[2] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *Proceedings of Computer Vision and Pattern Recognition*, 2010.

[3] A. G. Anderson, C. P. Berg, D. P. Mossing, and B. A. Olshausen, "Deepmovie: Using optical flow and deep neural networks to stylize movies," *arXiv preprint arXiv:1605.08153*, 2016.

[4] R. M. Anwer, F. S. Khan, J. van de Weijer, and J. Laaksonen, "Combining holistic and part-based deep representations for computational painting categorization," in *Proceedings of the 2016 ACM International Conference on Multimedia Retrieval*. ACM, 2016, pp. 339–342.

[5] M. Aubry, D. Maturana, A. A. Efros, B. C. Russel, and J.Sivic, "Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models," in *Proceedings of Computer Vision and Pattern Recognition*, 2014.

[6] Y. Aytar and A. Zisserman, "Enhancing exemplar svms using part level transfer regularization." in *British Machine VIsion Conference*, 2012, pp. 1–11.

[7] R. Ballagas, M. Rohs, J. G. Sheridan, and J. Borchers, "Byod: Bring your own device," in *Proceedings of the Workshop on Ubiquitous Display Environments, Ubicomp*, vol. 2004, 2004.

[8] L. Baraldi, F. Paci, G. Serra, L. Benini, and R. Cucchiara, "Gesture recognition using wearable vision sensors to enhance visitors museum experiences," *IEEE Sensors Journal*, vol. 15, no. 5, pp. 2705–2714, 2015.

[9] F. Becattini, L. Seidenari, and A. Del Bimbo, "Indexing ensembles of exemplar-svms with rejecting taxonomies," in *Content-Based Multimedia Indexing (CBMI), 2016 14th International Workshop on*. IEEE, 2016, pp. 1–6.

[10] ——, "Indexing quantized ensembles of exemplar-svms with rejecting taxonomies," *Multimedia Tools and Applications*, pp. 1–22, 2017.

[11] F. Becattini, T. Uricchio, L. Ballan, L. Seidenari, and A. Del Bimbo, "Am i done? predicting action progress in videos," *arXiv preprint arXiv:1705.01781*, 2017.

[12] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian detection at 100 frames per second," in *Proceedings of Computer Vision and Pattern Recognition*, 2012.

[13] S. Bengio, J. Weston, and D. Grangier, "Label embedding trees for large multi-class tasks," in *Proceedings of NIPS*, 2010.

[14] L. Bourdev and J. Brandt, "Robust object detection via soft cascade," in *Proceedings of Computer Vision and Pattern Recognition*, 2005.

[15] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles, "SST: Single-stream temporal action proposals," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[16] C.-Y. Chen, B. R. Chang, and P.-S. Huang, "Multimedia augmented reality information system for museum guidance," *Personal and ubiquitous computing*, vol. 18, no. 2, pp. 315–322, 2014.

[17] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proceedings of Computer Vision and Pattern Recognition*, 2014.

[18] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[19] E. Crowley and A. Zisserman, "The state of the art: Object retrieval in paintings using discriminative regions." in *British Machine VIsion Conference*, 2014.

[20] G. Cuffaro, F. Becattini, C. Baecchi, L. Seidenari, and A. Del Bimbo, "Segmentation free object discovery in video," in *Computer Vision–European Conference on Computer Vision 2016 Workshops*. Springer, 2016, pp. 25–31.

[21] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of Computer Vision and Pattern Recognition*, 2005, pp. 886–893.

[22] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. G. Snoek, and T. Tuytelaars, "Online action detection," in *European Conference on Computer Vision*, 2016, pp. 269–284.

[23] S. de Sousa Borges, V. H. Durelli, H. M. Reis, and S. Isotani, "A systematic mapping on gamification applied to education," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*. ACM, 2014, pp. 216–222.

[24] T. Dean, M. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik, "Fast, accurate detection of 100,000 object classes on a single machine," in *Proceedings of Computer Vision and Pattern Recognition*, Washington, DC, USA, 2013.

[25] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam, "Large-scale object classification using label relation graphs," in *European Conference on Computer Vision*. Springer, 2014, pp. 48–64.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. Computer Vision and Pattern Recognition 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[27] J. Deng, S. Satheesh, A. C. Berg, , and L. Fei-Fei, "Fast and balanced: Efficient label tree learning for large scale object recognition," in *Proceedings of NIPS*, 2011.

[28] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west," in *Proceedings of British Machine VIsion Conference*, 2010.

[29] C. Dubout and F. Fleuret, "Exact acceleration of linear object detectors," in *Proceedings of European Conference on Computer Vision*, 2012, Oral, pp. 301–311.

[30] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. ACM, 2001, pp. 341–346.

[31] A. Elgammal and C.-S. Lee, "Separating style and content on a nonlinear manifold," in *Computer Vision and Pattern Recognition, 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004, pp. I–478.

[32] V. Escorcia, F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Daps: Deep action proposals for action understanding," in *European Conference on Computer Vision*. Springer, 2016, pp. 768–784.

[33] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results."

[34] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[35] B. G. Fabian Caba Heilbron, Victor Escorcia and J. C. Niebles, "ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 961–970.

[36] J. Fan, N. Zhou, J. Peng, and L. Gao, "Hierarchical learning of tree classifiers for large-scale plant species identification," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 4172–4184, 2015.

[37] B. Fanini, E. d'Annibale, E. Demetrescu, D. Ferdani, and A. Pagano, "Engaging and shared gesture-based interaction for museums the case study of k2r international expo in rome," in *2015 Digital Heritage*, vol. 1. IEEE, 2015, pp. 263–270.

[38] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1933–1941.

[39] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, Sep. 2010.

[40] P. F. Felzenszwalb, R. B. Girshick, and D. A. McAllester, "Cascade object detection with deformable part models," in *Proceedings of Computer Vision and Pattern Recognition*, 2010, pp. 2241–2248.

[41] C. Fermüller, F. Wang, Y. Yang, K. Zampogiannis, Y. Zhang, F. Barranco, and M. Pfeiffer, "Prediction of manipulation actions," *International Journal of Computer Vision*, pp. 1–17, 2016.

[42] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars, "Modeling video evolution for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5378–5387.

[43] A. Fineschi and A. Pozzebon, "A 3d virtual tour of the santa maria della scala museum complex in siena, italy, based on the use of oculus rift hmd," in *3D Imaging (IC3D), 2015 International Conference on*. IEEE, 2015, pp. 1–5.

[44] J. R. Flanagan and R. S. Johansson, "Action plans used in action observation," *Nature*, vol. 424, no. 6950, p. 769, 2003.

[45] A. Gaidon, Z. Harchaoui, and C. Schmid, "Actom sequence models for efficient action detection," in *Computer Vision and Pattern Recognition, 2011 IEEE Conference on*. IEEE, 2011, pp. 3201–3208.

[46] ——, "Temporal localization of actions with actoms," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2782–2795, 2013.

[47] ——, "Temporal localization of actions with actoms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 11, pp. 2782–2795, 2013.

[48] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia, "Turn tap: Temporal unit regression network for temporal action proposals," in *IEEE International Conference on Computer Vision*, 2017.

[49] T. Gao and D. Koller, "Discriminative learning of relaxed hierarchy for large-scale visual recognition," in *Proceedings of International Conference on Computer Vision*, 2011.

[50] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, 2015.

[51] R. Girshick, "Fast r-cnn," in *Proceedings of International Conference on Computer Vision*, 2015.

[52] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[53] ——, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 142–158, 2016.

[54] G. Gkioxari and J. Malik, "Finding action tubes," in *IEEE International Conference on Computer Vision*, 2015, pp. 759–768.

[55] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks." in *Artificial Intelligence and Statistics Conference*, 2010, pp. 249–256.

[56] G. Griffin and P. Perona, "Learning and using taxonomies for fast visual categorization," in *Computer Vision and Pattern Recognition, 2008. Computer Vision and Pattern Recognition 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[57] P. Gronat, G. Obozinski, J. Sivic, and T. Pajdla, "Learning and calibrating per-location classifiers for visual place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 907–914.

[58] M. Guillaumin and V. Ferrari, "Large-scale knowledge transfer for object localization in imagenet," in *Computer Vision and Pattern Recognition (Computer Vision and Pattern Recognition), 2012 IEEE Conference on*. IEEE, 2012, pp. 3202–3209.

[59] B. Hariharan, J. Malik, and D. Ramanan, "Discriminative decorrelation for clustering and classification," in *Computer Vision–European Conference on Computer Vision 2012*. Springer, 2012, pp. 459–472.

[60] H. Harzallah, F. Jurie, and C. Schmid, "Combining efficient object localization and image classification," in *Proceedings of International Conference on Computer Vision*, 2009.

[61] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *European Conference on Computer Vision*, 2014, pp. 346–361.

[62] ——, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[63] Z. He, B. Cui, W. Zhou, and S. Yokoi, "A proposal of interaction system between visitor and collection in museum hall by ibeacon," in *Computer Science & Education (ICCSE), 2015 10th International Conference on*. IEEE, 2015, pp. 427–430.

[64] F. Heidarivincheh, M. Mirmehdi, and D. Damen, "Beyond action recognition: Action completion in rgb-d data," in *British Machine Vision Conference*, 2016.

[65] F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Fast temporal activity proposals for efficient detection of human actions in untrimmed videos," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1914–1923.

[66] M. Hoai and F. De la Torre, "Max-margin early event detectors," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 191–202, 2014.

[67] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, pp. 814–830, 2016.

[68] R. Hou, C. Chen, and M. Shah, "Tube convolutional neural network (t-cnn) for action detection in videos," in *IEEE International Conference on Computer Vision*, 2017.

[69] H. Idrees, A. R. Zamir, Y.-G. Jiang, A. Gorban, I. Laptev, R. Sukthankar, and M. Shah, "The THUMOS challenge on action recognition for videos "in the wild"," *Computer Vision and Image Understanding*, vol. 155, pp. 1–23, 2017.

[70] A. Jain, A. Gupta, M. Rodriguez, and L. S. Davis, "Representing videos using mid-level discriminative patches," in *Proceedings of Computer Vision and Pattern Recognition*, 2013.

[71] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Computer Vision and Pattern Recognition, 2010 IEEE Conference on*. IEEE, 2010, pp. 3304–3311.

[72] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3192–3199.

[73] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM MM*. ACM, 2014, pp. 675–678.

[74] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," *arXiv preprint arXiv:1603.08155*, 2016.

[75] L. Johnson, S. Adams Becker, V. Estrada, and A. Freeman, *The NMC Horizon Report: 2015 Museum Edition.* ERIC, 2015.

[76] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid, "Action tubelet detector for spatio-temporal action localization," in *IEEE International Conference on Computer Vision*, 2017.

[77] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller, "Recognizing image style," *arXiv preprint arXiv:1311.3715*, 2013.

[78] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[79] T. Kobayashi, "Three viewpoints toward exemplar svm," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2765–2773.

[80] Y. Kong, Z. Tao, and Y. Fu, "Deep sequential context networks for action prediction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1473–1481.

[81] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[82] D. Kuettel, M. Guillaumin, and V. Ferrari, "Segmentation propagation in imagenet," in *European Conference on Computer Vision*. Springer, 2012, pp. 459–473.

[83] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid, "Unsupervised object discovery and tracking in video collections," in *Proceedings of International Conference on Computer Vision*, 2015.

[84] J. E. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg, "State of the "art": A taxonomy of artistic stylization techniques for images and video," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 5, pp. 866–885, 2013.

[85] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *Proceedings of Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[86] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Computer Vision and Pattern Recognition, 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[87] H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang, "Unsupervised representation learning by sorting sequences," in *IEEE International Conference on Computer Vision*, 2017.

[88] B. Liu, F. Sadeghi, M. Tappen, O. Shamir, and C. Liu, "Probabilistic label trees for efficient large scale image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 843–850.

[89] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[90] S. Ma, L. Sigal, and S. Sclaroff, "Learning activity progression in lstms for activity detection and early detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1942–1950.

[91] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *Proceedings of International Conference on Computer Vision*, 2011.

[92] L. Malomo, F. Banterle, P. Pingi, F. Gabellone, and R. Scopigno, "Virtualtour: a system for exploring cultural heritage sites in an immersive way," in *2015 Digital Heritage*, vol. 1. IEEE, 2015, pp. 309–312.

[93] M. Marszałek and C. Schmid, "Constructing category hierarchies for visual recognition," in *Proceedings of European Conference on Computer Vision*. Springer, 2008.

[94] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *arXiv preprint arXiv:1511.05440*, 2015.

[95] R. E. Mayer and R. Moreno, "Nine ways to reduce cognitive load in multimedia learning," *Educational Psychologist*, vol. 38, no. 1, pp. 43–52, 2003.

[96] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[97] D. Modolo, A. Vezhnevets, and V. Ferrari, "Context forest for object class detection," *Arxiv preprint*, 2015.

[98] D. Modolo, A. Vezhnevets, O. Russakovsky, and V. Ferrari, "Joint calibration of ensemble of exemplar svms," in *Computer Vision and Pattern Recognition (Computer Vision and Pattern Recognition), 2015 IEEE Conference on*. IEEE, 2015, pp. 3955–3963.

[99] D. Mrowca, M. Rohrbach, J. Hoffman, R. Hu, K. Saenko, and T. Darrell, "Spatial semantic regularisation for large scale object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2003–2011.

[100] S. a. A. Negrusa, Toader, "Exploring gamification techniques and applications for sustainable tourism," *Sustainability*, vol. 7, no. 8, p. 11160, 2015.

[101] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proceedings of NIPS*, 2002.

[102] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *European conference on computer vision*. Springer, 2010, pp. 392–405.

[103] D. Oneata, J. Revaud, J. Verbeek, and C. Schmid, "Spatio-temporal object detection proposals," in *Proceedings of European Conference on Computer Vision*. Springer, 2014.

[104] A. Pagano, G. Armone, and E. De Sanctis, "Virtual museums and audience studies: the case of keys to rome exhibition," in *2015 Digital Heritage*, vol. 1. IEEE, 2015, pp. 373–376.

[105] S. Park, "The effects of social cue principles on cognitive load, situational interest, motivation, and achievement in pedagogical agent multimedia learning," *Journal of Educational Technology & Society*, no. 4, pp. 211–229, 2015.

[106] K.-C. Peng and T. Chen, "Cross-layer features in convolutional neural networks for generic classification tasks," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 3057–3061.

[107] X. Peng and C. Schmid, "Multi-region two-stream R-CNN for action detection," in *European Conference on Computer Vision*, 2016, pp. 744–759.

[108] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Computer Vision and Pattern Recognition, 2007. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[109] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," *Computer Vision–European Conference on Computer Vision 2010*, pp. 143–156, 2010.

[110] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, vol. 28, no. 6, pp. 976–990, 2010.

[111] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *Proceedings of Computer Vision and Pattern Recognition*. IEEE, 2012.

[112] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (Computer Vision and Pattern Recognition)*, June 2016.

[113] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.

[114] T. Reiners and L. C. Wood, Eds., *Gamification in Education and Business*. Springer International Publishing, 2015.

[115] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[116] M. Ristin, M. Guillaumin, J. Gall, and L. Gool, "Incremental learning of ncm forests for large-scale image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3654–3661.

[117] M. Rohrbach, S. Ebert, and B. Schiele, "Transfer learning in a transductive setting," in *Advances in neural information processing systems*, 2013, pp. 46–54.

[118] B. Ruf, E. Kokiopoulou, and M. Detyniecki, "Mobile museum guide based on fast sift recognition," in *International Workshop on Adaptive Multimedia Retrieval*. Springer, 2008, pp. 170–183.

[119] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[120] M. A. Sadeghi and D. Forsyth, "Fast template evaluation with vector quantization," in *Proceedings of NIPS*, 2013, pp. 2949–2957.

[121] M. Sadeghi and D. Forsyth, "30hz object detection with dpm v5," in *Proceedings of European Conference on Computer Vision*, ser. Lecture Notes in Computer Science, vol. 8689. Springer International Publishing, 2014, pp. 65–79.

[122] S. Saha, G. Singh, M. Sapienza, P. H. Torr, and F. Cuzzolin, "Deep learning for detecting multiple space-time action tubes in videos," in *British Machine Vision Conference*, 2016.

[123] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," *arXiv preprint arXiv:1703.01515*, 2017.

[124] ——, "Cdc: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[125] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1049–1058.

[126] G. A. Sigurdsson, O. Russakovsky, and A. Gupta, "What actions are needed for understanding human actions in videos?" in *IEEE International Conference on Computer Vision*, 2017, pp. 2137–2146.

[127] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[128] ——, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[129] G. Singh, S. Saha, M. Sapienza, P. Torr, and F. Cuzzolin, "Online real-time multiple spatiotemporal action localisation and prediction," in *IEEE International Conference on Computer Vision*, 2017.

[130] S. Singh, A. Gupta, and A. A. Efros, "Unsupervised discovery of mid-level discriminative patches," in *Proceedings of European Conference on Computer Vision*, 2012.

[131] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild," *arXiv preprint arXiv:1212.0402*, 2012.

[132] B. Soran, A. Farhadi, and L. Shapiro, "Generating notifications for missing actions: Don't forget to turn the lights off!" in *IEEE International Conference on Computer Vision*, 2015, pp. 4669–4677.

[133] D. A. Spielmat and S.-H. Teng, "Spectral partitioning works: Planar graphs and finite element meshes," in *Foundations of Computer Science, 1996. Proceedings., 37th Annual Symposium on*. IEEE, 1996, pp. 96–105.

[134] O. Stretcu and M. Leordeanu, "Multiple frames matching for object discovery in video," in *Proceedings of British Machine VIsion Conference*, 2015.

[135] F. Temmermans, B. Jansen, R. Deklerck, P. Schelkens, and J. Cornelis, "The mobile museum guide: artwork recognition with eigenpaintings and surf," in *WIAMIS 2011, Delft, The Netherlands, April 13-15, 2011*. TU Delft; EWI; MM; PRB, 2011.

[136] J. B. Tenenbaum and W. T. Freeman, "Separating style and content with bilinear models," *Neural computation*, vol. 12, no. 6, pp. 1247–1283, 2000.

[137] J. Tighe and S. Lazebnik, "Finding things: Image parsing with regions and per-exemplar detectors," in *Proceedings of Computer Vision and Pattern Recognition*, 2013.

[138] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[139] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[140] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *Proceedings of International Conference on Computer Vision*, 2009.

[141] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[142] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[143] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "Hoggles: Visualizing object detection features," in *Proceedings of International Conference on Computer Vision*, 2013.

[144] C. Vondrick, H. Pirsiavash, and A. Torralba, "Anticipating visual representations with unlabeled video," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 98–106.

[145] ——, "Generating videos with scene dynamics," in *Advances In Neural Information Processing Systems*, 2016, pp. 613–621.

[146] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558.

[147] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *European Conference on Computer Vision*. Springer, 2016, pp. 20–36.

[148] X. Wang, A. Farhadi, and A. Gupta, "Actions $\sim$ transformations," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2658–2667.

[149] P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Learning to track for spatio-temporal action localization," in *IEEE International Conference on Computer Vision*, 2015, pp. 3164–3172.

[150] F. Xiao and Y. J. Lee, "Track and segment: An iterative unsupervised approach for video object proposals," in *Proceedings of Computer Vision and Pattern Recognition*, 2016.

[151] X. Xie, F. Tian, and H. S. Seah, "Feature guided texture synthesis (fgts) for artistic style transfer," in *Proceedings of the 2nd international conference on Digital interactive media in entertainment and arts*. ACM, 2007, pp. 44–49.

[152] Y. Xiong, Y. Zhao, L. Wang, D. Lin, and X. Tang, "A pursuit of temporal accuracy in general activity detection," *arXiv preprint arXiv:1703.02716*, 2017.

[153] M. C. Y. Chen and J. Ghosh, "Integrating support vector machines in a hierarchical output space decomposition framework," in *Proceedings of IGARSS*, 2004.

[154] B. Yao, A. Khosla, and L. Fei-Fei, "Combining randomization and discrimination for fine-grained image categorization," in *Computer Vision and Pattern Recognition (Computer Vision and Pattern Recognition), 2011 IEEE Conference on*. IEEE, 2011, pp. 1577–1584.

[155] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei, "Every moment counts: Dense detailed labeling of actions in complex videos," *International Journal of Computer Vision*, pp. 1–15, 2015.

[156] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, "End-to-end learning of action detection from frame glimpses in videos," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2678–2687.

[157] S. A. Yoon and J. Wang, "Making the invisible visible in science museums through augmented reality devices," *TechTrends*, vol. 58, no. 1, pp. 49–55, 2014.

[158] G. Yu and J. Yuan, "Fast action proposals for human action detection and search," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1302–1311.

[159] J. Yuan, B. Ni, X. Yang, and A. A. Kassim, "Temporal action localization with pyramid of score distribution features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3093–3102.

[160] Z. Yuan, J. C. Stroud, T. Lu, and J. Deng, "Temporal action localization by structured maximal sums," *arXiv preprint arXiv:1704.04671*, 2017.

[161] J. Zepeda and P. Perez, "Exemplar svms as visual feature encoders," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3052–3060.

[162] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *IEEE International Conference on Computer Vision*, 2017, pp. 2914–2923.

[163] ——, "Temporal action detection with structured segment networks," in *IEEE International Conference on Computer Vision*, 2017.

[164] H. Zhu, R. Vial, and S. Lu, "Tornado: A spatio-temporal convolutional regression network for video action proposal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5813–5821.

[165] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proceedings of European Conference on Computer Vision*. Springer, 2014.