Large-scale Learning from Video and Natural Language

Antoine Miech

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à l'École Normale Supérieure

Large-scale Learning from Video and Natural Language

Soutenue par
Antoine Miech
Le 14 Octobre 2020

École doctorale n°386
Sciences Mathématiques de Paris Centre

Spécialité
Informatique

Composition du jury :

Bernard Ghanem King Abdullah University of Science and Technology	*Rapporteur*
Dima Damen University of Bristol	*Rapporteur*
Vincent Lepetit ENPC	*Président du jury*
Cordelia Schmid Inria, Google	*Examinateur*
Jean-Baptiste Alayrac DeepMind	*Examinateur*
Ivan Laptev Inria	*Directeur de thèse*
Josef Sivic Inria, CIIRC CTU	*Directeur de thèse*

Abstract

The goal of this thesis is to build and train machine learning models capable of understanding the content of videos. Current video understanding approaches mainly rely on large-scale manually annotated video datasets for training. However, collecting and annotating such dataset is cumbersome, expensive and time-consuming. To address this issue, this thesis focuses on leveraging large amounts of readily-available, but noisy annotations in the form of natural language. In particular, we exploit a diverse corpus of textual metadata such as movie scripts, web video titles and descriptions or automatically transcribed speech obtained from narrated videos. Training video models on such readily-available textual data is challenging as such annotation is often imprecise or wrong. In this thesis, we introduce learning approaches to deal with weak annotation and design specialized training objectives and neural network architectures. In the first contribution, we propose a neural network architecture specifically designed for long-term representation of video and apply it to the YouTube-8M dataset with weak labels automatically obtained from metadata. Our proposed approach notably ranked first at the YouTube-8M video understanding challenge. In the next contribution, we use multiple-instance learning and propose a scalable discriminative clustering algorithm to recognize actors and their actions in movies given movie scripts as supervision. In the following contribution, we propose the HowTo100M dataset, a large-scale uncurated narrated instructional video dataset with annotations obtained using automatic speech recognition. Equipped with this data, we learn a joint text-video embedding that performs well on several text-video retrieval downstream tasks despite the highly noisy annotations. In the last contribution, we propose a multiple-instance contrastive loss for learning a visual representation only from the weakly annotated HowTo100M dataset and show the learned video representation can outperform fully-supervised methods on several downstream tasks.

Résumé

Nous nous intéressons à l'apprentissage automatique d'algorithmes pour la compréhension automatique de vidéos. Une majorité des approaches en compréhension de vidéos dépend de large base de données de vidéos manuellement annotées pour l'entraînement. Cependant, la collection et l'annotation de telles base de données est fastidieuse, coûte cher et prend du temps. Pour palier à ce problème, cette thèse se concentre sur l'exploitation de large quantité d'annotations publiquement disponible, cependant bruitées, sous forme de language naturel. En particulier, nous nous intéressons à un corpus divers de métadonnées textuelles incluant des scripts de films, des titres et descriptions de vidéos internet ou encore des transcriptions de paroles. L'usage de ce type de données publiquement disponibles est difficile car l'annotation y est faible. Pour cela, nous introduisons différentes approches d'apprentissage telles que de nouvelles fonctions de coûts ou architectures de réseaux de neurones, adaptées à de faibles annotations. En première contribution, nous présentons une architecture de réseau de neurones adaptées à la représentation de vidéos à longue durée. Nous évaluons notre modèle sur YouTube-8M, base de données faiblement annotée grâce à des métadonnées textuelles. Notre approche s'est notamment distinguée en obtenant la première place au challenge YouTube-8M. Dans la contribution suivante, nous introduisons un algorithme de clustering discriminatif à faible complexité spatiale et temporelle dans le cadre d'apprentissage multi-instance pour identifier des acteurs et leurs actions dans des films étant donnés des scripts de films pour supervision. Dans la prochaine contribution, nous présentons HowTo100M, une large base de données de vidéos narratives instructionelles avec pour annotation, la transcription automatique des paroles. Nous montrons qu'il est possible d'apprendre depuis cette données, une représentation jointe entre texte et vidéo particulièrement efficace sur plusieurs tâches de recherches de vidéos par language naturel, malgré les annotations très bruitées. En dernière contribution, nous proposons une approche d'apprentissage multi-instance et contrastive pour l'apprentissage d'une représentation visuelle depuis HowTo100M et montrons que la représentation apprise obtient de meilleurs performances que des représentations fortement supervisées sur plusieurs tâches.

Acknowledgements

Ivan, Josef. I believe I will never be grateful enough for having you as my advisors. I still remember sending a hopeless email to Ivan when I was looking for my master internship. I had almost no experience in computer vision nor attended the vision class from MVA and yet you surprisingly believed in me, took me as an intern and quickly after as a Ph.D. student. I would like to thank you for all the time you spent with me, for teaching me how to pursue good research, write papers, for conveying your vision, curiosity and passion for research.

I am also grateful to Bernard Ghanem, Dima Damen and Cordelia Schmid for accepting to be part of my jury defense with a special thanks to Vincent Lepetit who kindly accepted to physically chair the defense a week before. Thanks to Francis Bach and Benoît Sagot for being part of my comité doctoral. I would also like to thank Jean for creating and leading such an incredible and stimulating research lab that is WILLOW. The research lab would not have been possible without all of our amazing present and past research assistants: Kim, Sabrine, Hélène and Mathieu.

Being partly in charge of our wonderful sequoia cluster, I would like to especially thank all the people that have helped in contributing or maintaining sequoia. Notably Jean-Marc, Jean-Claude and Nassère for your invaluable help in maintaining or upgrading sequoia, Guillaume Seguin for being its first cluster manager and creating so many amazing monitoring tools we still extensively use now, Loïc for your constant support especially with the new Jean-Zay cluster and finally Yana and Yann for volunteering in being the next cluster managers while also managing your own Ph.D.

This thesis would not have been possible without the amazing collaborators that directly contributed to this thesis. Thank you Piotr for being one of my first influence and conveying me your rigorous approach of research. Thank you JB for being the person I had the chance to collaborate the most and for teaching me so many amazing things (such as how to debug python code without printing random stuff on screen). As you know, one rule of thumb to determine whether a paper I submit to a conference will be accepted is to check if you are also in. So I hope we will be able to continue this streak throughout our next collaboration at DeepMind. I am also grateful more generally for your work on the sequoia cluster, for accepting to be part of the jury and also for engaging the collaboration with DeepMind. Thank you, Dimitri and Makarand for the collaboration on the HowTo100M work. This project is the one I enjoyed the most and that will surely lead to many other exciting results. Thank you, Lucas for your help in the engineering part of the DeepMind collaboration. I hope we will be able to collaborate on even more exciting projects in the future. And finally, Antoine, I am very excited about this new ongoing collaboration we have.

A special thanks to the wonderful people I had the chance to share my office with. Thank you, Ignacio, for your constant good mood, for supporting me and helping me when needed. Thank you also to Ronan and Hadrien for making our office such a great place. More globally, I would also like to thanks the whole WILLOW and SIERRA teams for the memorable time I had in the lab, especially during deadlines, retreats, climbing sessions and amazing conferences abroad. I want to give a special shoutout to Thomas, who visited me when I was in the Bay Area and London.

During these four wonderful years, I also had amazing collaborations with industries. I would like to thank Paul Natsev for creating the popular YouTube-8M challenge. This challenge was very special to me as it put a large spotlight on my research and career. For instance, this lead to my Google Ph.D. fellowship award which you also help me with your mentorship. So I would like to say thank you for your invaluable impact on my career, and also Google AI for the financial support. I was also lucky to collaborate with amazing people at Facebook AI, Menlo Park. I would especially like to thank Du for hosting me, Heng, Lorenzo for the collaboration and fellow intern Gedas for all the great time we had during this internship. Finally, I would also like to thank DeepMind, its vision team and Andrew Zisserman who hosted me in 2019. I had a memorable summer in London and I am looking forward to returning as a full-time employee.

I am also grateful for the support of my friends, especially Willy and Mike who shown interest in my research by attending my talks or inviting me to give one.

Last but not least, nothing of this would have been possible without the support of Catherine, my family and especially my parents.

Contents

1 Introduction **2**

1.1 Goal 2

1.2 Motivation 3

1.3 Challenges 4

1.4 Contributions 7

1.4.1 Learnable pooling with context gating for video 8

1.4.2 Learning from movie scripts via large-scale discriminative clustering 9

1.4.3 HowTo100M: Learning a text-video embedding from millions of uncurated narrated instructional videos 10

1.4.4 Learning of visual representations from HowTo100M 10

1.5 Publications 11

1.5.1 The kaggle YouTube-8M challenge winning entry 12

1.5.2 The HowTo100M dataset and web search demo 12

1.5.3 HowTo100M pretrained text-video model 13

1.6 Manuscript outline 13

2 Related work **15**

2.1 Video representations 15

2.1.1 Image representations 16

2.1.2 Action recognition as a benchmark for video models 16

2.1.3 Short-term video representations 21

2.1.4 Longer-term video representations 25

2.2 Video and Language 27
2.2.1 Video captioning 27
2.2.2 Text-Video retrieval 29
2.2.3 Text and Video alignment 31
2.2.4 Visual grounding for translation 32
2.2.5 Video question answering 34
2.3 Learning from readily available data 36
2.3.1 Self-supervised learning 36
2.3.2 Learning from speech 37
2.3.3 Movie scripts, subtitles and DVS 39
2.3.4 Metadata from the internet 41

3 Learnable Pooling with Context Gating for Video Understanding 44
3.1 Introduction 45
3.2 Related work 47
3.2.1 Feature extraction 47
3.2.2 Feature aggregation 48
3.2.3 Gating 49
3.3 Network architecture 49
3.3.1 Context Gating 50
3.3.2 Relation to residual connections 51
3.3.3 Relation to LSTM/GRU 52
3.3.4 Motivation for Context Gating 54
3.4 Learnable pooling methods 55
3.4.1 NetVLAD aggregation 55
3.4.2 Beyond NetVLAD aggregation 56
3.5 Experiments 58
3.5.1 Youtube-8M Dataset 58
3.5.2 MPII Movie Description Dataset 59
3.5.3 Implementation details 60

3.5.4 Model evaluation for Classification 61
3.5.5 Model evaluation for Text-Video retrieval 62
3.5.6 Context Gating ablation study 62
3.5.7 Video-Audio fusion 64
3.5.8 Generalization 64
3.5.9 Ensembling 66
3.6 Conclusion 67

4 Learning from Video and Text via Large-Scale Discriminative Clustering 69
4.1 Introduction 70
4.1.1 Related Work 71
4.2 Discriminative Clustering for Weak Supervision 73
4.2.1 Large-Scale optimization 75
4.3 Weakly supervised Person-Action model 79
4.3.1 Weak-supervision as constraints 79
4.3.2 Person-Action model formulation 81
4.3.3 Slack variables 82
4.3.4 Lower bound 82
4.4 Experimental Setup 83
4.4.1 Dataset 83
4.4.2 Text pre-processing 84
4.4.3 Person detection and Features 84
4.4.4 Combining face and body tracks 85
4.5 Evaluation 86
4.5.1 Evaluation of person recognition 86
4.5.2 Evaluation of action recognition 87
4.6 Conclusion 91

5 HowTo100M: Learning a Text-Video Embedding from Uncurated Narrated Videos 92

5.1 Introduction 93
5.2 Related work 95
5.3 The HowTo100M dataset 98
5.3.1 Data collection 98
5.3.2 Paired video clips and captions 99
5.4 Text-video joint embedding model 101
5.4.1 Loss implementation 103
5.5 Experiments 104
5.5.1 Implementation details 105
5.5.2 Datasets and evaluation setups 105
5.5.3 Study of negative pair sampling strategy 107
5.5.4 Scale matters 107
5.5.5 Comparison with state-of-the-art 108
5.5.6 Cross-dataset fine-tuning evaluation 111
5.5.7 Sampling strategy for positive pairs 111
5.5.8 Qualitative results 113
5.6 Conclusion 114

6 End-to-End Learning of Visual Representations from Uncurated Instructional Videos 119

6.1 Introduction 120
6.2 Related work 123
6.3 Leveraging Uncurated Instructional Videos 125
6.3.1 A simple joint probabilistic model 126
6.3.2 Learning from uncurated data: MIL-NCE 127
6.4 Experiments 129
6.4.1 Implementation details 130
6.4.2 Downstream tasks 131
6.4.3 Ablation studies 132
6.4.4 Comparison to the state-of-the-art 136

6.4.5 Max+NCE and Attn+NCE baselines 140
6.5 Conclusion 141

7 Discussion and perspectives 143
7.1 Summary of contributions 143
7.2 Perspectives 145
7.2.1 Leveraging narrated videos 145
7.2.2 Tackling more complex video and language tasks 148

Bibliography 149
rea

Chapter 1

Introduction

1.1 Goal

The goal of this thesis is to design machine learning models and algorithms enabling the understanding of video content. For instance, we wish to create models with the ability to understand basics concepts such as: Where does the scene take place? What objects are seen? Who is in the scene? or What are the performed actions.

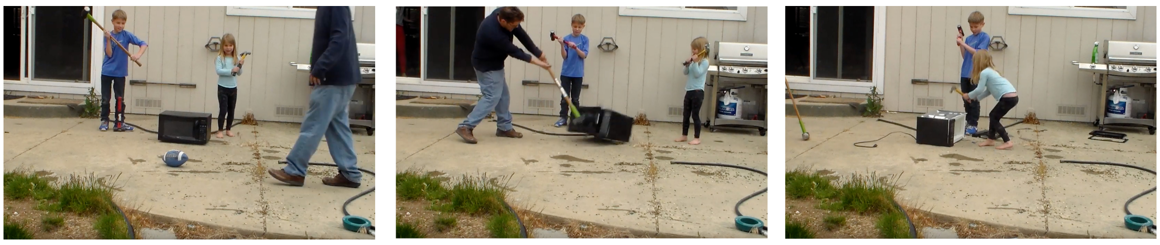


Figure 1-1: A typical YouTube video depicting three people: a man, likely a dad with his two young children, having a good time sledgehammering a microwave in a backyard.

In this thesis, we focus our efforts on leveraging readily-available natural language resources for supervising video models. To learn such video models, we need a way to bridge the gap between the vast knowledge of the world on the one hand, and video data represented as raw pixel-wise values, on the other hand. Natural language can be used to express in a semantically dense manner, the content of visual data. As an illustration, the caption from the Figure 1-1 provides useful information about the content of the shown video: *i.e.* Where does the scene take place? In the *backyard.* Who is in the scene? *a dad with his two young children.* What are they doing? They are *sledgehammering a microwave.* Why are they doing that? To have *a good time.* In this thesis, we consider a wide range of *readily-available* and large-scale natural language resources such as movie scripts, transcripts of audio description from movies aimed at visually impaired people, titles and descriptions of web video or speech transcripts of narrated instructional videos.

We apply and evaluate our learnt video models on two main tasks. The first one is video category classification: *i.e.* given an input video, what are the most relevant labels, that accurately describe the video content? The second one is text-video retrieval: *i.e.* given an input text query and a pool of thousands of testing videos, what are the most relevant videos best matching the text query? Conversely, given an input video and a set of textual descriptions, what is the best description for the input video? Note that as opposed to video classification, the retrieval task requires jointly modeling video and natural language. We believe being able to solve these two tasks provides the first step toward basic video understanding by machines.

1.2 Motivation

Videos are widely used to share content on social media, to archive special moments of life, for news and entertainment, for surveillance and as a means of communication in video calls. Due to the growth of video data and its consumption, the automatic understanding of video content is becoming useful and sometimes critical for a wide range of applications. For example, being able to automatically moderate newly uploaded videos that show inappropriate content is essential for social media platforms. It can also be useful to sort videos based on categories or to provide a way to search for videos given textual queries to enhance user experience. Similarly, being able to automatically search within a collection of personal videos is convenient when one needs to find back a specific video recorded years ago among hundreds of others. In video surveillance, the manual inspection of videos is becoming impractical and prevents scalability. On the other hand, machines could enable real-time, accurate and large-scale processing of surveillance footage.

One drawback with current methods for video understanding is that they mostly rely on visual datasets with clean manual labels such as ImageNet [Russakovsky et al., 2015] for images or Kinetics [Carreira and Zisserman, 2017] for videos. Manual annotation of videos and images is time-consuming, expensive and thus not scalable. While it is a necessary step to annotate video datasets for evaluation, it would be highly beneficial to avoid manual annotation for training. Moreover, labeling videos is not trivial: Which videos should be annotated? How to collect the videos? What annotation platform to use? What taxonomy to consider? How to check that the collected annotation is consistent and reliable? In this thesis, we aim at leveraging readily available natural language resources to avoid this burdensome process when training video models. Another motivation for considering natural language is that it enables to train video models for text-to-video retrieval, which as described above, is useful in many practical applications.

1.3 Challenges

Automatic video understanding from readily-available natural language data is an emerging research field with many challenges. The first one being the video understanding part, and more precisely, how we can design a representation for video that can effectively encode all of its useful information? Second, working with readily available natural language data for video modeling is challenging since the supervision is often weak, unreliable and sparse. It is thus often crucial to consider video models and training methods compatible with such supervision. Finally, training models on large-scale video datasets implies high computational demands. One challenge we also need to address is in designing computationally efficient video models. We will individually elaborate on these challenges next.

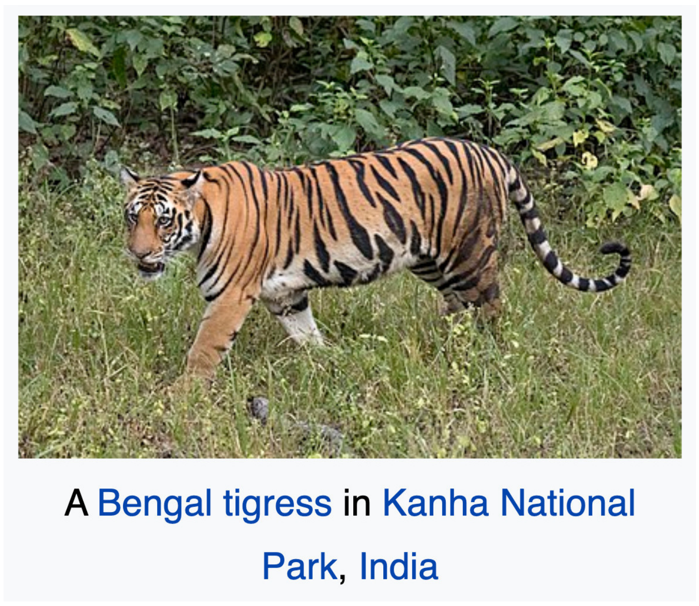

A Bengal tigress in Kanha National Park, India

Figure 1-2: An image from Wikipedia with its descriptive caption.

Video representation. One of the main challenges addressed in this thesis is the design of efficient video representations that can capture useful high-level semantic information about the video content. As opposed to 2D convolutional neural network (CNN) which have consistently demonstrated great efficiency in encoding high-level semantic information in images, it is not clear yet which representation or neural network architecture could show a similar unanimous success for videos. In fact, as opposed to image representations, video representations should capture temporal dynamics in videos, for example, to differentiate between *Opening Door* and *Closing Door* actions. Moreover, it is desirable to encode videos of variable length within a fixed-size representation while preserving temporal information.

In this thesis, we first propose and analyze alternative neural network architectures in aggregating visual information over time in Chapter 3 In Chapter 6, we demonstrate that the training of complex state-of-the-art 3D CNNs can be done from scratch without using manually labeled videos.

Learning visual models from readily-available natural language resources. There is an abundant amount of readily-available textual data on the Internet (*e.g.*

Wikipedia or Google News). However, the first challenge for computer vision applications is to identify resources that can provide useful supervision to train vision models. More specifically, a corpus of natural language could be more useful for computer vision if it is describing the content of visual data. Figure 1-2 illustrates an example of a picture from Wikipedia with its underlying descriptive caption. This readily-available caption can be considered as a useful supervisory signal for training visual models as it provides the information of what is in this picture (a *Bengal tigress*) and also where the picture was taken (in a *National Park*). In this thesis, we leverage supervisory information obtained from readily-available natural language resources, namely, web video title and description metadata in Chapter 3, movie scripts in Chapter 4 and automatic speech transcripts from narrated videos in Chapters 5 and 6.

Using readily-available natural language resources implies two main advantages. First, we can eliminate all burden of manual video annotation. Second, such data typically contains high diversity and can be found in large quantities. However, the downside when considering such data is that it is often not reliable and thus challenging to exploit. Since the textual data is not specifically created for machine learning purposes, it may lead to multiple issues such as sparse supervision (*i.e.* only one word in an entire paragraph is useful for the supervision), wrong supervision (*i.e.* textual information is unrelated to the visual content) or missing correspondence (*i.e.* which part of the video corresponds to which sentence). For illustration, Figure 1-3 shows a YouTube video on a race of remotely controlled cars with the video title: *Turkey Farm Nate and Duck.* As opposed to the Wikipedia image example from Figure 1-2, here the title of the video does not correspond to its visual content. Moreover, the title provides misleading visual cues such as *Duck* which is a kid name and not the animal and *Turkey* which is the name of the place and not the animal. In this thesis, we address these issues and design models and training methods enabling learning from weak and noisy supervision.

Learning from large-scale video datasets and the computational challenge.

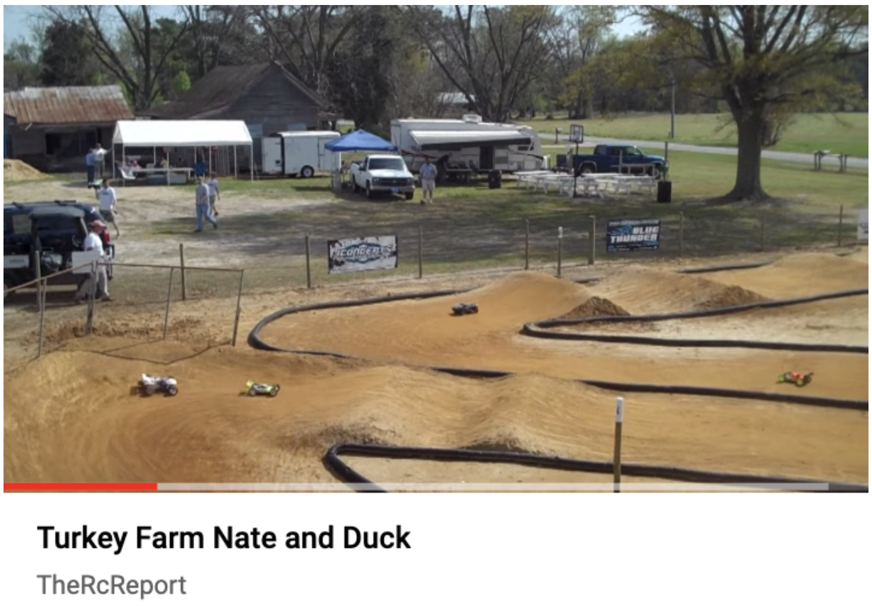


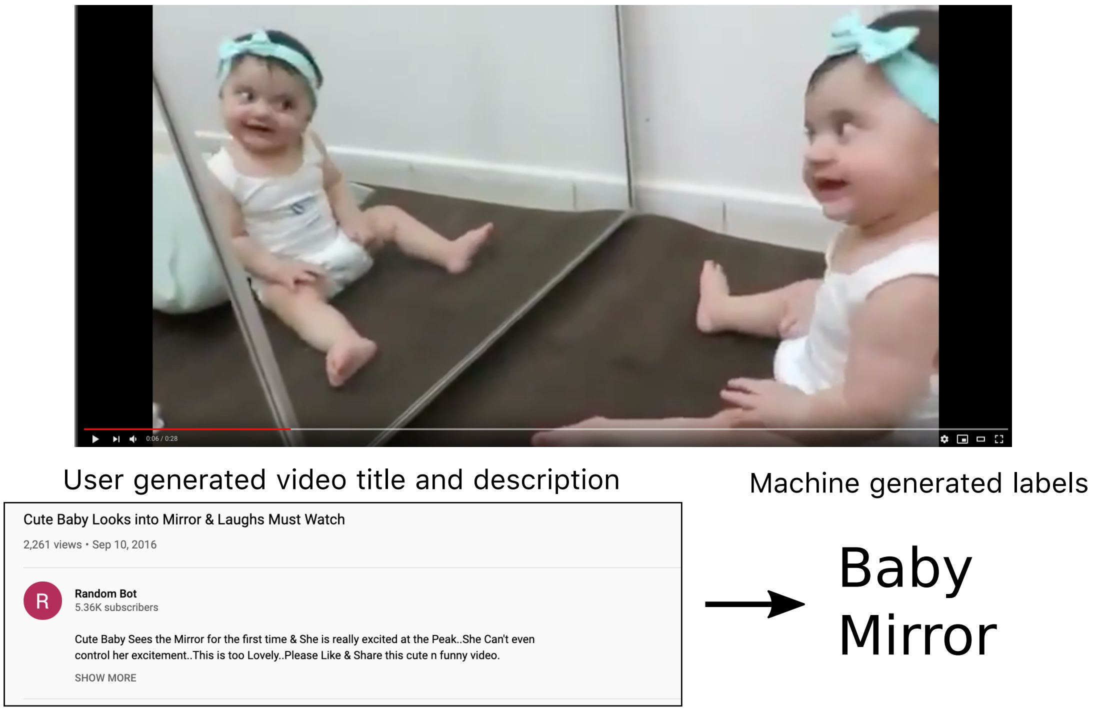
Figure 1-3: A YouTube video on a race of remotely controlled cars (`www.youtube.com/watch?v=--7brUsSjvQ`) and its misleading title: *Turkey Farm Nate and Duck*.

To fully leverage large-scale resources with weak supervision, we need to design models and learning methods with the ability to scale to potentially millions of training videos. Video processing requires significantly more computational and memory resources compared to images, audio or textual data. For this reason, we address computational complexity and design efficient methods that can seamlessly scale to large number of training videos.

1.4 Contributions

This thesis provides the following four contributions. First, we introduce a state-of-the-art neural network architecture for long-term video representation and apply it on the large-scale YouTube-8M dataset [Abu-El-Haija et al., 2016] with labels automatically generated from natural language metadata. Second, we propose a video model capable of learning to recognize people and their actions in movies by mining supervision from a large corpus of readily-available movie scripts. Third, we collect a new large-scale uncurated dataset of narrated instructional videos named

HowTo100M and show how to leverage it for learning a joint text-video representation by only using readily available speech transcripts. Finally, we propose a new MIL-NCE loss and demonstrate its ability to learn a state-of-the-art video representation from HowTo100M without any manually annotated visual data. These four contributions are outlined in more detail below.



Cute Baby Looks into Mirror & Laughs Must Watch

2,261 views • Sep 10, 2016

Random Bot
5.36K subscribers

Cute Baby Sees the Mirror for the first time & She is really excited at the Peak..She Can't even control her excitement..This is too Lovely..Please Like & Share this cute n funny video.

SHOW MORE

Figure 1-4: A YouTube video with its user generated title and description that are used to automatically generate visual labels in the YouTube-8M dataset.

1.4.1 Learnable pooling with context gating for video

An enormous amount of internet videos such as YouTube videos come with readily available descriptive metadata in the form of natural language such as video titles or descriptions. The YouTube-8M dataset [Abu-El-Haija et al., 2016] leverages these metadata by automatically extracting video labels from such metadata using NLP methods [1], as illustrated in Figure 1-4. However, despite having a large number of training videos, the YouTube-8M dataset comes with two issues. First, the machine-

[1]https://www.youtube.com/watch?v=wf_77z1H-vQ

generated labels are not fully reliable since they are not manually verified. Second, because these labels are generated at the video level, we need to deal with full videos with duration in the order of several minutes. In our first contribution, we propose a neural network architecture for aggregating visual features for minute-long videos, capable of classifying these videos despite noisy labels. Our proposed method achieves state-of-the-art results on the YouTube-8M dataset and reached the first place at the *Google Cloud and YouTube-8M Video Understanding Challenge* [2] over 655 other participating teams. Besides, our method enables better scalability and requires fewer training examples to perform well compared to competing approaches.

1.4.2 Learning from movie scripts via large-scale discriminative clustering

A large number of movie scripts are freely available on the Internet [3]. These scripts provide detailed information about the dialogues, speakers and the actions of actors in the movies. As a second contribution of this thesis, we propose a video model that learns to recognize actors and actions in movies using movie scripts as the sole source of supervision. We formulate the problem in the framework of multiple-instance learning and introduce a model based on discriminative clustering [Bach and Harchaoui, 2007]. To scale the training algorithm to many movies, we also propose a novel optimization method based on the Block-Coordinate Frank-Wolfe [Osokin et al., 2016] algorithm that enables efficient optimization of the training objective. We train our model on 66 feature-length movies and show significant improvements in action and actor recognition compared to the prior state-of-the-art.

[2] `https://www.kaggle.com/c/youtube8m`

[3] `https://www.imsdb.com/`

1.4.3 HowTo100M: Learning a text-video embedding from millions of uncurated narrated instructional videos

State-of-the-art approaches to automatic speech recognition (ASR) have made significant advances in the past few years. In our third contribution, we leverage the recent advances in ASR by collecting HowTo100M, a large-scale dataset of 1.2 millions of uncurated narrated instructional videos together with their automatically generated speech transcripts. We propose a fast procedure for data collection and do not rely on any manually annotated video. Instructional videos typically contain thorough narrations explaining the actions and the setup required to perform a task. We exploit this and learn a joint text-video embedding by matching video clips to ASR outputs on HowTo100M and show that our learnt text-video embedding can be used on several downstream tasks despite not being trained on any manually annotated videos. We evaluate our learnt text-video embedding on the retrieval task and show that our model can outperform models trained on manually annotated videos from a similar domain to instructional videos. For videos with a larger domain gap such as movies, we show that our HowTo100M pretrained text-video embedding can still be used as a strong pretraining initialization leading to significant improvements over randomly initialized models. We emphasize this work being first to train an efficient joint text-video embedding without any manually labeled video description dataset.

1.4.4 Learning of visual representations from HowTo100M

In the previous contributions, we avoid video annotations by leveraging readily available natural language resources. Our visual models, however, were initialized by pretraining them on manually annotated datasets such as ImageNet [Russakovsky et al., 2015] or Kinetics [Carreira and Zisserman, 2017]. In this last contribution, we address this issue by showing we can also learn the video representation from scratch without relying on any manually annotated visual data. We propose a multiple-instance learning loss (MIL-NCE), capable of learning a state-of-the-art video representation from scratch using HowTo100M. We evaluate our learnt representations on several downstream tasks, namely, action recognition, action segmentation, text-video video retrieval and action localization and show that our weakly-supervised representations can outperform fully-supervised ones on many downstream tasks. We emphasize this approach to be the first published method demonstrating improvements over fully supervised state-of-the-art video representations while using no manual annotation.

1.5 Publications

This thesis is based on the following four publications:

- [Miech et al., 2017b]: Antoine Miech, Ivan Laptev and Josef Sivic. Learnable pooling with Context Gating for video classification. In CVPRW YouTube-8M workshop, 2017. (based on Chapter 3)
- [Miech et al., 2017a]: Antoine Miech, Jean-Baptiste Alayrac, Piotr Bojanowski, Ivan Laptev and Josef Sivic. Learning From Video and Text via Large-Scale Discriminative Clustering. In ICCV (Spotlight), 2017. (based on Chapter 4)
- [Miech et al., 2019b]: Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In ICCV, 2019. (based on Chapter 5)
- [Miech et al., 2020]: Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In CVPR (Oral), 2020. (based on Chapter 6)

Other work by the author not included in this thesis is described in the following papers:

- [Miech et al., 2018]: Antoine Miech, Ivan Laptev and Josef Sivic. Learning a Text-Video Embedding from Incomplete and Heterogeneous Data. arXiv preprint arXiv:1804.02516, 2018.

- [Miech et al., 2019a]: Antoine Miech, Ivan Laptev, Josef Sivic, Heng Wang, Lorenzo Torresani and Du Tran. Leveraging the Present to Anticipate the Future in Videos. In CVPR Precognition workshop, 2019.

1.5.1 The kaggle YouTube-8M challenge winning entry

The workshop publication [Miech et al., 2017b] is the report describing the winning entry of the kaggle Google Cloud and YouTube-8M Video Understanding Challenge (`https://www.kaggle.com/c/youtube8m`). The goal of this challenge was to correctly predict the visual tags of hundreds of thousands of YouTube videos by training video models on the YouTube-8M dataset [Abu-El-Haija et al., 2016]. Our proposed approach, described in Chapter 3 has been ranked first among 655 other participating teams [4]. We open-sourced the Tensorflow code reproducing the winning entry at `https://github.com/antoine77340/Youtube-8M-WILLOW`.

1.5.2 The HowTo100M dataset and web search demo

We have publicly release the HowTo100M dataset (`https://www.di.ens.fr/willow/research/howto100m/`) with the publication of [Miech et al., 2019b] (Chapter 5). We provide for downloading the collected speech transcripts as well as mirror links for the 1.2 million videos (12 TB of data).

For demonstration purposes, we have also implemented a real-time and interactive text-to-video search demo available on the HowTo100M project website. This demo allows searching given an input text query, localized video clips from the full 15 years of HowTo100M videos using a compressed version of the model described in Chapter 5. The demo runs in real-time on a single CPU machine using the approximate similarity search library FAISS [5].

[4] `https://www.kaggle.com/c/youtube8m/leaderboard`

[5] https://github.com/facebookresearch/faiss

1.5.3 HowTo100M pretrained text-video model

We have publicly open-sourced the Tensorflow I3D[6] as well the Tensorflow[7] and the PyTorch[8] S3D HowTo100M pretrained text-video model using the MIL-NCE method from the publication [Miech et al., 2020] (Chapter 6).

Moreover, a zero-shot YouCook2 [Zhou et al., 2018b] interactive web search demo using the S3D based pretrained HowTo100M text-video model is publicly available at `https://www.di.ens.fr/willow/research/mil-nce/`.

1.6 Manuscript outline

The thesis is organized into seven chapters, including this introduction.

The Chapter 2 is a literature review of work related to this thesis. The chapter is divided into three main sections: *(i)* Video representation, *(ii)* Video and natural language and finally *(iii)* Learning from readily available data in computer vision.

Chapter 3 presents details of our first contribution on learnable representations for long-term video modeling using machine generated labels from the YouTube-8M dataset [Abu-El-Haija et al., 2016], as summarized in section 1.4.1.

In Chapter 4, we describe the second contribution on scaling-up the training from readily available movies scripts to improve actor and action recognition in movies, as summarized in section 1.4.2.

In Chapter 5, we present the HowTo100M dataset with uncurated narrated instructional videos and demonstrate how we can use it to train a text-video embedding model without manual annotation, as summarized in section 1.4.3.

[6]`https://tfhub.dev/deepmind/mil-nce/i3d/1`
[7]`https://tfhub.dev/deepmind/mil-nce/s3d/1`
[8]`https://github.com/antoine77340/S3D_HowTo100M`

Chapter 6 describes details of our final contribution where we introduce the MIL-NCE loss and demonstrate state-of-the-art video representations learned entirely on the HowTo100M dataset as summarized in section 1.4.4

Finally, in Chapter 7, we summarize the thesis and outline perspectives on open problems and work.

Chapter 2

Related work

In this chapter, we review the literature closely related to this thesis. The chapter is divided into three main sections. We start first with foundational work related to the visual representation of videos in section 2.1. Next, we study works at the interface between video and natural language in section 2.2. Finally, this chapter ends with a review of works that leverage large-scale readily available data for learning visual models 2.3.

2.1 Video representations

Video representations (or descriptors) aim at encoding a raw video input into a compact code, rich of visual semantics. It is thus fundamental and necessary to consider an efficient video descriptor for its understanding. In particular, it should be made easy to decode basic information such as: what objects can be seen, where the video takes place, how many people there are, or what a person is doing in the video. We will review in this section, work related to the design or the learning of efficient video representations for video understanding.

2.1.1 Image representations

Before diving into the specific topic of video representation, it is first important to briefly review the key literature in the image domain as many of these works have influenced or have been extended to the video domain.

More than a decade ago, the most popular and efficient image representation techniques were based on hand-crafted descriptors. For example, [Lowe, 1999] proposed the highly known Scale Invariant Feature Transform (SIFT). The method, initially proposed for matching points between objects from different views, detects salient keypoints from an image and extracts scale invariant descriptors from these points. [Dalal and Triggs, 2005] proposed the Histogram of Oriented Gradient (HOG) by computing the gradient of images divided into grids and encoding each grid with histograms of the orientation of the gradients.

A major breakthrough in image recognition was made in 2012, when a convolutional neural network (CNN) [Krizhevsky et al., 2012] achieved first place at the ILSVRC-2012 challenge based on the ImageNet dataset [Russakovsky et al., 2015], outperforming all hand-crafted image representations for the first time. Instead of hand-crafting the image representation, the goal of such an approach is instead to learn in a supervised manner, a good image representation for recognition. This key contribution lead to many other improvements in CNNs [Simonyan and Zisserman, 2015; He et al., 2016] that kept improving the recognition accuracy on the ImageNet dataset. After 2012, CNNs quickly became the standard in image representations and we will see next that it also enabled similar breakthroughs in the video domain.

2.1.2 Action recognition as a benchmark for video models

In the video domain, the task of human action recognition quickly became the de facto standard for evaluating video representations. In fact, to recognize complex human action in videos, it is often required to understand the content of the images, frame by frame, to identify for example: where the video takes place or what object a person is interacting with. But more importantly, it is crucial to also model the

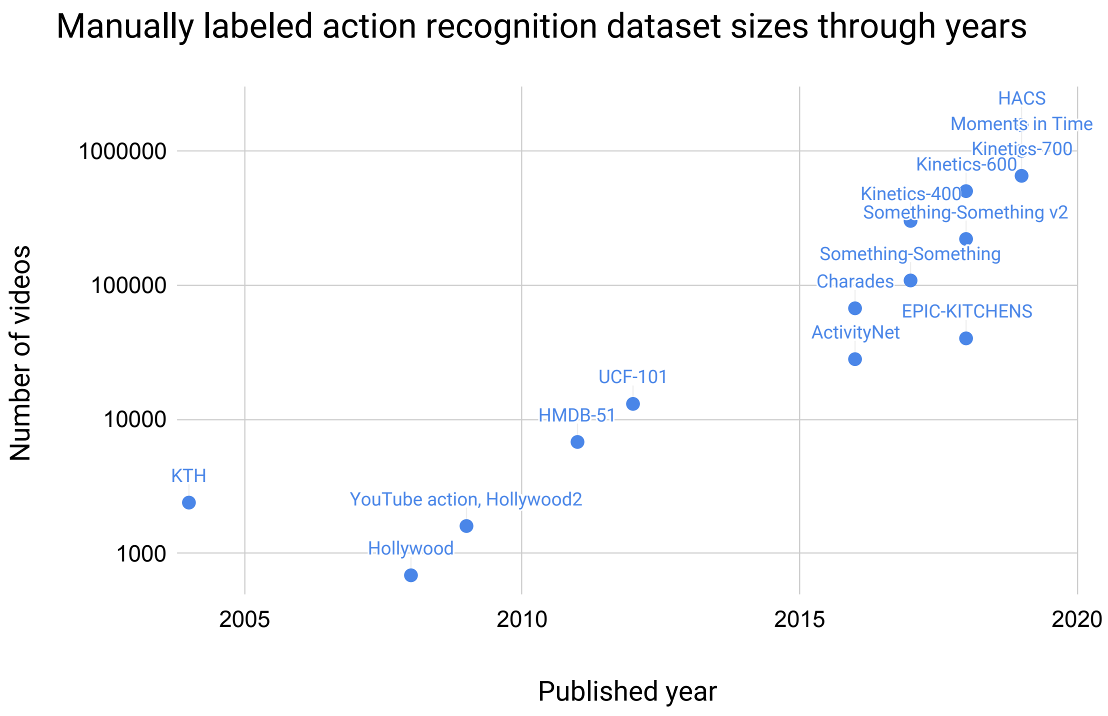


Figure 2-1: Manually annotated action recognition dataset number of videos through years.

temporal dynamic required to distinguish an action *Opening a door* from an action *Closing a door*, or to distinguish an action *Jogging* from an action *Sprinting*. Next, we briefly review in chronological order, the main action recognition datasets used for evaluating video representations and how each one of them has influenced the current datasets. The Figure 2-1 summarizes via a plot, the growing size of the main manually annotated action recognition datasets over time.

Action recognition datasets. The early days action recognition video datasets were small, with a limited diversity of action classes. The KTH [Schüldt et al., 2004] dataset is a manually recorded set of 2391 videos annotated with 6 simple action classes: walking, jogging, running, boxing, hand waving and hand clapping. The Hollywood [Laptev et al., 2008] dataset focused on human actions from Hollywood movies. It leveraged for the first time movie scripts to mine action descriptions as opposed to manually recorded videos. The dataset contains 687 videos annotated with 8 action classes. It was later on extended to Hollywood2 [Marszalek et al., 2009] with 12 classes and 1694 annotated videos. The YouTube action dataset [Liu et al., 2009] was the first action recognition video dataset collected from a public video sharing platform of 1600 videos with 11 action labels. These videos introduced novel challenges in video understanding such as severe camera motions, inconsistent video quality across the dataset, or user edited videos.

The HMDB-51 [Kuehne et al., 2011] is a collection of movie clips uploaded to YouTube. It both significantly increased the number of annotated videos to 6766 as well as the number of action classes to a diverse set of 51 actions. One year later was published another larger-scale video dataset, UCF-101 [Soomro et al., 2012], also collected from YouTube, which doubled the size of annotated videos and action classes of HMDB-51. These two datasets are still as of today, one of the most popular action recognition benchmarks.

In 2014, the Sports-1M [Karpathy et al., 2014b] dataset significantly outsized UCF-101 by reaching the symbolic number of 1 million of sports videos from YouTube, as well as increasing the number of action classes to 487. This significantly larger-scale dataset was motivated by the recent adoption of CNN models and their need for even more training data compared to prior traditional computer vision models. To scale-up the annotation, they were the first to automatically generate labels from metadata which are not as reliable as labels annotated manually. Another downside of the dataset is that the action classes are not fine-grained as they represent sports rather than atomic actions. In many cases, there is no need to model temporal dynamics from these videos as one frame can be enough to predict the sport class.

In 2016, the ActivityNet-200 dataset [Caba Heilbron et al., 2015] was published. It contains more than 28k videos and 200 action classes. One key feature about the provided annotations is the precise start and end time information for each annotated action sample, which can be used for action localization. This benchmark became popular due to its yearly challenge in action recognition and localization [1].

[1]http://activity-net.org/challenges/2020/index.html

Many of these datasets are focused on actions that are *interesting* enough so they can be easily collected from video sharing platforms or frequently occur in movies. [Sigurdsson et al., 2016a] argue that in many practical applications of computer vision, it is useful to also recognize *boring* actions: *i.e.* daily actions representing our everyday lives such as *Opening a refrigerator* or *Drinking from a cup*. However, such actions are more challenging to mine from movies or video-sharing platforms as they are ordinary. To address this issue, they collected Charades [Sigurdsson et al., 2016a], a dataset of 9848 videos with 157 classes, collected by instructing people to record themselves performing an ordinary action. Later, [Fouhey et al., 2018] propose to mine such daily activities from *Lifestyle VLOGs*, which can be found at scale on social media platforms.

Next, in 2017 was published the Kinetics-400 [Carreira and Zisserman, 2017] dataset, which quickly became as of today, the most popular benchmark in action recognition and video representation learning due to its size, label diversity and quality of annotations. The dataset consists of more than 300k short 10 seconds video clips annotated with 400 different action labels, making it the largest manually annotated video dataset. It was then extended in 2018 to Kinetics-600 [Carreira et al., 2018] (500k videos, 600 classes) and more recently in 2019 to Kinetics-700 [Carreira et al., 2019] (650k videos, 700 classes).

Another very popular dataset in modern action recognition is the Atomic Visual Actions (AVA) dataset [Gu et al., 2018], a collection of movie clips publicly available on YouTube and annotated with 80 atomic actions. The dataset is commonly used for benchmarking video representations that require spatio-temporal localization of actions as each of them is delimited with a spatio-temporal bounding box.

For many of these datasets, a significant number of actions can be recognized from a single frame. Such actions can often be discriminated by recognizing the context from objects or surroundings. For instance, someone holding a bow is very likely to do archery or someone in a swimming pool is likely to be swimming. In these examples, image recognition models work fine and no model of the temporal dynamic of the frames is required. To put more emphasis on the temporal aspect of videos,

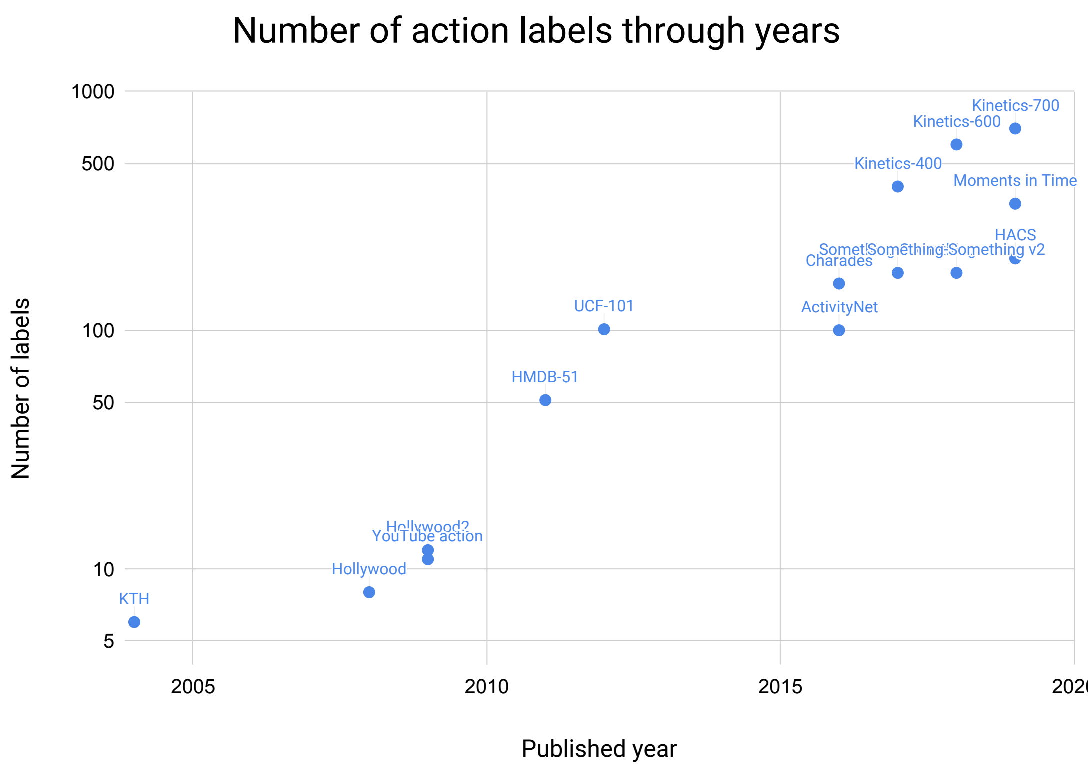


Figure 2-2: Number of action labels in manually annotated action recognition datasets through years.

[Goyal et al., 2017] published the Something-Something dataset, a large-scale dataset of 108k crowdsourced videos with 174 annotated actions. Each action from this dataset is specifically chosen to be impossible to recognize from one single image and instead require to understand the temporal dynamics of the video. Such challenging actions include: *Putting something on a surface*, *Stuffing something into something* or *Pretending to spread air onto something*. The dataset was later extended to a total of 220k videos.

All of these action recognition datasets are mainly third-person view recorded videos, which is problematic for mixed-reality or augmented-reality applications where the input videos have typically the first-person view. [Damen et al., 2018] introduced the EPIC-KITCHENS dataset, the largest published ego-centric action recognition dataset with roughly 40k annotated video clips. Moreover, as opposed to many action recognition datasets, the notion of the action label is defined as a *composition* of a verb (*e.g.* *Cut*, *Open* or *Wash*) plus a noun (*e.g.* *Carrot*, *Cup* or *Fridge*). With a taxonomy of 125 verbs and 331 nouns, the theoretical number of possible action labels in EPIC-KITCHENS is $125 \times 331 = 41375$, which exceeds by far the number of action labels from prior non-compositional action recognition datasets. This is valuable as the growth of the number of action labels from these non-compositional action recognition dataset saturates to a number below 1000 labels, as shown in Figure 2-2.

In this thesis, we instead focus on supervising video models with natural language as opposed to simple labels. It can be considered as an approach to address the label diversity limitation we are currently witnessing in categorized video datasets.

Next, we review work related to short-term (*i.e.* a few seconds) video representations that were mainly benchmarked on these action recognition datasets.

2.1.3 Short-term video representations

Identically to image representation approaches, the main methods for short-term video representation can be separated into two distinct categories: the hand-crafted video representations and the learned video representations, which rely on deep convolutional neural networks.

Hand-crafted descriptors. The first short-term video representation methods were exclusively hand-crafted local video features. [Laptev and Lindeberg, 2003] introduced the space-time interest point (STIP) descriptors, which aims at detecting local structures in space and time where image frames have significant local variations. Local spatio-temporal and scale-invariant descriptors are then extracted from these interest points to represent the video. Later, [Laptev et al., 2008] extended the STIP by dividing the video input into different spatio-temporal grids and computing bag-of-features representation of histogram of oriented gradient (HOG) and histogram of optical flow (HOF) features inside each bin. The method outperformed the prior state-of-the-art action recognition models on KTH [Schüldt et al., 2004] by a significant margin.

[Wang et al., 2011] introduced a novel video description method based on dense trajectories (DT). They proposed an efficient approach for densely sampling points within a video and tracking them using the optical flow fields. Additionally to the HOF and HOG local descriptors, [Wang et al., 2011] also proposed to extract motion boundary histograms (MBH) features along these trajectories to represent videos in a manner less sensitive to camera motion. Dense trajectories outperformed prior state-of-the-art on Hollywood2 [Marszalek et al., 2009] and the YouTube action [Liu et al., 2009] datasets by a large margin.

[Wang and Schmid, 2013] improved the dense trajectories by proposing a method to efficiently filter out the trajectories induced by camera motion. The improved dense trajectories were thus more effective at capturing human motion rather than noisy camera motion and demonstrated state-of-the-art-results on the HMDB-51 [Kuehne et al., 2011] dataset. This video representation is, as of today, the best hand-crafted video representation for action recognition and can, in some cases, still compete with convolutional neural network based video representations we review next.

Representations based on convolutional neural networks. As opposed to hand-crafted video features, the philosophy behind convolutional neural network based approaches is to learn the representation while still hand-crafting a particular convolutional neural network architecture. These CNN based methods quickly emerged after the breakthrough by CNNs in image recognition.

[Karpathy et al., 2014b] explored the use of 2D CNNs for sports video classification on their large-scale Sports-1M dataset. They study how to leverage image CNN architectures to model the temporal dimension in videos with Early Fusion, Late Fusion and Slow Fusion approaches. The study, however, did not conclude with an approach capable of efficiently leveraging the temporal information to get a better classification accuracy.

Thus, to model motion between subsequent frames, [Simonyan and Zisserman, 2014] proposed a two-stream approach. In this method, the first branch (the appearance branch) consists of a 2D CNN operating on an RGB stream while the second branch (the motion branch) consists of another 2D CNN operating on a two channels optical flow stream representing the motion between two subsequent frames. The proposed method reported significant improvements by combining the appearance branch with the motion branch. Two-stream approaches were further studied and improved in [Feichtenhofer et al., 2017; Wang et al., 2016a].

Another popular alternative to better model spatio-temporal patterns with CNNs is to instead replace 2D spatial convolutions by 3D spatio-temporal convolutions blocks as proposed by [Baccouche et al., 2011; Ji et al., 2013] for the task of human action recognition. However, [Baccouche et al., 2011] only experimented on the small KTH dataset [Schüldt et al., 2004] while [Ji et al., 2013] relied on multiple preprocessing steps such as running a human detector and a head tracking system to segment humans in videos. Later, [Tran et al., 2015] demonstrated the use of 3D CNNs in a larger-scale setup by training on UCF-101 [Soomro et al., 2012] and without using any preprocessing of the video frames. Moreover, this study has shown that as opposed to hand-crafted features, it is possible to learn a compact and strong video representation with convolutional neural networks as they reported a $52,8\%$ top-1 accuracy on UCF-101 with a video representation of only 10 dimensions. [Varol et al., 2017] study the benefits of using 3D CNNs for representing longer videos, *i.e.* 100 frames instead of only 16 frames. [Carreira and Zisserman, 2017] propose to inflate 2D convolution kernels from a CNN pretrained on images to 3D convolution kernels as initialization for training 3D CNNs. They demonstrate the benefit of doing so even in the large-scale training regime on the Kinetics dataset. Interestingly, [Carreira and Zisserman, 2017; Varol et al., 2017] also shows that although 3D convolutions are theoretically able to model motion between subsequent frames, adding a motion stream based on optical flow input yields further improvements. This suggests that vanilla 3D CNNs architectures might be over parametrized or not constrained enough for learning motion patterns.

Instead of using vanilla 3D convolutions to capture spatio-temporal patterns, [Qiu et al., 2017; Tran et al., 2018; Xie et al., 2018] study CNN architectures that factor out a 3D convolution as a separated 2D convolution for capturing spatial features and a 1D convolution for encoding the temporal features. More specifically, [Qiu et al., 2017] explore how to combine the 2D and 1D convolutions and [Tran et al., 2018; Xie et al., 2018] study at which layer decomposing the 3D convolutions is the most effective. They show that decomposing 3D convolutions as 2D and 1D separated convolutions leads to better performance while also reducing the complexity of the models. [Feichtenhofer et al., 2019] also use such decomposition in their proposed SlowFast neural network. In this approach, the network is composed of a fast pathway that processes the video at a high frame rate while a second slow pathway processes the video input at a low frame rate but at a higher spatial resolution. The motivation is that the fast pathway can better focus on short and subtle motion patterns while the slow pathway is rather dedicated to the global appearance of the video. [Chen et al., 2019] propose OctConv, which aims at separating 3D vanilla convolutions by high-frequency and low-frequency 3D convolutions. They show that this decomposition allows reducing the time complexity of 3D CNNs while also improving their performance on the Kinetics dataset. More recently, [Lin et al., 2019] proposed TSM, a method that replaces 3D convolutions by a temporal shifting of input channels before applying 2D convolutions. The temporal shifting of the channels followed by a 2D convolution has a similar effect to 3D convolutions but is computationally cheaper than conventional 3D convolutions. On top of that, the authors show that their proposed approach achieves state-of-the-art results on the challenging Something-Something dataset [Goyal et al., 2017], which demonstrates that TSM is moreover efficient at capturing temporal patterns.

Another alternative to decrease the size of video models and their complexity while improving the accuracy is to replace the standard dense 3D convolutions with cheaper group convolutions [Hara et al., 2018] or depth-wise convolutions [Tran et al., 2019; Feichtenhofer, 2020]. These latest video architectures are currently achieving state-of-the-art results on many challenging action recognition benchmarks such as AVA or Kinetics.

All of the reviewed video representations are short-term representations for video inputs in the order of a few seconds. However, in many cases, it is required to consider long-term reasoning for minute-long videos for recognizing more elaborated events or actions. Next, we review longer-term video representation methods.

2.1.4 Longer-term video representations

Designing video representations that go beyond a few seconds is critical for a wide range of applications. For example, only watching a few seconds of a cooking video might not be sufficient to understand what is being cooked or which ingredients are needed to prepare the meal. In action anticipation, it is often crucial to accumulate a long video input of the present to better anticipate what is going to happen next [Sun et al., 2019a; Miech et al., 2019a]. It is also critical to design long-term video representations for video summarization where the goal is to output the set of main labels from minute-long videos [Abu-El-Haija et al., 2016].

In all cases, these long-term video representations model a hierarchical temporal structure. First, short-term video representations are extracted throughout the video to capture temporally low-level semantics. Then follows the aggregation of these short-term representations to form a single representation capturing high-level semantic cues for the whole video.

The simplest form of aggregation is the average or maximum pooling of video features [Wang et al., 2016a] over time. The average and maximum pool quickly becomes sub-optimal when considering long videos with a large number of short-term video representations to aggregate. To overcome this issue, several non-differentiable methods for aggregating vectors over time rely on a codebook usually learnt with unsupervised clustering algorithms. Each feature of the video is encoded using this learnt codebook to form the video level representation. These methods include bag-of-visual-words [Csurka et al., 2004; Sivic and Zisserman, 2003], Vector of Locally aggregated Descriptors (VLAD) [Jegou et al., 2010] or Fisher Vectors (FV) [Perronnin and Dance, 2007]. They have been extensively used for aggregating hand-crafted local video descriptors [Laptev et al., 2008; Marszalek et al., 2009; Wang et al., 2011; Wang and Schmid, 2013]. These methods, however, have the following two main limitations. First, they are non-differentiable, which makes it difficult to include them in end-to-end trainable neural architectures. Second, they are orderless aggregation methods, which do not model the sequential order of the features.

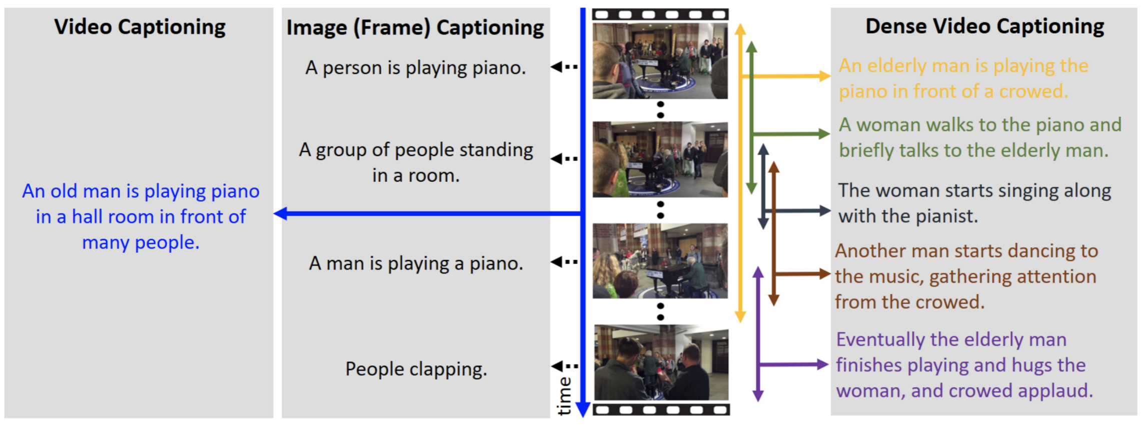
To address the first issue, [Arandjelović et al., 2016; Peng et al., 2014a; Girdhar et al., 2017] explore the use of a differentiable approach for VLAD while [Peng et al., 2014b] study the use of a differentiable FV encoding for action recognition.

To model the sequential nature of videos, many works [Donahue et al., 2014; Ibrahim et al., 2016; Lev et al., 2016; Yue-Hei Ng et al., 2015] rely on the use of recurrent neural networks (RNN), such as Long Short-Term Memory (LSTM) [Hochreiter and Schmidhuber, 1997] or Gated Reccurent Units (GRU) [Cho et al., 2014], which allows modeling sequential patterns as well as an implicit memory. In detail, [Lev et al., 2016] combine the concept of RNN with Fisher Vectors for video feature aggregation. Instead of using RNNs, [Basura et al., 2015] proposed the rank pooling approach, which consists of modeling the evolution of video frames by learning a linear ranking function for ordering frames for all action classes. More recently, [Hussein et al., 2019] proposed the Timeception module, which consists of many lightweight 3D convolutional blocks for aggregating the video features spatio-temporally.

For long videos, it is critical to have an aggregation method capable of encoding all the relevant video descriptors without losing too much information over time. To address this issue, [Wu et al., 2019] introduce the concept of a long-term feature bank that acts as a non-differentiable memory. The feature bank aims at storing relevant video descriptors over time. [Korbar et al., 2019] instead propose to learn a short video clip sampler capable of detecting the salient moments of long videos. Then it only uses a sparse amount of these salients clips for classification.

The first contribution of the thesis, in Chapter 3, is closely related to long-term video representations where we propose an efficient and differentiable approach for video descriptor aggregation for minute-long videos from the YouTube-8M dataset [Abu-El-Haija et al., 2016].

Next, we review works related to joint modeling of video and language.



Video Captioning
An old man is playing piano in a hall room in front of many people.
Image (Frame) Captioning
A person is playing piano.
A group of people standing in a room.
A man is playing a piano.
People clapping.
time
Dense Video Captioning
An elderly man is playing the piano in front of a crowed.
A woman walks to the piano and briefly talks to the elderly man.
The woman starts singing along with the pianist.
Another man starts dancing to the music, gathering attention from the crowed.
Eventually the elderly man finishes playing and hugs the woman, and crowed applaud.

Figure 2-3: The task of video captioning (left part) is to generate a global descriptive caption given an input video. On the other hand, the task of image captioning (middle part) is to generate a descriptive caption for a single image frame. Finally, the task of dense video captioning (right part) is to temporally localize events of interest within a given input video and generate a descriptive caption for each localized event. This illustration was taken from [Aafaq et al., 2019].

2.2 Video and Language

In this section, we review the wide variety of research topics at the interface between video and natural language modeling. We start first with video captioning in section 2.2.1, one of the most popular tasks in the field. Follows the task of video and text retrieval in section 2.2.2. Finally, we review video and text alignment in section 2.2.3, visual grounding for machine translation in section 2.2.3 and video question answering in section 2.2.5.

2.2.1 Video captioning

Video captioning is one typical and popular sub-field of the joint modeling of video and natural language. Given an input video, the task of video captioning is to generate grammatically correct sentences describing the content of the input video. The very left part of Figure 2-3 illustrates an example of video captioning. One practical application of video captioning is for helping visually impaired people to understand the visual content of videos.

Most of the very first and successful works on visual captioning are in the image domain (See Figure 2-3 for an illustration of image captioning). Many of these works [Chen and Zitnick, 2014; Donahue et al., 2014; Vinyals et al., 2015; You et al., 2016] use the combination of a CNN encoder for the image and a RNN for encoding and decoding sentences while others [Ordonez et al., 2011; Fang et al., 2015; Lebret et al., 2015] use non RNN based language models instead. More recently, [Lu et al., 2019a] instead leveraged a transformer [Vaswani et al., 2017] based language model for captioning.

[Johnson et al., 2016] proposed the task of dense captioning, which is to generate bounding boxes around regions of interests of an image and to generate a caption for each bounding box. A large majority of these works rely on a few manually annotated image description datasets such as COCO [Lin et al., 2014], Flickr 8K [Hodosh et al., 2013], Flickr 30K [Young et al., 2014] or Visual Genome [Krishna et al., 2016] well suited for dense captioning.

The evaluation of such models either relies on automated evaluation or human evaluation. However, performing human evaluations on thousands of generated sentences is burdensome as it takes times and is costly, so many works only rely on automated evaluation using metrics such as BLEU [Papineni et al., 2002], ROUGE [Lin, 2004], METEOR [Banerjee and Lavie, 2005] or CIDEr [Vedantam et al., 2015].

Visual captioning models in the video domain are very similar to their image counterparts. Many of these approaches also use CNNs to encode the video frames and RNNs to encode and decode sentences [Pan et al., 2016a; Yu et al., 2016a, 2017b; Wang et al., 2018c]. Slightly different from video captioning, [Zeng et al., 2016] instead propose to only summarize videos with their most salient moment. [Krishna et al., 2017; Shen et al., 2017] focus on the task of dense video captioning where the goal is to localize temporal events of interest and generate a caption for each localized event. More recently, [Zhou et al., 2018c] leveraged a transformer based [Vaswani et al., 2017] language model for dense video captioning. An illustration of dense video captioning can be found in the right part of Figure 2-3. While all of these works rely on supervised learning for training the video captioning model, [Wang et al., 2018b] instead propose to use a reinforcement learning framework for training such models.

The advantage of reinforcement learning is that the models can be trained to optimize the non-differentiable metrics often used when evaluating captioning models instead of maximizing the log-likelihood. Similarly to their image work counterpart, a large majority of these works rely on manually annotated video description datasets such as: MSR-VTT [Xu et al., 2016], MSVD [Chen and Dolan, 2011], YouCook2 [Zhou et al., 2018b] and more recently VATEX [Wang et al., 2019b], which are all YouTube based datasets. [Krishna et al., 2017] also introduced the ActivityNet Captions dataset, which is specifically dedicated to the task of dense video captioning. For more details about video captioning datasets, approaches and evaluation metrics, we invite the reader to check this thorough survey paper [Aafaq et al., 2019].

As described in [Aafaq et al., 2019], automatic evaluation of captioning models is challenging as the models are usually evaluated on how close the generated caption is similar to a groundtruth one or a set of groundtruth captions. However, in many cases, captions that are not part of the groundtruth of a video can also be possible and the usual BLEU, ROUGE, METEOR or CIDEr metrics would fail to measure that. Because evaluating video captioning models is highly subjective to the collected groundtruth, in this thesis, we instead focus our work on the task of text-video retrieval, which can be evaluated in a more objective manner. We review next, works related to text-video retrieval.

2.2.2 Text-Video retrieval

The task of text-video retrieval, often named as the task of text-to-video retrieval, is to retrieve the most relevant video from a large pool of unseen testing videos given an input query in the form of natural language. One of the most popular and natural practical application of retrieval is in video search engines, as illustrated in Figure 2-4. The goal is to enable people to quickly find a video given a short description of it. In this thesis, we have implemented different video search engine demos such as the HowTo100M search engine [2] as the result of Chapter 5 and the YouCook2 search

[2] `https://www.di.ens.fr/willow/research/howto100m/`

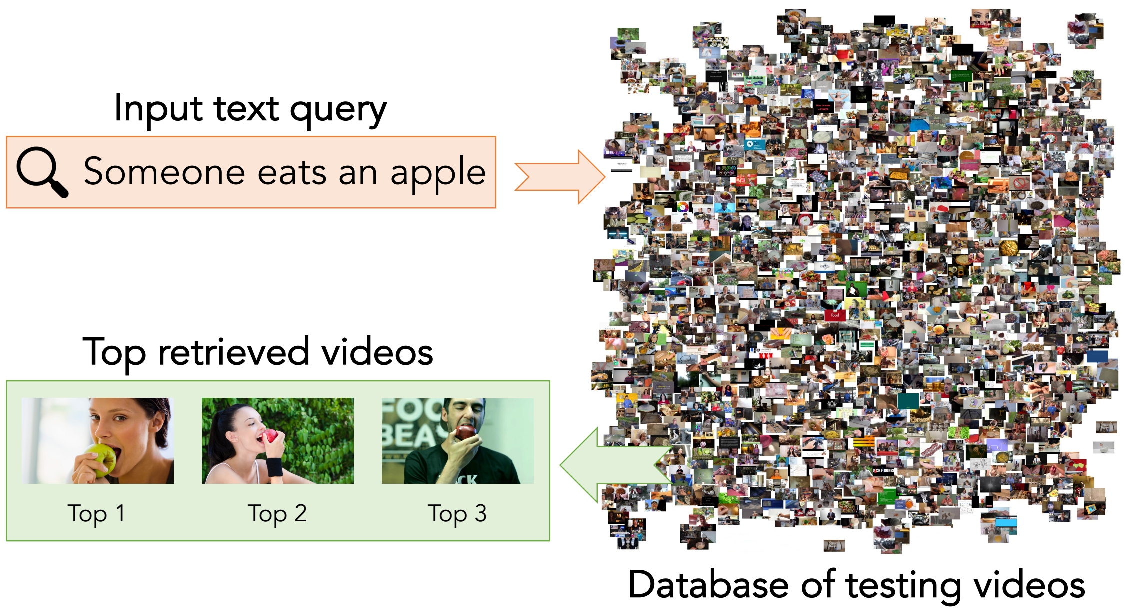


Figure 2-4: The task of text to video retrieval is to retrieve the top videos from a database given an input text query.

engine [3] as the result of Chapter 6.

A large majority of text and visual retrieval approaches [Chowdhury et al., 2018; Dong et al., 2019; Gong et al., 2014a,b; Mithun et al., 2018; Pan et al., 2016b; Xu et al., 2015a; Wang et al., 2018a, 2016b; Wu et al., 2017; Liu et al., 2019] aim at learning a joint text and visual embedding space where visual and textual inputs are close in that space if and only if they are semantically similar. Overall, many neural network based approaches use a bi-directional max-margin ranking loss [Karpathy et al., 2014a] while non neural network based approaches tend to rely on a Canonical Correlation Analysis (CCA) [Hotelling, 1992] training objective. Instead of learning a joint text-video embedding, [Yu et al., 2018] propose an early fusion approach of all video frames to all words by constructing a spatio-temporal correlation tensor of the video with the input sentence. This tensor is then used to compute the overall similarity scores between the video and the textual description.

One clear advantage of the text-video retrieval task over video captioning is that

[3] `https://www.di.ens.fr/willow/research/mil-nce/`

the retrieval evaluation metrics are less subjective than the captioning ones. In fact, the retrieval metrics measure the rank of the retrieved results of each input query. A good retrieval method would minimize the rank of each retrieved result. It is a more objective metric than the ones used in video captioning which are highly sensitive to the groundtruth descriptions. For example, a captioning model would be penalized when generating the caption `A man gets in a vehicle` if the groundtruth caption is `A man enters a car`. While the two sentences have several words in difference, they, however, share the same semantic and current captioning metrics fail to take into account that different sentences can have the same meaning.

A sub-field of text-to-video retrieval instead consists of retrieving and localizing a moment in a long video given an input text query. For that purpose, [Hendricks et al., 2017] introduced the DiDeMo dataset with precise temporal annotations of the events, additionally to the existing ActivityNet Captions [Krishna et al., 2017] and the Charades-STA [Gao et al., 2017] datasets. The retrieval localization method was then significantly improved in [Zhang et al., 2019] with the idea of an iterative graph adjustment network for the localization of moments. Differently to [Hendricks et al., 2017; Zhang et al., 2019], [Mithun et al., 2019] propose an approach for moment retrieval in a weakly-supervised setup, where no localization annotation is used at training. DiDeMo was later extended in [Hendricks et al., 2018] with input queries that aim at measuring the temporal compositionality of events. For instance, a query from this dataset looks like *Someone is clapping after standing up* instead of *Someone is clapping.*

All of these works have in common training on manually annotated video description datasets. In this thesis, we instead focus on learning such joint text-video embeddings from readily available and uncurated data in Chapter 5 and Chapter 6.

2.2.3 Text and Video alignment

Another useful task in video and text, similar to the video moment retrieval task, consists of aligning videos with natural language data. It is useful when a long video comes with an unaligned description of it and one wishes to synchronize the

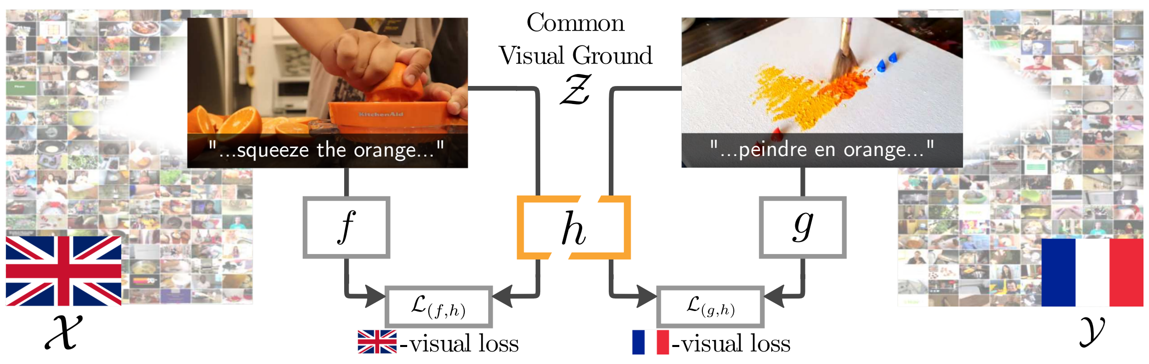


Figure 2-5: The visual world can be used as a common ground for different languages without requiring manual annotation. This illustration was taken from [Sigurdsson et al., 2020].

video with its description. For example, [Plummer et al., 2017] propose to perform video summarization by aligning the salient moments of a video to an input textual description.

Video and language alignment is particularly well studied in the domain of cooking videos [Bojanowski et al., 2015] and more generally in instructional videos [Alayrac et al., 2016; Huang et al., 2016; Richard et al., 2017] where the goal is to align step by step instructions to their corresponding moments in the video.

Other works have studied the alignment of movie screenplays to movies [Cour et al., 2008; Laptev et al., 2008]. [Tapaswi et al., 2015] instead address the problem of aligning movies with extracts from their book version.

In this thesis, Chapter 4 reuses the work of [Laptev et al., 2008] to align movies scripts to the video for action and person recognition while Chapter 6 aims at addressing the temporal misalignment between instructional videos and their narration transcripts for learning video representations.

2.2.4 Visual grounding for translation

A recent and emergent field in jointly modeling video and natural language is for machine translation. One issue in machine translation is that many translation approaches rely on a huge amount of training examples of paired multilingual sentences.

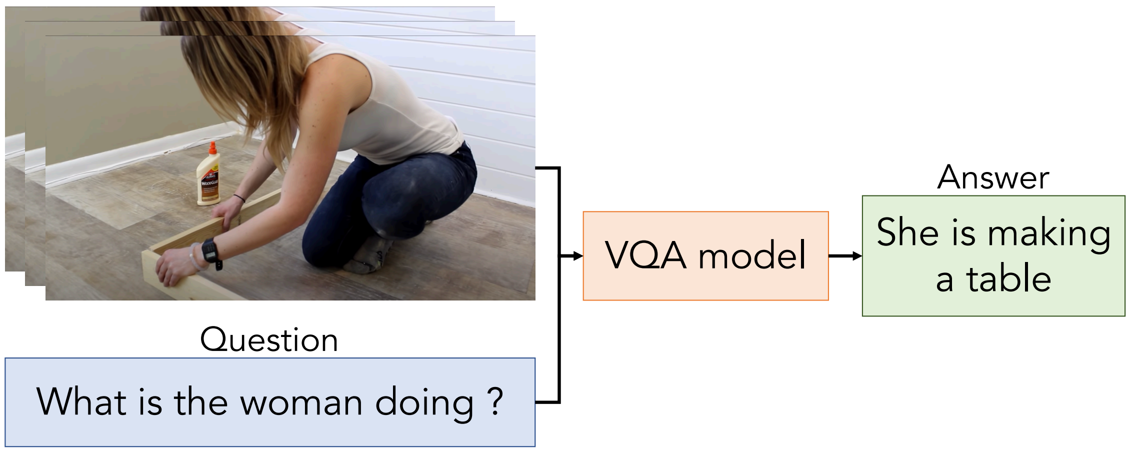


Figure 2-6: The task of video question answering is to answer a question about the content of the input video.

This is not such a problem for paired high-resource languages such as English and French, but it becomes quickly problematic for low-resource languages such as Tamil or Urdu. One way to address this issue is by grounding in the visual world to bridge the gap between different languages, as illustrated in Figure 2-5.

In that direction, [Wang et al., 2019b] proposed the VATEX dataset, which is video captioning dataset with both English and Chinese descriptions of videos. [Sanabria et al., 2018] instead propose to leverage English narration from instructional videos and a translation of these narrations to Portuguese in their How2 video dataset. These two works, unfortunately, rely on manually annotated data hard to get at scale as they either require collecting video descriptions in multiple languages or translating sentences. To address this issue, [Sigurdsson et al., 2020] propose to collect the HowTo-World dataset, an extension of our HowTo100M dataset from Chapter 5 with English, French, Korean and Japanese narrated videos. The method thus does not rely on any manually annotated data and can scale to millions of training examples from different languages. In particular, [Sigurdsson et al., 2020] propose to improve word translation by leveraging the Text-Video retrieval-based approach from Chapter 6 and they show improvement in unsupervised word translation compared to language-based only methods. Figure 2-5 illustrates the proposed approach.

2.2.5 Video question answering

Another task which involves natural language in video understanding is Video Question Answering (VQA). VQA consists of watching a video and answering a question about its content, as illustrated in Figure 2-6. VQA can be especially useful for helping visually impaired people to better understand the content of a video through questions and answers. Most of the initial and influential works in free-form and open-ended visual question answering are in the image domain [Antol et al., 2015; Malinowski and Fritz, 2014; Malinowski et al., 2015; Fukui et al., 2016; Zhu et al., 2016] with manually annotated image question-answering datasets such as COCO-QA [Lin et al., 2014], DAQUAR [Malinowski and Fritz, 2014], Visual7W [Zhu et al., 2016] or VQA [Antol et al., 2015].

In contrast to image question answering, less work has explored question answering for videos, notably because there are fewer existing benchmarks in video. For example, [Tapaswi et al., 2016] propose the MovieQA dataset, a collection of movies with the corresponding plot, subtitles, scripts and a set of questions with multiple choice answers. This benchmark focuses on the global understanding of the plot of the movies rather than what is visually seen. Thus the scripts and subtitles often play a more important role than the video. In contrast to MovieQA, [Lei et al., 2018] propose the TVQA dataset a TV shows based dataset which instead focuses on the understanding of the visual content rather than the plot. The most popular and largest video question answering dataset is TGIF-QA dataset [Jang et al., 2017] which is composed of short 71k GIFs instead of videos. The dataset has five different benchmarks to assess different video understanding capabilities: the counting task, the repeated action task (*i.e.* how many time an action is repeated), the state transition task (*e.g. What does the bear do after sitting ? Stand*) and the open-ended FrameQA task where most questions can be answered by only looking at a single frame.

To address the lack of video datasets for question answering, [Zeng et al., 2017; Xu et al., 2017] propose to generate questions and answers for videos already annotated with textual descriptions. They leverage question generation models based on textual description only [Heilman and Smith, 2010] to generate questions and answers for each video. In particular, [Xu et al., 2017] apply their method on the MSR-VTT [Xu et al., 2016] and the MSVD [Chen and Dolan, 2011] video description datasets to generate their video question answering counterparts: MSR-VTT-QA and MSVD-QA [4].

Most of the methods in image or video question answering follow a similar approach [Malinowski and Fritz, 2014; Zhu et al., 2016; Antol et al., 2015; Fukui et al., 2016], which consists of encoding the visual data with a CNN and the question with an RNN. Then the encoded image and questions are fused to form a single representation that is later on used to perform a classification over the set of most frequent answers or by generating the answer word by word. [Hu et al., 2018] instead, propose an approach similar to retrieval where the inputs $(video, question)$ and the $answer$ are embedded into a joint space.

In this thesis, we do not tackle the task of video question answering but we propose approaches that share many similarities in the way video and language representations are combined.

We have seen through this related work section the wide range of tasks involving video and natural language and all of the useful practical applications. Notably, leveraging natural language enables to incorporate more semantically nuanced supervision than simple categories. However, one drawback of the reviewed approaches is that they mostly rely on few manually annotated videos with annotation in the form of natural language. This is problematic as annotating videos with natural language is cumbersome and thus few publicly available and large-scale annotated videos with natural language exist. As an illustration, Figure 2-7 shows the size of video datasets with annotation in the form of natural language compared to video action recognition datasets with simple labels as annotation. The figure clearly illustrates that the scale of the video datasets with natural language annotations is overall significantly smaller than their counterpart with label annotations. To address this issue, one alternative to the annotation of video with natural language is to leverage readily available data

[4]https://github.com/xudejing/video-question-answering

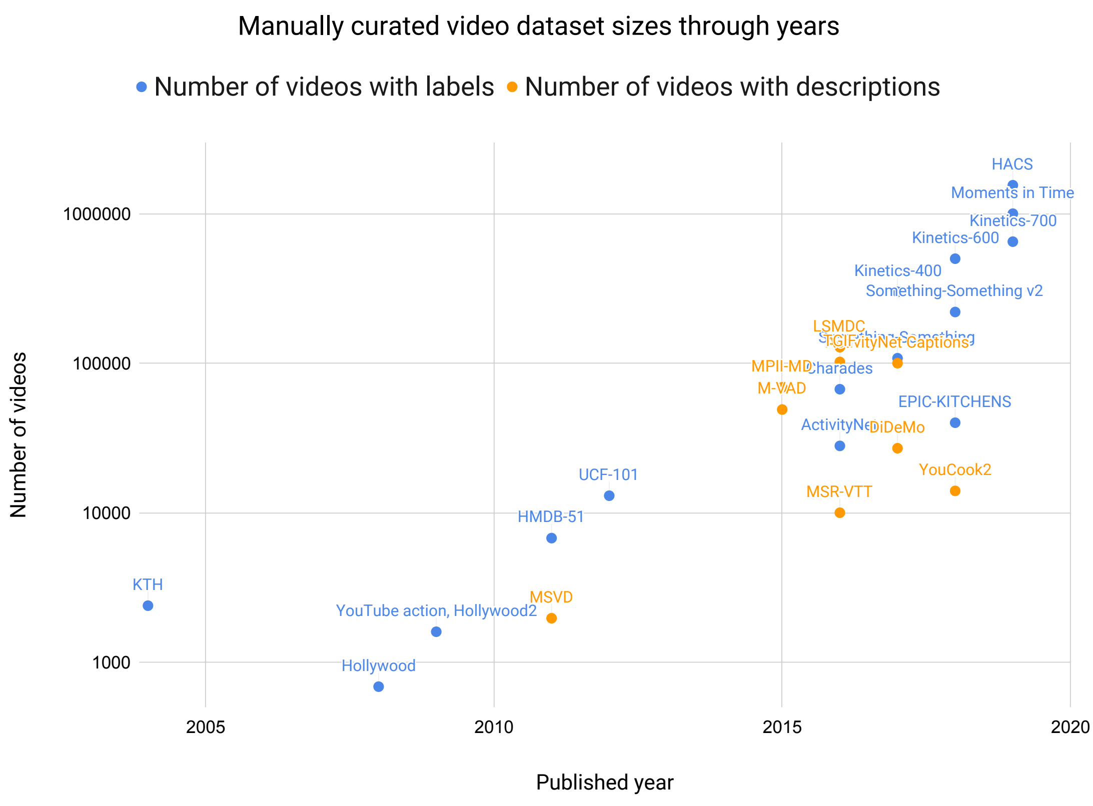


Figure 2-7: A summary of manually curated video dataset sizes through years. The blue points represent video datasets with class labels while the orange points represent video datasets with annotation in the form of natural language descriptions.

instead. Next, we review works exploiting readily available data for the supervision of video models and the challenges implied with these weaker annotations.

2.3 Learning from readily available data

We review here, works that have considered using readily available data instead of manually annotated data for the supervision of visual models.

2.3.1 Self-supervised learning

Self-supervised learning consists of using the data itself for supervision through a pretext task. It is thus the simplest form using readily available data for supervision. Recent work in the image domain has demonstrated great improvements over fully-supervised approaches in the learning of visual representations using contrastive learning [He et al., 2019; Chen et al., 2020]. In the video domain, many other works have considered self-supervised learning using only the video as supervision [Lee et al., 2017; Misra et al., 2016; Tian et al., 2019; Xu et al., 2019; Jing and Tian, 2018; Han et al., 2019] but none of them have reported equivalent successes to the image domain. Instead, the top performing self-supervised video representation approaches also exploit the available audio data through the tasks of video to audio matching [Arandjelović and Zisserman, 2017; Alwassel et al., 2019] or synchronization [Korbar et al., 2018; Piergiovanni et al., 2020].

In this thesis, we propose an approach similar to self-supervised learning in Chapter 6. We learn a video representation from narrated videos through the pretext task of matching videos to speech transcripts instead. While this is not exactly a self-supervised approach as the method relies on a speech recognition model trained on manually annotated speech data, our pretext task shows a valuable benefit. In fact, our pretext task is not only useful for learning a visual representation but also for learning a joint text-video representation without manual annotation. This is valuable as all the self-supervised visual approaches eventually require a few manually annotated data for target downstream tasks such as classification or detection while our method can be used out of the box without manual annotation on downstream tasks such as text-video retrieval or classification. We will next review work also using readily-available speech for supervising video models.

2.3.2 Learning from speech

Many works have explored using available speech from videos as a supervision signal. This has become recently possible thanks to recent significant advances in automatic speech recognition [Wang et al., 2020; Synnaeve et al., 2019].

A large majority of these works focus on narrated instructional videos [Alayrac et al., 2016; Sener et al., 2015; Malmaud et al., 2015; Sanabria et al., 2018; Yu et al., 2014; Sun et al., 2019b]. In fact, in these videos, the narration tends to describe what

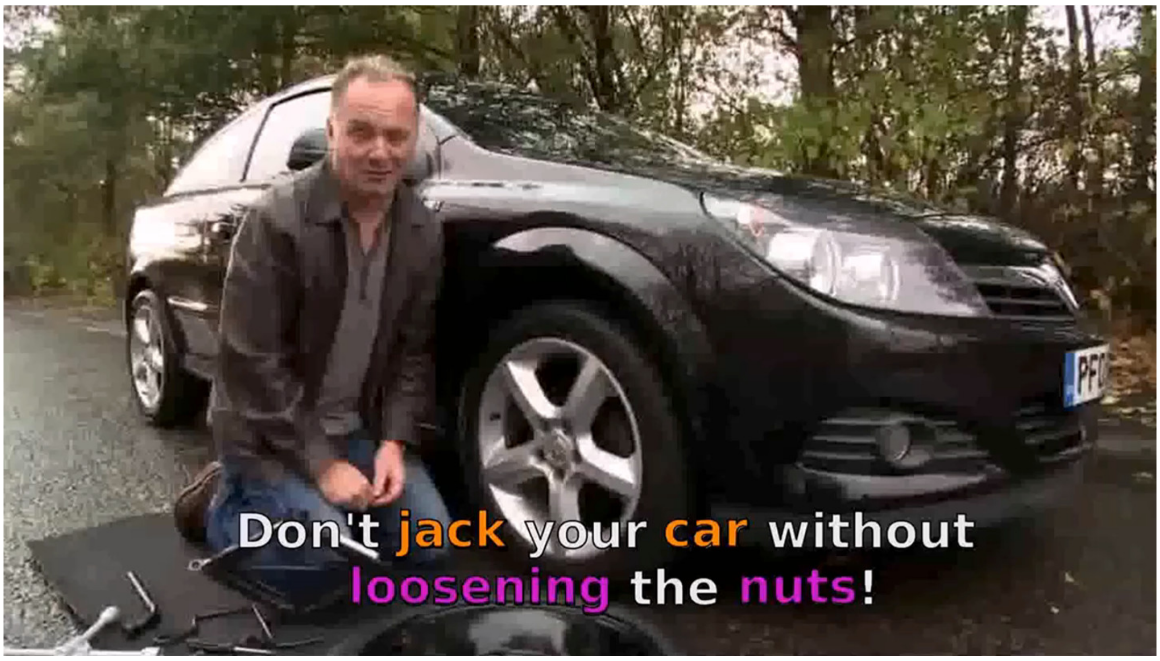


Figure 2-8: An illustration of an instructional video with its narration transcript.

is visually seen on the video for demonstration purposes as illustrated in Figure 2-8. There is thus a strong correlation between the narration and the video, which can be leveraged for supervising visual models.

For example, [Alayrac et al., 2016; Sener et al., 2015] leverage the narration of such videos to automatically discover and localize the key steps of a given a task. [Yu et al., 2014] propose to harvest action examples using narrations for action recognition. [Sanabria et al., 2018] propose to tackle the task of machine translation and video summarization. In particular, [Malmaud et al., 2015] focuses on cooking instructional videos and propose to learn a model to align cooking recipes to their corresponding instructional videos. [Sun et al., 2019b] also focuses on cooking videos and propose VideoBERT: a joint model for video and language. They use a large number of videos with narrations for pre-training their cross-modal Transformer [Vaswani et al., 2017].

One issue with these prior works is: either the dataset only contains cooking videos [Malmaud et al., 2015; Sun et al., 2019b], is small [Alayrac et al., 2016; Sener et al., 2015; Yu et al., 2014; Sanabria et al., 2018] or not publicly available [Sun et al., 2019b]. We address this issue in Chapter 5 by introducing the HowTo100M dataset, a large-scale collection of 1.2M of uncurated narrated instructional videos from a wide range of more than 23k different activities. We showed we can use HowTo100M to learn a text-video embedding in Chapter 5 and a state-of-the-art video representation in Chapter 6. HowTo100M was later extended to HowTo-World in [Sigurdsson et al., 2020], a collection also containing narrated videos in French, Japanese and Korean for the purpose of unsupervised machine translation through visual grounding.

Apart from instructional videos, [Chen et al., 2017a] use speech from documentaries to automatically obtain object labels. They focus on learning an object detector from a dataset containing 9 documentary movies. And more recently, [Nagrani et al., 2020] exploit the speech from a large-scale amount of 288K movies to mine action examples for simple classes such as running, kissing or dancing. We will see next that movies and TV series are also a great source of readily available annotation in the form of natural language.

2.3.3 Movie scripts, subtitles and DVS

Movie scripts or screenplays are also a great source of readily available [5] annotations in the form of natural language. They aim at describing precisely how the scenes should look like before shooting them. They are thus perfectly suited for supervising visual models as illustrated in Figure 2-9. One of the earlier works [Laptev et al., 2008] leveraged movie scripts as a way to mine action clips from Hollywood movies in a time where action recognition video datasets are small and scarce. [Duchenne et al., 2009] later on propose to perform localization of actions using scripts from the Hollywood2 dataset [Laptev et al., 2008]. An extension of this work [Bojanowski et al., 2014] exploits the temporal order of actions as a learning constraint. [Bojanowski et al., 2013; Everingham et al., 2006; Sivic et al., 2009; Parkhi et al., 2015a] propose to also recognize the actors given in movie or TV series scripts as they also contain the identity of the actor speaking (see Figure 2-9). Our contribution from Chapter 4 extends and improves [Bojanowski et al., 2013] by proposing a large-scale algorithm

[5] They are publicly available on websites such as `https://www.imsdb.com/` or `https://www.weeklyscript.com/`

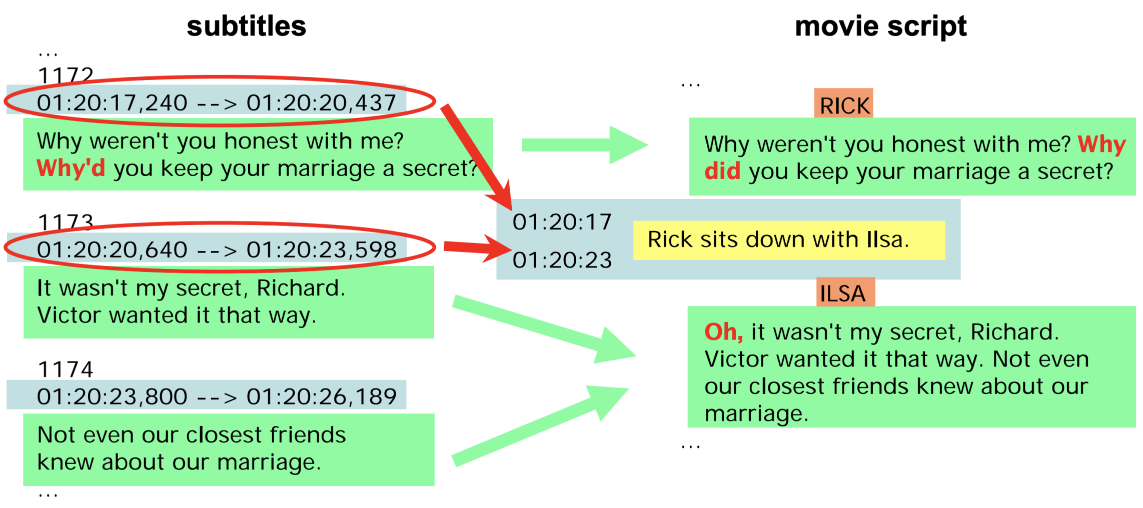

subtitles
movie script
...
1172
01:20:17,240 --> 01:20:20,437
Why weren't you honest with me?
Why'd you keep your marriage a secret?
1173
01:20:20,640 --> 01:20:23,598
It wasn't my secret, Richard.
Victor wanted it that way.
1174
01:20:23,800 --> 01:20:26,189
Not even our closest friends
knew about our marriage.
...
...
RICK
Why weren't you honest with me? Why did you keep your marriage a secret?
01:20:17
01:20:23
Rick sits down with Ilsa.
ILSA
Oh, it wasn't my secret, Richard. Victor wanted it that way. Not even our closest friends knew about our marriage.
...

Figure 2-9: An extract of a script from the Casablanca movie synchronized using the subtitles. The script provides the information of what is being seen and also which person is speaking. The illustration was taken from [Laptev et al., 2008].

that can scale to significantly more data.

Additionally to movie scripts, [Torabi et al., 2015; Rohrbach et al., 2015] introduce the M-VAD and the MPII movie description datasets which also leverage Descriptive Video Service (DVS). DVS is an audio narration describing the visual elements and actions in a movie for the visually impaired, which makes them perfectly suitable for supervising video models. They are readily available in many movie DVDs which makes their collection cheap and fast. These movie datasets with perfectly aligned descriptions are used for many language tasks such as retrieval, captioning or multiple choice answers [Torabi et al., 2016; Yu et al., 2016b; Kauman et al., 2017; Yu et al., 2017b, 2018].

More recently, [Liu et al., 2020; Lei et al., 2020; Nagrani et al., 2020] propose to leverage subtitles from popular TV shows and movies. As opposed to movie scripts that contain rich visual information about a movie, the subtitles mostly inform about the plot. However, subtitles can be obtained from any movies or TV series as opposed to movie scripts. The collection of subtitles can be thus scaled to several order of magnitude higher than movie scripts. For example, [Nagrani et al., 2020] leverage an internal collection of 288k movies with subtitles while the largest publicly available

Alt-text: A Pakistani worker helps to clear the debris from the Taj Mahal Hotel November 7, 2005 in Balakot, Pakistan.

Conceptual Captions: a worker helps to clear the debris.

Alt-text: Musician Justin Timberlake performs at the 2017 Pilgrimage Music & Cultural Festival on September 23, 2017 in Franklin, Tennessee.

Conceptual Captions: pop artist performs at the festival in a city.

Figure 2-10: The Conceptual Caption dataset leverages alternative text attributes (Alt-text) from HTML pages to mine clean descriptive captions for millions of images. This illustration was taken from [Sharma et al., 2018].

movie scripts database [6] contains roughly 1k scripts.

2.3.4 Metadata from the internet

Titles, descriptions and tags. User uploaded medias such as images or videos often come with descriptive metadata such as a title, tags and even a short description. Such readily available metadata can be used as a form of weak annotation for a large amount of uploaded images or videos. Moreover, these images or videos can be found at an unprecedented scale since billions of images or videos are uploaded on social media such as Flickr, YouTube or Instagram.

For example, [Thomee et al., 2016] propose the YFCC100M, a collection of 99 million images and 800k videos from Flickr with diverse metadata collected from title, descriptions and tags. [Ordonez et al., 2011] also leverage Flickr to mine a collection of 1 million photographs with clean descriptive captions. [Abu-El-Haija et al., 2016] introduce the YouTube-8M dataset, a dataset of more than 7 million minute-long

[6] `https://www.imsdb.com/`

YouTube videos with machine-generated weak visual labels from metadata such as titles and descriptions. In Chapter 3, we specifically focus on learning a long-term video representation on this dataset. More specifically related to sports, [Karpathy et al., 2014b] propose the Sports-1M dataset of sports videos from YouTube, also annotated with machine-generated sports labels from metadata. [Zeng et al., 2016] leverage the titles of YouTube videos for the task of title generation. One advantage of producing a title for a long video is that they tend to describe the most salient event in the video as opposed to describing a video as a whole.

One issue with the above datasets is that they provide coarsely localized labels for minute-long videos. This makes them not suitable for learning visual representations. Instead, the learning of visual representations has been specially studied with images, which do not suffer from any localization issue. For example, [Sun et al., 2017] leverage the Google Image Search engine to mine 300 million images with weak labels. [Mahajan et al., 2018b] goes further by collecting a dataset of more than 1 billion Instagram images with visual hashtags. More recently, an extension of this work [Ghadiyaram et al., 2019] in the video domain collected a dataset of 65M videos with visual hashtags. All of these works [Sun et al., 2017; Mahajan et al., 2018b; Ghadiyaram et al., 2019] demonstrated significant improvements when pretraining visual models on such large-scale, yet weakly labeled datasets. Our main contribution from Chapter 6 also shows significant improvements by pretraining a video representation on millions of video clips with annotation in the form of speech transcripts. One advantage of leveraging speech transcripts over metadata such as titles, descriptions or tags [Abu-El-Haija et al., 2016; Ghadiyaram et al., 2019] is the precise temporal localization that comes with the transcripts.

Alternative text HTML attributes. Recent works [Sharma et al., 2018; Qi et al., 2020] have exploited readily available alternative (alt) text html attributes from popular web pages. The alt text attribute specifies an alternative text for an image that cannot be displayed in HTML web pages [7]. They are hence likely to describe

[7]https://moz.com/learn/seo/alt-text

the content of their corresponding images, as illustrated in Figure 2-10. One other main advantage of this approach is that the data collection can scale up to millions of examples. For example, [Sharma et al., 2018] introduced the Conceptual Caption dataset with more than 3.3 million images and [Qi et al., 2020] introduced LAIT dataset with more than 10 million images. The images all come with clean captions generated from alt text attributes, as illustrated in Figure 2-10.

These datasets are 10$\times$ to 100$\times$ larger than manually annotated image description datasets such as COCO [Lin et al., 2014], Flickr 30K [Young et al., 2014] or Visual Genome [Krishna et al., 2016], making them particularly suited for large-scale vision and language pretraining methods [Lu et al., 2019a,b; Qi et al., 2020].

Chapter 3

Learnable Pooling with Context Gating for Video Understanding

In this chapter, we tackle the problem of learning a video feature aggregation method for learning long-term video representations. Having an efficient video representation is a key prerequisite for designing strong video models for various applications. That is why we will reuse building blocks from this chapter later in the video representation described in Chapter 5.

In detail, this chapter will first explore clustering-based aggregation layers and propose a two-stream architecture aggregating audio and visual features. We then introduce a learnable non-linear unit, named Context Gating, aiming to model interdependencies among network activations. Our experimental results show the advantage of our video representation for the tasks of text-video retrieval and video classification. In particular, we evaluate our method on the large-scale weakly-labeled Youtube-8M dataset for classification and the MPII Movie Description dataset for text-video retrieval. Our proposed approach outperforms state-of-the-art models on both datasets and tasks.

3.1 Introduction

Understanding and recognizing video content is a major challenge for numerous applications including surveillance, personal assistance, smart homes, autonomous driving, stock footage search and sports video analysis. In this chapter, we address both the problems of multi-label video classification for user-generated videos on the Internet and text-video retrieval when working with a set of pre-extracted temporal video features. The analysis of such data involves several challenges. Internet videos have a great variability in terms of content and quality (see Figure 3-1). Moreover, user-generated labels are typically incomplete, ambiguous and may contain errors.

Current approaches for video analysis typically represent videos by features extracted from consecutive frames, followed by feature aggregation over time. Example methods for feature extraction include deep convolutional neural networks (CNNs) pre-trained on static images [He et al., 2016; Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; Szegedy et al., 2016]. Representations of motion and appearance can be obtained from CNNs pre-trained for video frames and short video clips [Tran et al., 2015; Feichtenhofer et al., 2016], as well as hand-crafted video features [Laptev et al., 2008; Schüldt et al., 2004; Wang and Schmid, 2013]. Other more advanced models employ hierarchical spatio-temporal convolutional architectures [Baccouche et al., 2011; Carreira and Zisserman, 2017; Feichtenhofer et al., 2017; Ji et al., 2013; Tran et al., 2015; Varol et al., 2017] to both extract and temporally aggregate video features at the same time.

Common methods for temporal feature aggregation include simple averaging or maximum pooling as well as more sophisticated pooling techniques such as VLAD [Jegou et al., 2010] or more recently recurrent models (LSTM [Hochreiter and Schmidhuber, 1997] and GRU [Cho et al., 2014]). These techniques, however, may be suboptimal. Indeed, simple techniques such as average or maximum pooling may become inaccurate for long sequences. Recurrent models are frequently used for temporal aggregation of variable-length sequences [Donahue et al., 2014; Abu-El-Haija et al., 2016] and often outperform simpler aggregation methods, however, their training re-

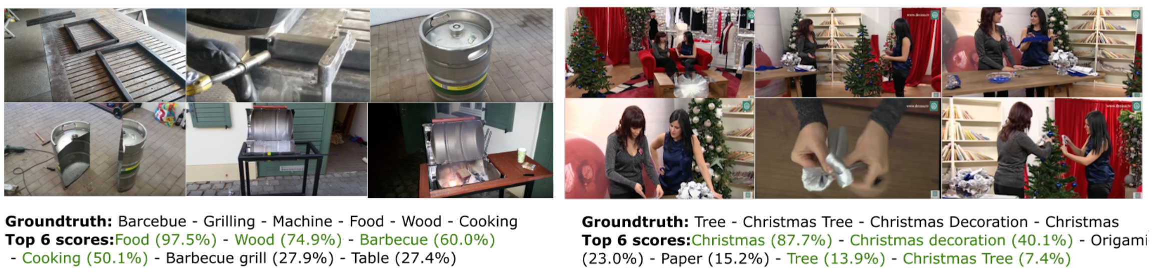

Groundtruth: Barcebue - Grilling - Machine - Food - Wood - Cooking
Top 6 scores:Food (97.5%) - Wood (74.9%) - Barbecue (60.0%) - Cooking (50.1%) - Barbecue grill (27.9%) - Table (27.4%)
Groundtruth: Tree - Christmas Tree - Christmas Decoration - Christmas
Top 6 scores:Christmas (87.7%) - Christmas decoration (40.1%) - Origami (23.0%) - Paper (15.2%) - Tree (13.9%) - Christmas Tree (7.4%)

Figure 3-1: Two example videos from the Youtube-8M V2 dataset together with the ground truth and top predicted labels. Predictions colored as green are labels from the groundtruth annotation.

mains cumbersome. As we show in Section 3.5, training recurrent models requires relatively large amount of data. Moreover, recurrent models can be sub-optimal for processing of long video sequences during GPU training. It is also not clear if current models for sequential aggregation are well-adapted for video representation. Indeed, our experiments with training recurrent models using temporally-ordered and randomly-ordered video frames show similar results.

Another research direction is to exploit traditional orderless aggregation techniques based on clustering approaches such as Bag-of-visual-words [Csurka et al., 2004; Sivic and Zisserman, 2003], Vector of Locally Aggregated Descriptors (VLAD) [Jegou et al., 2010] or Fisher Vectors [Perronnin and Dance, 2007]. It has been recently shown that integrating VLAD as a differentiable module in a neural network can significantly improve the aggregated representation for the task of place retrieval [Arandjelović et al., 2016]. This has motivated us to integrate and enhance such clustering-based aggregation techniques for the task of video representation and classification.

Contributions. In this chapter, we make the following contributions: (i) we introduce a new state-of-the-art architecture aggregating video and audio features for video representation, (ii) we introduce the Context Gating layer, an efficient non-linear unit for modeling interdependencies among network activations, and (iii) we experimentally demonstrate benefits of our approach by ranking first at the Youtube 8M large-scale video classification challenge over 655 teams and achieving state-of-the-art results of the MPII Movie Description dataset on both Text-to-Video and Video-to-Text retrieval tasks.

Results. We evaluate our method on the large-scale multi-modal Youtube-8M dataset containing about 8M videos and 4716 unique tags. We use pre-extracted visual and audio features provided with the dataset [Abu-El-Haija et al., 2016] and demonstrate improvements obtained with the Context Gating as well as by the combination of learnable poolings. Our method obtains top performance, out of more than 650 teams, in the Youtube-8M Large-Scale Video Understanding challenge[1]. Compared to the common recurrent models, our models are faster to train and require less training data. Figure 3-1 illustrates some qualitative results of our method. Moreover, We have experimented our approach on the tasks of Text-to-Video and Video-to-Text retrieval on the MPII Movie Description dataset [Rohrbach et al., 2015] and reached state-of-the-art results by leveraging our Context Gating module for representation.

3.2 Related work

This chapter is related to previous methods for video feature extraction, aggregation and gating reviewed below.

3.2.1 Feature extraction

Successful hand-crafted representations [Laptev et al., 2008; Schüldt et al., 2004; Wang and Schmid, 2013] are based on local histograms of image and motion gradient orientations extracted along dense trajectories [de Souza et al., 2016; Wang and Schmid, 2013]. More recent methods extract deep convolutional neural network activations computed from individual frames or blocks of frames using spatial [Feichtenhofer et al., 2016; Karpathy et al., 2014b; Girdhar et al., 2017; Wang et al., 2015] or spatio-temporal [Baccouche et al., 2011; Carreira and Zisserman, 2017; Feichtenhofer et al., 2017; Ji et al., 2013; Tran et al., 2015; Varol et al., 2017] convolutions. Con-

[1]`https://www.kaggle.com/c/youtube8m`

volutional neural networks can be also applied separately on the appearance channel and the pre-computed motion field channel resulting in the, so called, two-stream representations [Carreira and Zisserman, 2017; Feichtenhofer et al., 2016; Girdhar et al., 2017; Simonyan and Zisserman, 2014; Varol et al., 2017]. As the chapter is motivated by the Youtube-8M large-scale video understanding challenge [Abu-El-Haija et al., 2016], we will assume for the rest of the chapter that features are provided (more details are provided in Section 3.5) and we will mainly focus on the temporal aggregation of the provided features.

3.2.2 Feature aggregation

Video features are typically extracted from individual frames or short video clips. The remaining question is: how to aggregate video features over the entire and potentially long video? One way to achieve this is to employ recurrent neural networks, such as long short-term memory (LSTM) [Hochreiter and Schmidhuber, 1997] or gated recurrent unit (GRU) [Cho et al., 2014]), on top of the extracted frame-level features to capture the temporal structure of video into a single representation [Basura et al., 2015; Donahue et al., 2014; Ibrahim et al., 2016; Lev et al., 2016; Yue-Hei Ng et al., 2015]. Hierarchical spatio-temporal convolution architectures [Baccouche et al., 2011; Carreira and Zisserman, 2017; Feichtenhofer et al., 2017; Ji et al., 2013; Tran et al., 2015; Varol et al., 2017] can also be viewed as a way to both extract and aggregate temporal features at the same time. Other methods capture only the *distribution* of features in the video, not explicitly modeling their temporal ordering. The simplest form of this approach is the average or maximum pooling of video features [Wang et al., 2016a] over time. Other commonly used methods include bag-of-visual-words [Csurka et al., 2004; Sivic and Zisserman, 2003], Vector of Locally aggregated Descriptors (VLAD) [Jegou et al., 2010] or Fisher Vector [Perronnin and Dance, 2007] encoding. Application of these techniques to video include [Laptev et al., 2008; Peng et al., 2014a; Schüldt et al., 2004; Wang and Schmid, 2013; Xu et al., 2015b]. Typically, these methods [Lev et al., 2016; Perronnin and Larlus, 2015] rely on an unsupervised learning of the codebook. However, the codebook can also be learned in a discriminative manner [Peng et al., 2014a,b; Sydorov et al., 2014] or the entire encoding module can be included within the convolutional neural network architecture and trained in the end-to-end manner [Arandjelović et al., 2016]. This type of end-to-end trainable orderless aggregation has been recently applied to video frames in [Girdhar et al., 2017]. Here we extend this work by aggregating visual and audio inputs, and also investigate multiple orderless aggregations.

3.2.3 Gating

Gating mechanisms allow multiplicative interaction between a given input feature X and a gate vector with values in between 0 and 1. They are commonly used in recurrent neural network models such as LSTM [Hochreiter and Schmidhuber, 1997] and GRU [Cho et al., 2014] but have so far not been exploited in conjunction with other non-temporal aggregation strategies such as Fisher Vectors (FV), Vector of Locally Aggregated Descriptors (VLAD) or bag-of-visual-words (BoW). This chapter aims to fill this gap and designs a video classification architecture combining non-temporal aggregation with gating mechanisms. One of the motivations for this choice is the recent Gated Linear Unit (GLU) [Dauphin et al., 2016], which has demonstrated significant improvements in natural language processing tasks.

Our gating mechanism initially reported in [Miech et al., 2017b] is also related to the parallel work on Squeeze-and-Excitation architectures [Hu et al., 2017], that has suggested gated blocks for image classification tasks and have demonstrated excellent performance on the ILSVRC 2017 image classification challenge.

3.3 Network architecture

Our architectures for video classification (Figure 3-2) and Text-Video joint embedding (Figure 3-3) contain three main modules. First, the input features are extracted from video and audio signals. Words embeddings are also extracted for the Text-Video joint embedding model. Next, the pooling module aggregates the extracted features into a single compact (e.g. 1024-dimensional) representation for the entire video (and for

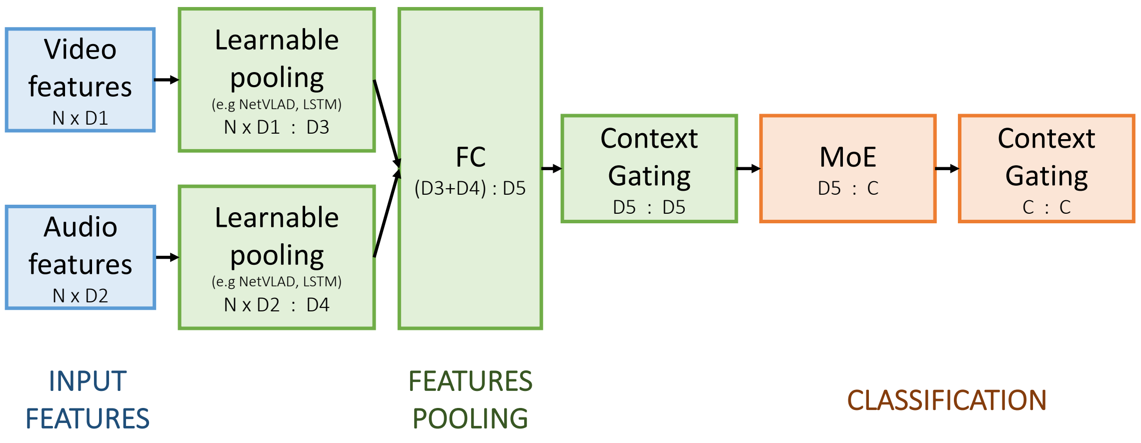


Figure 3-2: Overview of our network architecture for video classification. Below each block is shown the input and output dimensions of each module separated by a colon:. FC denotes a Fully-Connected layer. MoE denotes the Mixture-of-Experts classifier [Abu-El-Haija et al., 2016]. N is the maximum number of features to aggregate, C the number of classes to predict and D5 the dimension of the pooled video representation.

the sentence in the Text-Video joint embedding architecture). The aggregated video and sentence representation are then enhanced by a Context Gating layer (section 3.3.1). Finally, for the classification architecture, a Mixture-of-Experts [Jordan, 1994] classifier as described in [Abu-El-Haija et al., 2016], followed by another Context Gating layer outputs a set of labels for the given video. To train the classification model, the loss used is a standard cross entropy loss. On the other hand, for the Video-Text joint embedding architecture, the sentence representation is compared with the video representation using a cosine similarity. To train the Text-Video model, the loss used is a standard max margin ranking loss [Yu et al., 2017a].

3.3.1 Context Gating

The Context Gating (CG) module transforms the input feature representation X into a new representation Y as

$$Y = \sigma(WX + b) \circ X, \tag{3.1}$$

where $X \in \mathbb{R}^n$ is the input feature vector, σ is the element-wise sigmoid activation and $\circ$ is the element-wise multiplication. $W \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ are trainable parameters. The vector of weights $\sigma(WX + b) \in [0, 1]$ represents a set of learned gates applied to the individual dimensions of the input feature X.

The motivation behind this transformation is two-fold. First, we wish to introduce non-linear interactions among activations of the input representation. Second, we wish to recalibrate the strengths of different activations of the input representation through a self-gating mechanism. The form of the Context Gating layer is inspired by the Gated Linear Unit (GLU) introduced recently for language modeling [Dauphin et al., 2016] that considers a more complex class of transformations given by $\sigma(W_1X + b_1) \circ (W_2X + b_2)$, with two sets of learnable parameters W_1, b_1 and W_2, b_2. Compared to the the Gated Linear Unit [Dauphin et al., 2016], our Context Gating in (3.1) (i) reduces the number of learned parameters as only one set of weights is learnt, and (ii) re-weights directly the input vector X (instead of its linear transformation) and hence is suitable for situations where X has a specific meaning, such the score of a class label, that is preserved by the layer. As shown in Figure 3-2, we use Context Gating in the feature pooling and classification modules. First, we use CG to transform the feature vector before passing it to the classification module. Second, we use CG after the classification layer to capture the prior structure of the output label space. Details are provided below.

3.3.2 Relation to residual connections

Residual connections has been introduced in [He et al., 2016]. They demonstrate faster training of deep convolutional neural networks as well as better performance for a variety of tasks. Residual connections can be formulated as

$$Y = f(WX + b) + X, \tag{3.2}$$

where X are the input features, (W, b) the learnable parameters of the linear mapping (or it can be a convolution), f is a non-linearity (typically Rectifier Linear Unit as expressed in [He et al., 2016]). One advantage of residual connections is the possibility of gradient propagation directly into X during training, avoiding the vanishing gradient problem. To show this, the gradient of the residual connection can be written as:

$$\nabla Y = \nabla(f(WX + b)) + \nabla X. \tag{3.3}$$

One can notice that the gradient ∇Y is the sum of the gradient of the previous layer ∇X and the gradient $\nabla(f(WX + b))$. The vanishing gradient problem is overcome thanks to the term ∇X, which allows the gradient to backpropagate directly from Y to X without decreasing in the norm. A similar effect is observed with Context Gating which has the following gradient equation:

$$\nabla Y = \nabla(\sigma(WX + b)) \circ X + \sigma(WX + b) \circ \nabla X. \tag{3.4}$$

In this case, the term ∇X is weighted by activations $\sigma(WX + b)$. Hence, for dimensions where $\sigma(WX+b)$ are close to 1, gradients are directly propagated from Y to X. In contrast, for values close to 0 the gradient propagation is vanished. This property is valuable as it allows to stack several non-linear layers and avoid vanishing gradient problems.

3.3.3 Relation to LSTM/GRU

It is also worth noticing that an operation similar to Context Gating is also applied to GRU [Cho et al., 2014] and LSTM [Hochreiter and Schmidhuber, 1997] architectures at each iteration. In fact, let us consider the case of a one layer GRU (which is a simpler version than LSTM) network. In the same manner as in [Cho et al., 2014], we denote x_t as feature from iteration t, h_t as the output vector of GRU cell t, z_t the update gate vector, r_t the reset gate vector and W, U, b the GRU parameters to learn.

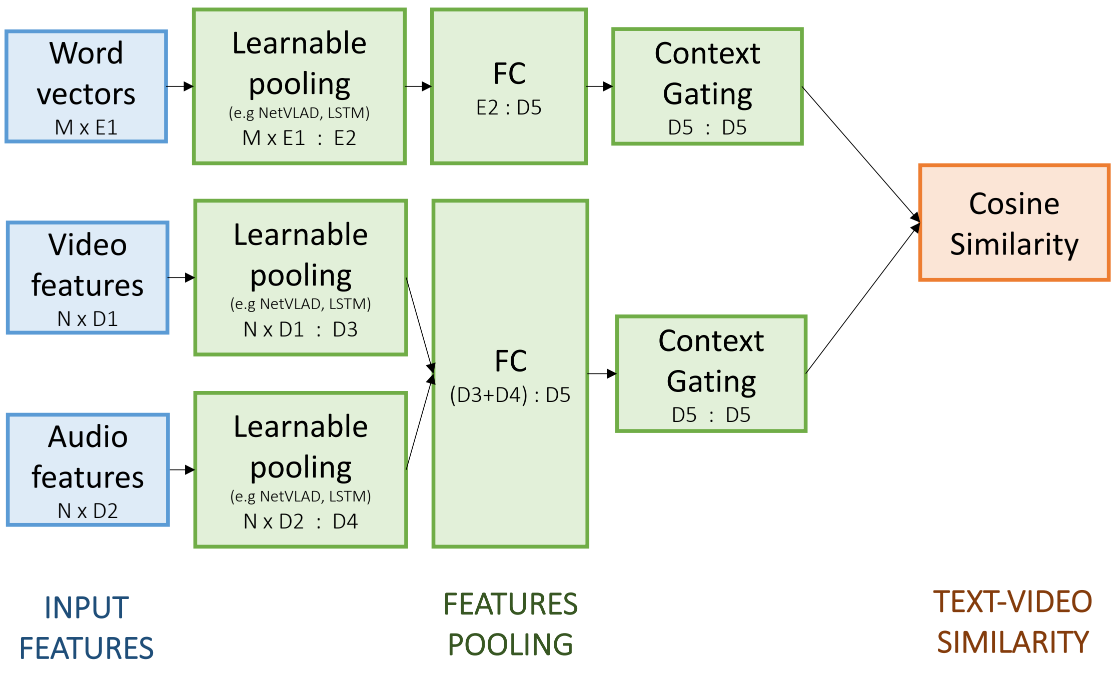


Figure 3-3: Overview of our network architecture for Video-Text joint embedding learning. Below each block is shown the input and output dimensions of each module separated by a colon **:**. FC denotes a Fully-Connected layer. N is the maximum number of video features to aggregate and M the number of word vectors from the input sentence.

The equation of an iteration of a GRU cell is given by:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \tag{3.5}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r), \tag{3.6}$$

$$h_t = z_t \circ h_{t-1} + (1 - z_t) \circ \tau(W_h x_t + U_h(r_t \circ h_{t-1}) + b_h), \tag{3.7}$$

where σ is the sigmoid function and τ is the hyperbolic tangent function. The aggregated video representation in the RNN case is the last hidden state h_t of the RNN. From equation (7), h_t is decomposed into two terms. The first term is $z_t \circ h_{t-1}$, which is equal to $\sigma(W_z x_t + U_z h_{t-1} + b_z) \circ h_{t-1}$. In other word, to compute h_t, an operation similar to Context Gating is applied to h_{t-1} where the gates are computed using both the previous hidden state h_{t-1} and the current sample x_t. For these reasons, we argue that adding Context Gating to aggregation models such as LSTM or GRU is actually

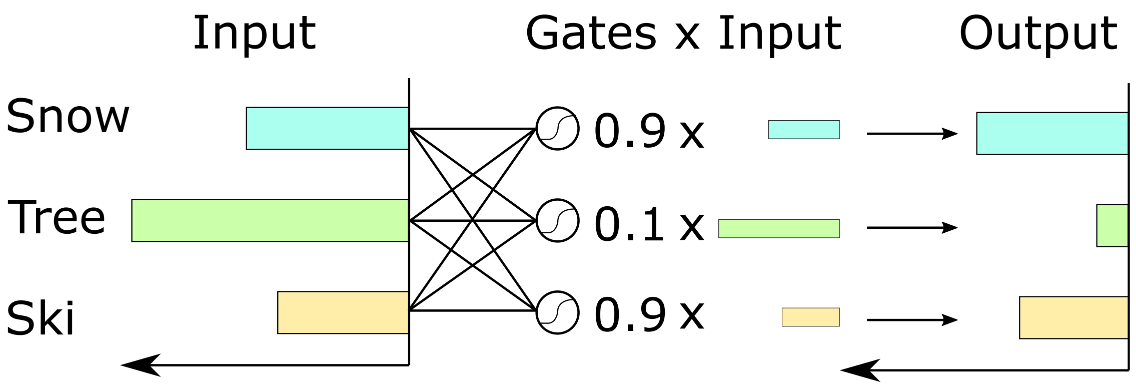


Figure 3-4: Illustration of Context Gating that down-weights visual activations of Tree for a skiing scene.

redundant.

3.3.4 Motivation for Context Gating

Our goal is to predict human-generated tags for a video. Such tags typically represent only a subset of objects and events which are most relevant to the context of the video. To mimic this behavior and to suppress irrelevant labels, we introduce the Context Gating module to re-weight both the features and the output labels of our architecture.

Capturing dependencies among features. Context Gating can help creating dependencies between visual activations. Take an example of a skiing video showing a skiing person, snow and trees. While network activations for the *Tree* features might be high, trees might be less important in the *context* of skiing where people are more likely to comment about the snow and skiing rather than the forest. Context Gating can learn to down-weight visual activations for *Tree* when it co-occurs with visual activations for *Ski* and *Snow* as illustrated in Figure 3-4.

Capturing prior structure of the output space. Context Gating can also create dependencies among output class scores when applied to the classification layer of the network. This makes it possible to learn a prior structure on the output probability space, which can be useful in modeling *biases in label annotations*.

3.4 Learnable pooling methods

Within our video architecture described above, we investigate several types of learnable pooling models, which we describe next. Previous successful approaches [Donahue et al., 2014; Abu-El-Haija et al., 2016] employed recurrent neural networks such as LSTM or GRU for the encoding of the sequential features. We chose to focus on non-recurrent aggregation techniques. This is motivated by several factors: first, recurrent models are computationally demanding for long temporal sequences as it is not possible to parallelize the sequential computation. Moreover, it is not clear if treating the aggregation problem as a sequence modeling problem is necessary. As we show in our experiments, there is almost no change in performance if we shuffle the frames in a random order as almost all of the relevant signal relies on the static visual cues. All we actually need to do is to find a way to efficiently *remember* all of the relevant visual cues. We will first review the NetVLAD [Arandjelović et al., 2016] aggregation module and then explain how we can exploit the same idea to imitate Fisher Vector and Bag-of-visual-Words aggregation scheme.

3.4.1 NetVLAD aggregation

The NetVLAD [Arandjelović et al., 2016] architecture has been proposed for place recognition to reproduce the VLAD encoding [Jegou et al., 2010], but in a differentiable manner, where the clusters are tuned via backpropagation instead of using k-means clustering. It was then extended to action recognition in video [Girdhar et al., 2017]. The main idea behind NetVLAD is to write the descriptor x_i hard assignment to the cluster k as a soft assignment:

$$a_k(x_i) = \frac{e^{w_k^\top x_i + b_k}}{\sum_{j=1}^{K} e^{w_j^\top x_i + b_j}} \tag{3.8}$$

where $(w_j)_j$ and $(b_j)_j$ are learnable parameters. In other words, the soft assignment $a_k(x_i)$ of descriptor x_i to cluster k measures on a scale from 0 to 1 how close the descriptor x_i is to cluster k. In the hard assignment way, $a_k(x_i)$ would be equal to 1 if x_i closest cluster is cluster k and 0 otherwise. For the rest of the chapter, $a_k(x_i)$ will define soft assignment of descriptor x_i to cluster k. If we write $c_j, j \in [1, K]$ the j-th learnable cluster, the NetVLAD descriptor can be written as

$$VLAD(j,k) = \sum_{i=1}^{N} a_k(x_i)(x_i(j) - c_k(j)), \tag{3.9}$$

which computes the weighted sum of residuals $x_i - c_k$ of descriptors x_i from learnable anchor point c_k in cluster k.

3.4.2 Beyond NetVLAD aggregation

By exploiting the same cluster soft-assignment idea, we can also imitate similar operations as the traditional Bag-of-visual-words [Csurka et al., 2004; Sivic and Zisserman, 2003] and Fisher Vectors [Perronnin and Dance, 2007] in a differentiable manner.

For bag-of-visual-words (BOW) encoding, we use soft-assignment of descriptors to visual word clusters [Arandjelović et al., 2016; Philbin et al., 2008] to obtain a differentiable representation. The differentiable BOW representation can be written as:

$$BOW(k) = \sum_{i=1}^{N} a_k(x_i). \tag{3.10}$$

Notice that the exact bag-of-visual-words formulation is reproduced if we replace the soft assignment values by its hard assignment equivalent. This formulation is closely related to the Neural BoF formulation [Passalis and Tefas, 2017], but differs in the way of computing the soft assignment. In detail, [Passalis and Tefas, 2017] performs a softmax operation over the computed L2 distances between the descriptors and the cluster centers, whereas we use soft-assignment given by eq. (3.8) where parameters w are learnable without explicit relation to computing L2 distance to cluster centers. It also relates to [Richard and Gall, 2015] that uses a recurrent neural network to perform the aggregation. The advantage of BOW aggregation over NetVLAD is that it aggregates a list of feature descriptors into a much more compact representation, given a fixed number of clusters. The drawback is that significantly more clusters are needed to obtain a rich representation of the aggregated descriptors.

Inspired by Fisher Vector [Perronnin and Dance, 2007] encoding, we also experimented with modifying the NetVLAD architecture to allow learning of second order feature statistics within the clusters. We will denote this representation as NetFV (for Net Fisher Vectors) as it aims at imitating the standard Fisher Vector encoding [Perronnin and Dance, 2007]. Reusing the previously established soft assignment notation, we can write the NetFV representation as

$$FV1(j,k) = \sum_{i=1}^{N} a_k(x_i)\left(\frac{x_i(j) - c_k(j)}{\sigma_k(j)}\right), \tag{3.11}$$

$$FV2(j,k) = \sum_{i=1}^{N} a_k(x_i)\left(\left(\frac{x_i(j) - c_k(j)}{\sigma_k(j)}\right)^2 - 1\right), \tag{3.12}$$

where $FV1$ is capturing the first-order statistics, $FV2$ is capturing the second-order statistics, $c_k, k \in [1, K]$ are the learnable clusters and $\sigma_k, k \in [1, K]$ are the clusters' diagonal covariances. To define $\sigma_k, k \in [1, K]$ as positive, we first randomly initialize their value with a Gaussian noise with unit mean and small variance and then take the square of the values during training so that they stays positive. In the same manner as NetVLAD, c_k and σ_k are learnt independently from the parameters of the soft-assignment a_k. This formulation differs from [Simonyan et al., 2013; Sydorov et al., 2014] as we are not exactly reproducing the original Fisher Vectors. Indeed the parameters $a_k(x_i)$, c_k and σ_k are decoupled from each other. As opposed to [Simonyan et al., 2013; Sydorov et al., 2014], these parameters are not related to a Gaussian Mixture Model but instead are trained in a discriminative manner.

Finally, we have also investigated a simplification of the original NetVLAD architecture that averages the actual descriptors instead of residuals, as first proposed by [Douze et al., 2013]. We call this variant NetRVLAD (for Residual-less VLAD). This simplification requires less parameters and computing operations (about half in both cases). The NetRVLAD descriptor can be written as

$$RVLAD(j,k) = \sum_{i=1}^{N} a_k(x_i)x_i(j). \tag{3.13}$$

More information about our Tensorflow [Abadi et al., 2015] implementation of these different aggregation models can be found at: `https://github.com/antoine77340/LOUPE`

3.5 Experiments

This section evaluates alternative architectures for video aggregation and presents results for classification on the Youtube-8M [Abu-El-Haija et al., 2016] and text-video retrieval on the MPII Movie Description [Rohrbach et al., 2015] dataset.

3.5.1 Youtube-8M Dataset

The Youtube-8M dataset [Abu-El-Haija et al., 2016] is composed of approximately 8 millions videos. Because the dataset is large scale, visual and audio features are pre-extracted and provided with the dataset. Each video is labeled with one or multiple tags referring to the main topic of the video. Figure 3-7 illustrates examples of videos with their annotations. The original dataset is divided into training, validation and test subsets with 70%, 20% and 10% of videos, respectively. In this chapter, we keep around 20K videos for the validation, the remaining samples from the original training and validation subsets are used for training. This choice was made to obtain a larger training set and to decrease the validation time. We have noticed that the performance on our validation set was comparable (0.2%-0.3% higher) to the test performance evaluated on the Kaggle platform. As we have no access to the test labels, most results in this section are reported for our validation set. We report evaluation using the Global Average Precision (GAP) metric at top 20 as used in the Youtube-8M Kaggle competition (more details about the metric can be found at: `https://www.kaggle.com/c/youtube8m#evaluation`). Note that the performance

Method	GAP
Average pooling + Logistic Regression	71.4%
Average pooling + MoE + CG	74.1%
LSTM (2 Layers)	81.7%
GRU (2 Layers)	82.0%
BoW (4096 Clusters)	81.6%
NetFV (128 Clusters)	82.2%
NetRVLAD (256 Clusters)	82.3%
NetVLAD (256 Clusters)	82.4%
Gated BoW (4096 Clusters)	82.0%
Gated NetFV (128 Clusters)	83.0%
Gated NetRVLAD (256 Clusters)	83.1%
Gated NetVLAD (256 Clusters)	**83.2%**

Table 3.1: Performance comparison for individual aggregation schemes on Youtube 8M. Clustering-based methods are compared with and without Context Gating.

of a random guess is expected to be 0.03% GAP.

3.5.2 MPII Movie Description Dataset

This video dataset [Rohrbach et al., 2015] contains 118,081 short movie extracts from 202 movies. Each clip is annotated with a short text description that either comes from the movie script or a transcribed audio description. The LSMDC challenge [2] (Large Scale Movie Description Challenge) have been organized in 2016 and 2017 to evaluate models on Text-to-Video and Video-to-Text retrieval based on this movie dataset. We use this dataset to evaluate the benefit of our video representation for learning a joint text-video embedding on both of these public benchmarks. More precisely, we have evaluated our model using the LSMDC evaluation protocol [3] for both Text-to-Video and Video-to-Text retrieval tasks.

[2]https://sites.google.com/site/describingmovies/

[3]https://sites.google.com/site/describingmovies/lsmdc-2016/movieretrieval

3.5.3 Implementation details

Video classification on Youtube-8M. In the Youtube 8M competition dataset [Abu-El-Haija et al., 2016] video and audio features are provided for every second of the input video. The visual features consist of ReLU activations of the last fully-connected layer from a publicly available[4] Inception network trained on Imagenet. The audio features are extracted from a CNN architecture trained for audio classification [Hershey et al., 2017]. PCA and whitening are then applied to reduce the dimension to 1024 for the visual features and 128 for the audio features. More details on feature extraction are available in [Abu-El-Haija et al., 2016].

All of our models are trained using the Adam algorithm [Kingma and Ba, 2015] and mini-batches with data from around 100 videos. The learning rate is initially set to 0.0002 and is then decreased exponentially with the factor of 0.8 every 4M samples. We use gradient clipping and batch normalization [Ioffe and Szegedy, 2015] before each non-linear layer.

For the clustering-based pooling models, i.e. BoW, NetVLAD, NetRVLAD and NetFV, we randomly sample N features with replacement from each video. We fix N to 300 for all videos at training and testing, which is equivalent to 5 minutes of video. As opposed to the original version of NetVLAD [Arandjelović et al., 2016], we did not pre-train the codebook with a k-means initialization as we did not notice any improvement by doing so. For training of recurrent models, i.e. LSTM and GRU, we process features in the temporal order. We have also experimented with the random sampling of frames for LSTM and GRU which performs surprisingly similarly. Finally, to enable a fair comparison, all pooled representations have the same size of 1024 dimensions. We denote the pooled representation as the output of the first Context Gating block from the green part of Figure 3-2, so we have D5 = 1024.

Our implementation uses the TensorFlow framework [Abadi et al., 2015]. Each training is performed on a single NVIDIA TITAN X (12Gb) GPU.

Video-Text retrieval on MPII MD. As opposed to the Youtube 8M dataset, we extract visual features from an Imagenet pretrain Resnet-152 layers network [He

[4]`https://www.tensorflow.org/tutorials/image_recognition`

Evaluation task	Text-to-Video				Video-to-Text
Method	R@1	R@5	R@10	MR	Accuracy
Random baseline	0.1	0.5	1.0	500	20.0
C+LSTM+SA+FC7 [Torabi et al., 2016]	4.2	13.0	19.5	90	58.1
SNUVL [Yu et al., 2016b]	3.6	14.7	23.9	50	65.7
CT-SAN [Yu et al., 2017b]	5.1	16.3	25.2	46	67.0
CCA (FV HGLMM) [Klein et al., 2015]	7.5	21.7	31.0	33	72.8
JSFusion [Yu et al., 2017a]	9.1	21.2	34.1	36	73.5
Ours without Context Gating	8.3	21.9	32.3	32	73.0
Ours with Context Gating	**9.8**	**24.6**	**34.3**	**28**	**74.7**

Table 3.2: Text-to-video and Video-to-Text retrieval results from the LSMDC test sets. MR stands for Median Rank (the lower the better).

et al., 2016] so we can compare with state-of-the-art approach [Yu et al., 2017a]. Moreover, as we have access to raw RGB videos, we can also extract motion descriptors based on optical flow using an I3D Kinetics pretrained model [Carreira and Zisserman, 2017]. Audio descriptors are extracted in the same manner as in Youtube 8M. Because the video clips are very short (few seconds) in this dataset, we use a simple max pooling to aggregate features temporally. For the text inputs, we extract for each word, embeddings from a Google News [5] pretrained word2vec word embedding model. Then to aggregate the word features for the sentence representation, we use a NetVLAD module. Eventually our model is trained using the usual max margin ranking loss [Yu et al., 2017a].

3.5.4 Model evaluation for Classification

We evaluate the performance of individual models in Table 3.1. The "Gated" versions for the clustering-based pooling methods include CG layers as described in Section 3.3.1. Using CG layers together with GRU and LSTM has decreased the performance in our experiments. We argue that CG is redundant to GRU and LSTM architectures. In fact, as explained in Section 3.3.3 both LSTM [Hochreiter and Schmidhuber, 1997] and GRU [Cho et al., 2014] implement similar gating mechanism

[5]GoogleNews-vectors-negative300

in their architecture as opposed to NetVLAD, NetFV and Soft-DBoW.

From Table 3.1 we can observe a significant increase of performance provided by all learnt aggregation schemes compared to the Average pooling baselines. Interestingly, the NetVLAD and NetFV representations based on the temporally-shuffled feature pooling outperforms the temporal models (GRU and LSTM). Finally, we can note a consistent increase in performance provided by the Context Gating for all clustering-based pooling methods.

3.5.5 Model evaluation for Text-Video retrieval

We evaluate our model on Text-to-Video and Video-to-Text retrieval in Table 3.2. The goal of the Text-to-Video retrieval task is to retrieve a video from a pool of 1000 videos given an input caption query. The Text-to-Video retrieval performance are measured in R@1, R@5 and R@10 (the higher the better) and in median rank (the lower the better). On the other hand, the goal of the Video-to-Text retrieval task is: given an input video and five different captions, find the only caption that describe the video. The performance is measure in accuracy instead (the higher the better). We compare our approach against the previous state-of-the-art [Yu et al., 2017a] which is also the winner of the LSMDC 2017 challenge in both task. Their approach combines both a bi-directional LSTM and convolutional layers to aggregate the set of descriptors from the different modalities (video, audio and text). Note that we have evaluated our full method with Context Gating (row *Ours with CG*) and our method without any Context Gating layer (row *Ours without CG*). For learning a Text-Video joint embedding, we demonstrate that the use of the Context Gating layer is also beneficial as it has allowed us to outperform the previous state-of-the-art approach on all metrics and tasks.

3.5.6 Context Gating ablation study

Table 3.3 reports an ablation study evaluating the effect of Context Gating on the NetVLAD aggregation with 128 clusters. The addition of CG layers in the feature

After pooling	After MoE	GAP
-	-	82.2%
Gated Linear Unit	-	82.4%
Context Gating	-	82.7%
Gated Linear Unit	Context Gating	82.7%
Context Gating	Context Gating	**83.0%**

Table 3.3: Context Gating ablation study on Youtube 8M. There is no GLU layer after MoE as GLU does not output probabilities.

Method	Early Concat	Late Concat
NetVLAD	81.9%	**82.4%**
NetFV	81.2%	**82.2%**
GRU	**82.2%**	82.1%
LSTM	**81.7%**	81.1%

Table 3.4: Evaluation of audio-video fusion methods (Early and Late Concat) on Youtube 8M.

pooling and classification modules gives a significant increase in GAP. We have observed a similar behavior for NetVLAD with 256 clusters. We also experimented with replacing the Context Gating by the GLU [Dauphin et al., 2016] after pooling. To make the comparison fair, we added a Context Gating layer just after the MoE. Despite being less complex than GLU, we observe that CG also performs better. We note that the improvement of 0.8% provided by CG is similar to the improvement of the best non-gated model (NetVLAD) over LSTM in Table 3.1.

Fig 3-7 illustrates top 5 recognition outputs on some videos from our validation set of a NetVLAD based model with and without Context Gating (CG). While some of the top recognized labels from the model without CG effectively appear in the videos such as *Nature* (video 2), *Animal* (video 3), *Lawn* (video 4), *Coin* (video 5) or *Finger* (video 6), these predicted labels are however not what the video is essentially about. On the other hand, we notice that the model with CG does not tend to do these errors as it seems to better model the context of the videos.

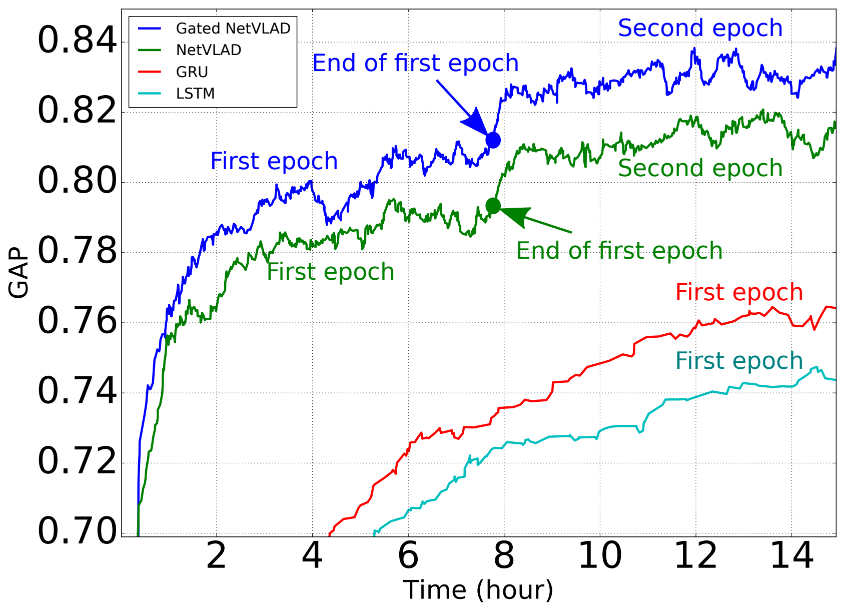


Figure 3-5: Training GAP performance within the very first training epoch of different models on Youtube 8M.

3.5.7 Video-Audio fusion

In addition to the late fusion of audio and video streams (Late Concat) described in Section 3.3, we have also experimented with a simple concatenation of original audio and video features into a single vector, followed by the pooling and classification modules in a "single stream manner" (Early Concat). Results in Table 3.4 illustrate the effect of the two fusion schemes for different pooling methods. The two-stream audio-visual architecture with the late fusion improves performance for the clustering-based pooling methods (NetVLAD and NetFV). On the other hand, the early fusion scheme seems to work better for GRU and LSTM aggregations.

3.5.8 Generalization

One valuable feature of the Youtube-8M dataset is the large-scale annotated data (almost 10 millions videos). More common annotated video datasets usually have sizes several orders of magnitude lower, ranging from 10k to 100k samples. With the large-scale dataset at hand we evaluate the influence of the amount of training

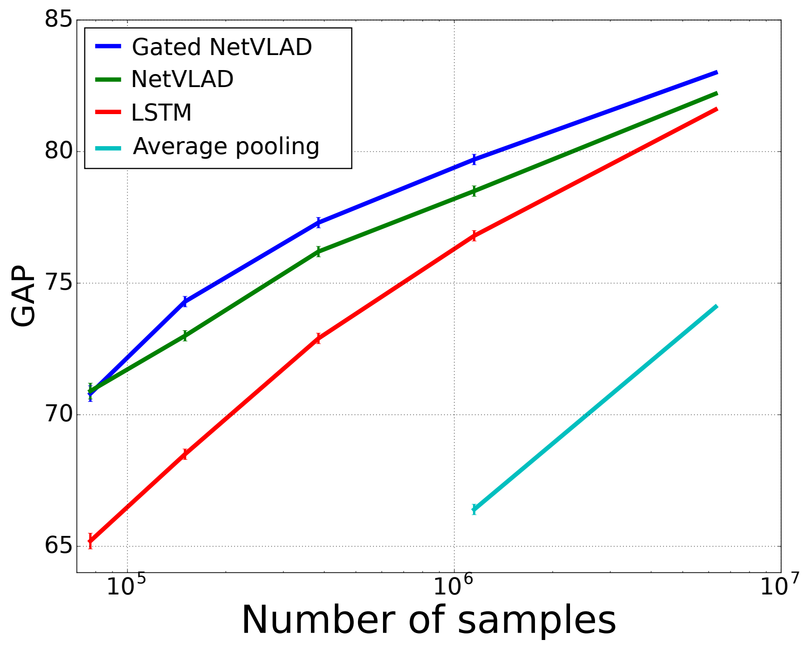


Figure 3-6: The validation GAP performance of the different main models when varying the training dataset size on Youtube 8M.

data on the performance of different models. To this end, we experimented with training different models: Gated NetVLAD, NetVLAD, LSTM and average pooling based model on multiple randomly sampled subsets of the Youtube 8M dataset. We have experimented with subsets of 70K, 150K, 380K and 1150K samples.

For each subset size, we have trained models using three non-overlapping training subsets and measured the variance in performance. Figure 3-6 illustrates the GAP performance of each model when varying the training size. The error bars represent the variance observed when training the models on the three different training subsets. We have observed low and consistent GAP variance for different models and training sizes. Despite the LSTM model has less parameters (around 40M) compared to NetVLAD (around 160M) and Gated NetVLAD (around 180M), NetVLAD and Gated NetVLAD models demonstrate better generalization than LSTM when trained from a lower number of samples. The Context Gating module still helps generalizing better the basic NetVLAD based architecture when having sufficient number of samples (at least 100k samples). We did not show results with smaller dataset sizes as the

Approach	Ensemble size	GAP
Ours (Full)	25	85.0
Ours (Light)	7	84.7
[Wang et al., 2017]	75	84.6
[Li et al., 2017]	57	84.5
[Chen et al., 2017b]	134	84.2
[Skalic et al., 2017]	75	84.2

Table 3.5: Ensemble model sizes of the top ranked teams (out of 655) from the Youtube 8M kaggle competition.

results for all models were drastically dropping down. This is mainly due to the fact that the task is a multi-label prediction problem with a large pool of roughly 5000 labels. As these labels have a long-tail distribution, decreasing the dataset size to less than 30k samples would leave many labels with no single training example. Thus, it would not be clear if the drop of performance is due to the aggregation technique or a lack of training samples for rare classes.

3.5.9 Ensembling

We explore the complementarity of different models and consider their combination through ensembling on the Youtube 8M dataset. Ensembling consists in averaging label prediction scores of different models. We have observed the increased effect of ensembling when combining diverse models. To select models from a set of 50 trained models with different hyper-parameters and aggregation functions, we follow a greedy approach. We start with the best performing model and choose the next one by maximizing the GAP of the ensemble on the validation set. Our final ensemble used in the Youtube 8M challenge contains 25 models. Note that a seven models ensemble is enough to reach the first place with a GAP on the private test set of 84.688. Our code to reproduce this smaller ensemble is available at: `https://github.com/antoine77340/Youtube-8M-WILLOW`. Table 3.5 shows the ensemble size of the other top ranked approaches, out of 655 teams, from the Youtube-8M kaggle challenge. Besides showing the best performance at the competition, we also designed a smaller set of models that ensemble more efficiently than others. Indeed, we need much less models in our ensemble than the other top performing approaches. Full ranking can be found at: `https://www.kaggle.com/c/youtube8m/leaderboard`.

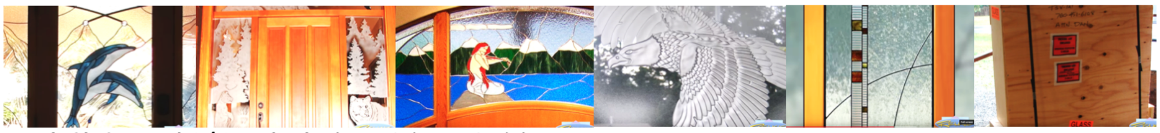
Training cost. We wish to emphasize that our single model is much faster to train than previous state-of-the-art RNN based approach. Figure 3-5 shows the evolution of the training GAP within the first epoch (thus can also be considered as the validation GAP) when training several models on a single GPU. We can observe that 1 hour of Gated NetVLAD training performs better than LSTM and GRU trained after more than 10 hours. This is due to two main factors: first, as explained in 3.5.8, to perform similarly, our model is required much less training data than RNNs. Second, even though our proposed architecture contains much more parameters than RNNs (See Table 3.1), our method fully exploits GPU hardware as most of the computation can be distributed as opposed to RNNs. As an example, we have empirically observed that a forward-backward model computation is 3 to 5 times faster than with RNN based approach. Our proposed method can be trained on a single GPU from 1 to 2 days as opposed to 5 to 7 days when using RNN based approach. This has enabled us to significantly reduce the training cost when ensembling different models.

3.6 Conclusion

In this chapter, we have addressed the problem of aggregating temporal video descriptors for video understanding by exploring trainable variants of classical pooling methods such as BoW, VLAD and FV. We have applied our video representation on the tasks of video classification / text-video retrieval and show benefits of our method compared to state-of-the-art. In this context we have observed NetVLAD, NetFV and BoW to outperform more common temporal models such as LSTM and GRU. We have also introduced the Context Gating mechanism and have shown its benefit for the trainable versions of BoW, VLAD and FV. The ensemble of our individual models has been shown to improve the performance further, enabling our method to win the Youtube 8M Large-Scale Video Understanding challenge.

Youtube id: aCbqzn3svbM / Groundtruth: Wood - Stairs - Furniture
With CG: Wood (99.5%) - Furniture (90.2%) - Stairs (82.6%) - Home improvement (37.0%) - Floor (20.6%)
Without CG: Wood (99.0%) - Furniture (94.0%) - Home improvement (67.3%) - Floor (41.8%) - Cabinetry (21.8%)

Youtube id: aDntpMvYF5s / Groundtruth: Outdoor recreation - Pokémon
With CG: Outdoor recreation (97.4%) - Pokémon (94.1%) - Pokémon (video game series) (65.4%) - Nature (28.7%) - Forest (11.4%)
Without CG: Outdoor recreation (99.8%) - Nature (81.6%) - Tree (26.2%) - Forest (21.5%) - Animation (20.5%)



Youtube id: aiVIRDFmdZQ / Groundtruth: Glass - Window - Stained glass - Art
With CG: Glass (74.0%) - Window (51.0%) - Art (41.6%) - Stained glass (31.8%) - Wall (31.6%)
Without CG: Animal (24.2%) - Wall (21.4%) - Window (16.8%) - Art (14.4%) - Home improvement (8.4%)

Youtube id: akkpW1WUCLE / Groundtruth: Radio-control aircraft - Vehicle - Car - Quadcopter - Unmanned aerial vehicle - Model aircraft - Radio-control model
With CG: Vehicle (99.4%) - Radio-control model (93.0%) - Model aircraft (88.1%) - Radio-control aircraft (88.0%) - Unmanned aerial vehicle: (87.8%)
Without CG: Vehicle (98.6%) - Lawn (86.2%) - Racing (58.7%) - Model aircraft (42.6%) - Mower (37.7%)

Youtube id: a8NEsTt7OpI / Groundtruth: Bible
With CG: : Bible (50.7%) - Book (13.9%) - Injury (7.5%) - Writing (5.5%) - Art (5.3%)
Without CG: Coin (18.7%) - House (15.8%) - Art (14.2%) - Book (13.8%) - Wall (8.1%)

Youtube id: apb41aPjRXU / Groundtruth: Jewellery - Beadwork - Earring
With CG: Earring (80.0%) - Jewellery (54.0%) - Beadwork (38.8%) - Bead: (34.5%) - Clay (15.6%)
Without CG: Jewellery (38.3%) - Finger (32.6%) - Beadwork (26.1%) - Nail (anatomy) (24.1%) - Nail art (16.1%)

Youtube id: aiBOrjDvoMg / Groundtruth: Wood - Vehicle - Canoe - Boat
With CG: Boat (79.5%) - Vehicle (69.1%) - Wood (63.1%) - Canoe: (59.5%) - Kayak (25.3%)
Without CG: Wood (30.1%) - Vehicle (9.5%) - Food (7.6%) - Biology (6.0%) - Canoe (5.7%)

Figure 3-7: This figure illustrates recognition outputs on our validation set. For each video, we show the groundtruth labels as well as the top 5 predictions from a NetVLAD based model with and without Context Gating. Green outputs are correctly identified ones.

Chapter 4

Learning from Video and Text via Large-Scale Discriminative Clustering

In the previous chapter, we have introduced a novel approach for representing videos. In particular, our proposed approach demonstrated state-of-the-art results on the large-scale and weakly-labeled YouTube-8M dataset [Abu-El-Haija et al., 2016]. In this chapter, we will instead study the problem of large-scale weakly-supervised learning from readily available movie scripts.

In particular, we will consider a discriminative clustering based approach, which has been successfully applied to a number of weakly-supervised learning tasks. Such applications include person and action recognition, text-to-video alignment, object co-segmentation and co-localization in videos and images. One drawback of discriminative clustering, however, is its limited scalability. In this chapter, we address this issue and propose an online optimization algorithm based on the Block-Coordinate Frank-Wolfe algorithm [Lacoste-Julien et al., 2013]. We apply the proposed method to the problem of weakly-supervised learning of actions and actors from movies together with corresponding movie scripts. The scaling up of the learning problem to 66 feature-length movies enables us to significantly improve weakly-supervised action recognition.

4.1 Introduction

Action recognition has been significantly improved in recent years. Most existing methods [Laptev et al., 2008; Simonyan and Zisserman, 2014; Wang and Schmid, 2013] rely on supervised learning and, therefore, require large-scale, diverse and representative action datasets for training. Collecting such datasets, however, is a difficult task given the high costs of manual search and annotation of the video. Notably, the largest action datasets today are still orders of magnitude smaller (UCF101 Soomro et al. [2012], ActivityNet [Caba Heilbron et al., 2015]) compared to large image datasets, they often contain label noise and target specific domains such as sports (Sports1M [Karpathy et al., 2014b]).

Weakly supervised learning aims to bypass the need of manually-annotated datasets using readily-available, but possibly noisy and incomplete supervision. Examples of such methods include learning of person names from image captions or video scripts [Berg et al., 2004; Everingham et al., 2006; Sivic et al., 2009; Tapaswi et al., 2012]. Learning actions from movies and movie scripts has been approached in [Bojanowski et al., 2013, 2014; Duchenne et al., 2009; Laptev et al., 2008]. Most of the work on weakly supervised person and action learning, however, has been limited to one or a few movies. Therefore the power of leveraging large-scale weakly annotated video data has not been fully explored.

In this chapter, we aim to scale weakly supervised learning of actions. In particular, we follow the work of [Bojanowski et al., 2013] and learn actor names together with their actions from movies and movie scripts. While actors are learned separately for each movie, differently from [Bojanowski et al., 2013], our method simultaneously learns actions from all movies and movie scripts available for training. Such an approach, however, requires solving a large-scale optimization problem. We address this issue and propose to scale weakly supervised learning by adapting the Block-Coordinate Frank-Wolfe approach [Lacoste-Julien et al., 2013]. Our optimization procedure enables action learning from tens of movies and thousands of action samples, readily available from our subset of movies or other recent datasets with movie

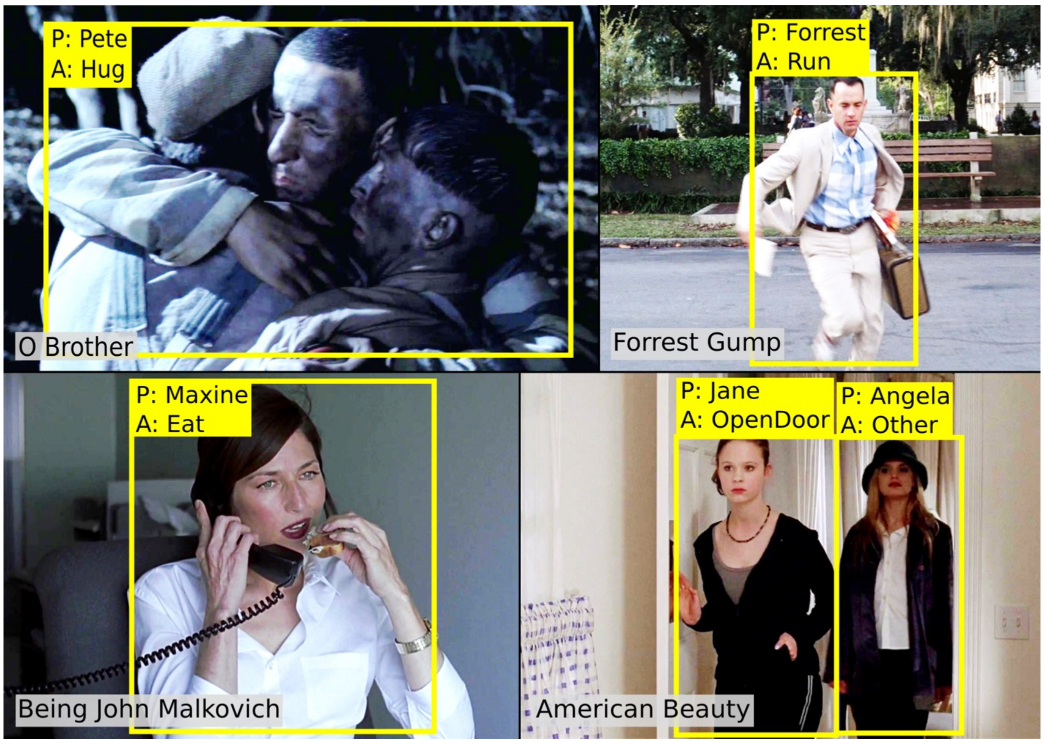


Figure 4-1: We automatically recognize actors and their actions in a of dataset of 66 movies with scripts as weak supervision.

descriptions [Rohrbach et al., 2015]. This, in turn, results in large improvements in action recognition.

Besides the optimization, the chapter introduces a new model for background class in the form of a constraint. It enables better and automatic modeling of the background class (*i.e.* unknown actors and actions). We evaluate our method on 66 movies and demonstrate significant improvements for both actor and action recognition. Example results are illustrated in Figure 4-1.

4.1.1 Related Work

This section reviews related work on discriminative clustering, Frank-Wolfe optimization and its applications to the weakly supervised learning of people and actions in video.

Discriminative clustering and Frank-Wolfe. The Frank-Wolfe algorithm [Frank and Wolfe, 1956; Jaggi, 2013] allows to minimize large convex problems over convex sets by solving a sequence of linear problems. In computer vision, it has been used in combination with discriminative clustering [Bach and Harchaoui, 2007] for action localization [Bojanowski et al., 2014], text-to-video alignment [Alayrac et al., 2016; Bojanowski et al., 2015], object co-localization in videos and images [Joulin et al., 2014b] or instance-level segmentation [Seguin et al., 2016]. A variant of Frank-Wolfe with randomized block coordinate descent was proposed in [Lacoste-Julien et al., 2013]. This extension leads to lower complexity in terms of time and memory requirements while preserving the convergence rate. In this chapter, we build on [Lacoste-Julien et al., 2013] and adapt it for the problem of large-scale weakly supervised learning of actions from movies.

Weakly supervised action recognition. Movie scripts are used as a source of weak supervision for temporal action localization in [Duchenne et al., 2009]. An extension of this work [Bojanowski et al., 2014] exploits the temporal order of actions as a learning constraint. Other [Lan et al., 2011] target spatio-temporal action localization and recognition in video using a latent SVM. A weakly supervised extension of this method [Shapovalova et al., 2012] localizes actions without location supervision at the training time. Another recent work [Weinzaepfel et al., 2016] proposes a multi-fold Multiple-Instance Learning (MIL) SVM to localize actions given video-level supervision at training time. Closer to us is the work of [Bojanowski et al., 2013] that improves weakly supervised action recognition by joint action-actor constraints derived from scripts. While the approach in [Bojanowski et al., 2013] is limited to a few action classes and movies, we propose here a scalable solution and demonstrate significant improvements in action recognition when applied to the large-scale weakly supervised learning of actions from many movies.

Weakly supervised person recognition. Person recognition in TV series has been studied in [Everingham et al., 2006; Sivic et al., 2009] where the authors propose a solution to the problem of associating speaker names in scripts and faces in videos. Speakers in videos are identified by detecting face tracks with lip motion. The method in [Cour et al., 2009] presents an alternative solution by formulating the association problem using a convex surrogate loss. Parkhi *et al.* [Parkhi et al., 2015a] present a method for person recognition combining a MIL SVM with a model for the background class. Most similar to our model is the one presented in [Bojanowski et al., 2013]. The authors propose a discriminative clustering cost under linear constraints derived from scripts to recover the identities and actions of people in movies. Apart from scaling-up the approach of [Bojanowski et al., 2013] to much larger datasets, our model extends and improves [Bojanowski et al., 2013] with a new background constraint.

Contributions. In this chapter, we make the following contributions: (i) We propose an optimization algorithm based on Block-Coordinate Frank-Wolfe that allows scaling up discriminative clustering models [Bach and Harchaoui, 2007] to much larger datasets. (ii) We extend the joint weakly-supervised Person-Action model of [Bojanowski et al., 2013], with a simple yet efficient model of the background class. (iii) We apply the proposed optimization algorithm to scale-up discriminative clustering to an order of magnitude larger dataset, resulting in significantly improved action recognition performance.

4.2 Discriminative Clustering for Weak Supervision

We want to assign labels (*e.g.* names or action classes) to samples (*e.g.* person tracks in the video). As opposed to the standard supervised learning setup, the exact labels of samples are not known at training time. Instead, we are given only partial information that some samples in a subset (or bag) may belong to some of the labels. This ambiguous setup, also known as multiple instance learning, is common, for example, when learning human actions from videos and associated text descriptions.

To address this challenge of ambiguous and partial labels, we build on the discriminative clustering criterion based on a linear classifier and a quadratic loss (DIFFRAC [Bach and Harchaoui, 2007]). This framework has shown promising results in weakly supervised and unsupervised computer vision tasks [Alayrac et al., 2016; Bojanowski et al., 2013, 2014, 2015; Joulin et al., 2010, 2012; Ramanathan et al., 2014; Seguin et al., 2016]. In particular, we use this approach to group samples into linearly separable clusters. Suppose we have N samples to group into K classes. We are given d-dimensional features $X \in \mathbb{R}^{N \times d}$, one for each of the N samples, and our goal is to find a binary matrix $Y \in \{0,1\}^{N \times K}$ assigning each of the N samples to one of the labels, where $Y_{nk} = 1$ if and only if the sample n (*e.g.* a person track in a movie) is assigned to the label k (*e.g.* action class running).

First, suppose the assignment matrix Y is given. In this case finding a linear classifier W can be formulated as a ridge regression problem

$$\min_{W \in \mathbb{R}^{d \times K}} \frac{1}{2N} \|Y - XW\|_{\mathrm{F}}^2 + \frac{\lambda}{2} \|W\|_{\mathrm{F}}^2, \tag{4.1}$$

where X is a matrix of input features, Y is the given labels assignment matrix, $\|.\|_{\mathrm{F}}$ is the matrix norm (or Frobenius norm) induced by the matrix inner product $\langle ., . \rangle_{\mathrm{F}}$ (or Frobenius inner product) and λ is a regularization hyper-parameter set to a fixed constant. The key observation is that we can resolve the classifier W^* in closed form as

$$W^*(Y) = (X^\top X + N\lambda I)^{-1} X^\top Y. \tag{4.2}$$

In our weakly supervised setting, however, Y is unknown. Therefore, we treat Y as a latent variable and optimize (4.1) w.r.t. W and Y. In details, plugging the optimal solution W^* (4.2) in the cost (4.1) removes the dependency on W and the cost can be written as a quadratic function of Y, i.e. $C(Y) = \langle Y, A(X, \lambda) Y \rangle_{\mathrm{F}}$, where $A(X, \lambda)$ is a matrix that only depends on the data X and a regularization parameter λ. Finding the best assignment matrix Y can then be written as the minimization of the cost $C(Y)$:

$$\min_{Y \in \{0,1\}^{N \times K}} \langle Y, A(X, \lambda) Y \rangle_{\mathrm{F}}. \tag{4.3}$$

Solving the above problem, however, can lead to degenerate solutions [Bach and Harchaoui, 2007] unless additional constraints on Y are provided. In section 4.3, we incorporate weak supervision in the form of constraints on the latent assignment matrices Y. The constraints on Y used for weak supervision generally decompose into small independent blocks. This block structure is the key for our optimization approach that we will present next.

4.2.1 Large-Scale optimization

The Frank-Wolfe (FW) algorithm has been shown to be effective for optimizing convex relaxation of (4.3) in a number of vision problems [Alayrac et al., 2016; Bojanowski et al., 2013, 2014, 2015; Joulin et al., 2014a; Seguin et al., 2016]. It only requires solving linear programs on a set of constraints. Therefore, it avoids costly projections and allows the use of complicated constraints such as temporal ordering [Bojanowski et al., 2014]. However, the standard FW algorithm is not well suited to solve (4.3) for a large number of samples N.

First, storing the $N \times N$ matrix $A(X, \lambda)$ in memory becomes prohibitive (*e.g.* the size of A becomes $\geq$ 100GB for $N \geq 200000$). Second, each update of the FW algorithm requires a full pass over the data resulting in a space and time complexity of order N for each FW step.

Weakly supervised learning is, however, largely motivated by the desire of using large-scale data with "cheap" and readily-available but incomplete and noisy annotation. Scaling up weakly supervised learning to a large number of samples is, therefore, essential for its success. We address this issue and develop an efficient version of the FW algorithm. Our solution builds on the Block-Coordinate Frank-Wolfe (BCFW) [Lacoste-Julien et al., 2013] algorithm and extends it with a smart block-dependent update procedure as described next. The proposed update procedure is one of the key contribution of this chapter.

Block-coordinate Frank-Wolfe (BCFW)

The Block-Coordinate version of the Frank-Wolfe algorithm [Lacoste-Julien et al., 2013] is useful when the convex feasible set $\mathcal{Y}$ can be written as a Cartesian product of n smaller sets of constraints: $\mathcal{Y} = \mathcal{Y}^{(1)} \times \ldots \times \mathcal{Y}^{(n)}$. Inspired by coordinate descent techniques, BCFW consists of updating one variable block $\mathcal{Y}^{(i)}$ at a time with a reduced Frank-Wolfe step. This method has potentially n times cheaper iterates both in space and time. We will see that most of the weakly supervised problems exhibit such a block structure on latent variables.

BCFW for discriminative clustering

To benefit from BCFW, we have to ensure that the time and space complexity of the block update does not depend on the total number of samples N (*e.g.* person tracks in all movies) but only depends on the size N_i of smaller blocks of samples i (*e.g.* person tracks within one movie). After a block is sampled, the update consists of two steps. First, the gradient with respect to the block is computed. Then the *linear oracle* is called to obtain the next iteration. As we show below, the difficult part in our case is to efficiently compute the gradient with respect to the block.

Block gradient: a naive approach. Let's denote N_i the size of block i. The objective function f of problem (4.3) is $f(Y) = \langle Y, A(X, \lambda)Y \rangle_{\mathrm{F}}$, where (see [Bach and Harchaoui, 2007])

$$A(X, \lambda) = \frac{1}{2N}(I_N - X(X^\top X + N\lambda I_d)^{-1}X^\top). \tag{4.4}$$

To avoid storing matrix $A(X, \lambda)$ of size $N \times N$, one can precompute the matrix $P = (X^\top X + N\lambda I_d)^{-1}X^\top \in \mathbb{R}^{d \times N}$. We can write the block gradient with respect to a subset of samples i as follows:

$$\nabla_{(i)} f(Y) = \frac{1}{N}(Y^{(i)} - X^{(i)}PY), \tag{4.5}$$

where $Y^{(i)} \in \mathbb{R}^{N_i \times K}$ and $X^{(i)} \in \mathbb{R}^{N_i \times d}$ are the label assignment variable and the feature matrix for block i (*e.g.* person tracks in movie i), respectively. Because of the PY matrix multiplication, naively computing this formula has the complexity $\mathcal{O}(NdK)$, where N is the total number of samples, d is the dimensionality of the feature space and K is the number of classes. As this depends linearly on N, we aim to find a more efficient way to compute block gradients, as described next.

Block gradient: a smart update. We now propose an update procedure that avoids re-computation of block gradients and whose time and space complexity at each iteration depends on N_i instead of N. The main intuition is that we need to find a way to store information about all the blocks in a compact form. A natural way of doing so is to maintain the weights of the linear regression parameters $W \in \mathbb{R}^{d \times K}$. From (4.2) we have $W = PY$. If we are able to maintain the variable W at each iteration with the desired complexity $\mathcal{O}(N_i dK)$, then the block gradient computation (4.5) can be reduced from $\mathcal{O}(NdK)$ to $\mathcal{O}(N_i dK)$. We now explain how to effectively achieve that.

At each iteration t of the algorithm, we only update a block i of Y while keeping all other blocks fixed. We denote the direction of the update by $D_t \in \mathbb{R}^{N \times K}$ and the step size by γ_t. With this notation the update becomes

$$Y_{t+1} = Y_t + \gamma_t D_t. \tag{4.6}$$

The update rule for the weight variable W_t can now be written as follows:

$$\begin{aligned} W_{t+1} &= P(Y_t + \gamma_t D_t) \\ W_{t+1} &= W_t + \gamma_t P D_t, \end{aligned} \tag{4.7}$$

Recall that at iteration t, BCFW only updates block i, therefore D_t has non zero

Algorithm 1 BCFW for Discriminative Clustering [Bach and Harchaoui, 2007]

```
Initiate $Y_0$, $P := (X^\top X + N\lambda I_d)^{-1}X^\top$, $W_0 = PY_0$, $g_i = +\infty,\ \forall i$.
for $t = 1 \dots N_{iter}$ do
    $i \leftarrow$ sample from distribution proportional to $g$ [Osokin et al., 2016]
    $\nabla_{(i)} f(Y_t) \leftarrow \frac{1}{N}(Y_t^{(i)} - X^{(i)}W_t)$ # Block gradient
    $Y_{min} \leftarrow \text{argmin}_{x \in \mathcal{Y}^{(i)}} \langle \nabla_{(i)} f(Y_t), x \rangle_{\text{F}}$ # Linear oracle
    $D \leftarrow Y_{min} - Y^{(i)}$
    $g_i \leftarrow -\langle D, \nabla_{(i)} f(Y_t) \rangle_{\text{F}}$ # Block gap
    $\gamma \leftarrow \min(1, \frac{g_i}{\frac{1}{N}\langle D, D - X^{(i)}P^{(i)}D \rangle_{\text{F}}})$ # Line-search
    $W_{t+1} \leftarrow W_t + \gamma P^{(i)} D$ # W update
    $Y_{t+1}^{(i)} \leftarrow Y_t^{(i)} + \gamma D$ # Block update
end for
```

value only at block i. In block notation we can therefore write the matrix product PD_t as:

$$\left[P^{(1)}, \cdots, P^{(i)}, \cdots, P^{(n)}\right] \times \begin{bmatrix} 0 \\ D_t^{(i)} \\ 0 \end{bmatrix} = P^{(i)} D_t^{(i)}, \tag{4.8}$$

where $P^{(i)} \in \mathbb{R}^{d \times N_i}$ and $D_t^{(i)} \in \mathbb{R}^{N_i \times K}$ are the i-th blocks of matrices P and D_t, respectively. The outcome is an update of the following form

$$W_{t+1} = W_t + \gamma_t P^{(i)} D_t^{(i)}, \tag{4.9}$$

where the computational complexity for updating W has been reduced to $\mathcal{O}(N_i dK)$ compared to $\mathcal{O}(NdK)$ in the standard update.

We have designed a Block-Coordinate Frank-Wolfe update with time and space complexity depending only on the size of the blocks and not the entire dataset. This allows to scale discriminative clustering to problems with a very large number of samples. The pseudo-code for the algorithm is summarized in Algorithm 1. Next, we describe an application of this large-scale discriminative clustering algorithm to weakly supervised person and action recognition in movies.

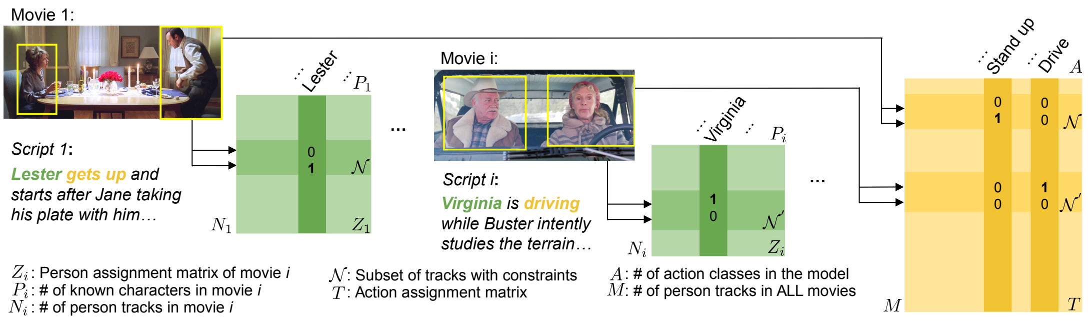

Movie 1:
Script 1:
Lester gets up and starts after Jane taking his plate with him…
Lester
P_1
N_1
Z_1
$\mathcal{N}$
Movie i:
Script i:
Virginia is driving while Buster intently studies the terrain…
Virginia
P_i
N_i
Z_i
$\mathcal{N}'$
Stand up
Drive
A
M
T
Z_i: Person assignment matrix of movie i
P_i: # of known characters in movie i
N_i: # of person tracks in movie i
$\mathcal{N}$: Subset of tracks with constraints
T: Action assignment matrix
A: # of action classes in the model
M: # of person tracks in ALL movies

Figure 4-2: Overview of the Person-Action weakly supervised model, see text for detailed explanations.

4.3 Weakly supervised Person-Action model

We now describe an application of our large-scale discriminative clustering algorithm with weak-supervision. The goal is to assign to each person track a name and an action. Both names and actions are mined from movie scripts. For a given movie i, we assume to have N_i automatically extracted person tracks as well as the parsing of a movie script into person names and action classes. We also assume that scripts and movies have been roughly aligned in time. In such a setup we can assign labels (*e.g.* a name or an action) from a script section to a subset of tracks $\mathcal{N}$ from the corresponding time interval of a movie (see Figure 4-2 for example). In the following, we explain how to convert such form of weak supervision into a set of constraints on latent variables corresponding to the names and actions of people. We will also show how these constraints easily decompose into blocks. We denote Z the latent variable assignment matrix for person names and T for actions.

4.3.1 Weak-supervision as constraints

We use linear constraints to incorporate weak supervision from movie scripts. In detail, we define constraints on subsets of person tracks that we call “bags". In the following we explain the procedure for construction of bags together with the definition of the appropriate constraints.

'At least one' constraint. Suppose a script reveals the presence of a person p in some time interval of the movie. We construct a set $\mathcal{N}$ with all person tracks in this interval. As first proposed by [Bojanowski et al., 2013], we model that *at least one* track in $\mathcal{N}$ is assigned to person p by the following constraint

$$\sum_{n \in \mathcal{N}} Z_{np} \geq 1. \tag{4.10}$$

We can apply the same type of constraint when solving for action assignment T.

Person-Action constraint. Scripts can also provide information that a person p is performing an action a in a scene. In such cases we can formulate stricter and more informative constraints as follows. We construct a set $\mathcal{N}$ containing all person tracks appearing in this scene. Following [Bojanowski et al., 2013], we formulate a joint constraint on presence of a person performing a specific action as

$$\sum_{n \in \mathcal{N}} Z_{np} T_{na} \geq 1. \tag{4.11}$$

Mutual exclusion constraint. We also model that each person track can only be assigned to exactly one label. This restriction can be formalized by the mutual exclusion constraint

$$Z 1_P = 1_N, \tag{4.12}$$

for Z (*i.e.* rows sum up to 1). Same constraint holds for T.

Background class constraint. One of our contributions is a novel way of coping with the background class. As opposed to previous work [Bojanowski et al., 2013], our approach allows us to have background model that does not require any external data. Also it does not require a specific background class classifier as in [Parkhi et al., 2015a].

Our background class constraint can be seen as a way to supervise people and actions that are not mentioned in scripts. We observe that tracks that are not subject to constraints from Eq. (4.10) and tracks that belong to crowded shots are likely to belong to the background class. Let us denote by $\mathcal{B}$ the set of such tracks. We impose that at least a certain fraction $\alpha \in [0,1]$ of tracks in $\mathcal{B}$ must belong to the background class. Assuming that person label $p = 1$ corresponds to the background, we obtain the following linear constraint (similar constraint can be defined for actions on T):

$$\sum_{n \in \mathcal{B}} Z_{n1} \geq \alpha \mid \mathcal{B} \mid . \tag{4.13}$$

4.3.2 Person-Action model formulation

Here we summarize the complete formulation of the person and action recognition problems.

Solving for names. We formulate the person recognition problem as discriminative clustering, where X_1 are face descriptors:

$$\begin{aligned} \min_{Z \in \{0,1\}^{N \times P}} \quad & \langle Z, A(X_1, \lambda) Z \rangle_{\mathrm{F}}, \quad \text{(Discriminative cost)} \\ \text{such that} \quad & \begin{cases} \sum_{n \in \mathcal{N}} Z_{np} \geq 1, & \text{(At least one)} \\ \sum_{n \in \mathcal{B}} Z_{n1} \geq \alpha \mid \mathcal{B} \mid, & \text{(Background)} \\ Z 1_P = 1_N. & \text{(Mutual exclusion)} \end{cases} \end{aligned} \tag{4.14}$$

Solving for actions. After solving the previous problem for names separately for each movie, we vertically concatenate all person name assignment matrices Z. We also define a single action assignment variable T in $\{0,1\}^{M \times A}$, where M is the total number of tracks across all movies and X_2 are action descriptors (details given later).

We formulate our action recognition problem as a large QP:

$$\min_{T\in\{0,1\}^{M\times A}} \quad \langle T, A(X_2,\mu)T\rangle_{\mathrm{F}}, \quad \text{(Discriminative cost)} \tag{4.15}$$

$$\text{such that} \quad \begin{cases} \sum_{n\in\mathcal{N}} T_{na} \geq 1, & \text{(At least one)} \\ \sum_{n\in\mathcal{N}} Z_{np}T_{na} \geq 1, & \text{(Person-Action)} \\ \sum_{n\in\mathcal{B}} T_{n1} \geq \beta \mid \mathcal{B} \mid, & \text{(Background)} \\ T1_A = 1_M. & \text{(Mutual exclusion)} \end{cases}$$

Block-Separable constraints. The set of linear constraints on the action assignment matrix T is block separable since each movie has it own set of constraints, i.e. there are no constraints spanning multiple movies. Therefore, we can fully demonstrate here the power of our large-scale discriminative clustering optimization (Algorithm 1).

4.3.3 Slack variables

To account for imprecise information in movie scripts, we add slack variables to our constraints. We penalyze the values of slack variables with the L_2 penalty. The slack-augmented constraints are defined as:

$$\sum_{n\in\mathcal{N}} Z_{np} \geq 1-\xi, \tag{4.16}$$

$$\sum_{n\in\mathcal{N}} T_{na} \geq 1-\xi, \tag{4.17}$$

$$\sum_{n\in\mathcal{N}} Z_{np}\, T_{na} \geq 1-\xi, \tag{4.18}$$

where ξ is the slack variable.

4.3.4 Lower bound

In practice, we noticed that modifying the value of the lower bound in constraints (4.16), (4.17), (4.18) from 1 to a higher value can significantly improve the performance of the algorithm. The constraints we use become:

$$\sum_{n \in \mathcal{N}} Z_{np} \geq \alpha_1 - \xi, \tag{4.19}$$

$$\sum_{n \in \mathcal{N}} T_{na} \geq \alpha_2 - \xi, \tag{4.20}$$

$$\sum_{n \in \mathcal{N}} Z_{np}\, T_{na} \geq \alpha_2 - \xi, \tag{4.21}$$

where $\alpha_1, \alpha_2 \in \mathbb{R}_+$ are hyper-parameters.

4.4 Experimental Setup

4.4.1 Dataset

Our dataset is composed of 66 Hollywood feature-length movies[1] that we obtained from either BluRay or DVD. For all movies, we downloaded their scripts (on `www.dailyscript.com`) that we temporally aligned with the videos and movie subtitles using the method described in [Laptev et al., 2008]. The total number of frames in all 66 movies is 11,320,252. The number of body tracks detected across all movies (see 4.4.3 for more details) is $M = 201874$.

[1]American Beauty, As Good As It Gets, Being John Malkovich, Big Fish, Bringing Out the Dead, Bruce the Almighty, Casablanca, Charade, Chasing Amy, Clerks, Crash, Dead Poets Society, Double Indemnity, Erin Brockovich, Fantastic Four, Fargo, Fear and Loathing in Las Vegas, Fight Club, Five Easy Pieces, Forrest Gump, Gandhi, Gang Related, Get Shorty, Hudsucker Proxy, I Am Sam, Independence Day, Indiana Jones and the Last Crusade, It Happened One Night, Jackie Brown, Jay and Silent Bob Strike Back, LA Confidential, Legally Blonde, Light Sleeper, Little Miss Sunshine, Living in Oblivion, Lone Star, Lost Highway, Men In Black, Midnight Run, Misery, Mission to Mars, Moonstruck, Mumford, Ninotchka, O Brother, Pirates of the Caribbean Dead Mans Chest, Psycho, Pulp Fiction, Quills, Raising Arizona, Rear Window, Reservoir Dogs, The Big Lebowski, The Butterfly Effect, The Cider House Rules, The Crying Game, The Godfather, The Graduate, The Grapes of Wrath, The Hustler, The Lord of the Rings The Fellowship of the Ring, The Lost Weekend, The Night of the Hunter, The Pianist, The Princess Bride, Truman Capote

ACTION	# movies	Other	St.U.	E.	S.D.	Si.U.	H.S.	F.	G.C.	K.	H.	A.	R.	O.D.	D.	Total
Ground truth	5	14532	146	24	112	19	28	90	26	47	74	28	277	131	59	15593
Constraints	66	$\emptyset$	237	85	146	46	49	70	81	244	44	99	156	208	169	1634

Table 4.1: Action recognition ground truth and constraint statistics. (St.U: `Stand Up`, E.: `Eat`, S.D: `Sit Down`, Si.U.: `Sit Up`, H.S: `Hand Shake`, F.: `Fight`, G.C.: `Get out of Car`, K.: `Kiss`, H.: `Hug`, A.: `Answer Phone`, R.: `Run`, O.D.: `Open Door`, D.: `Drive`)

4.4.2 Text pre-processing

To provide weak supervision for our method we process movie scripts to extract occurrences of the 13 most frequent action classes: `Stand Up`, `Eat`, `Sit Down`, `Sit Up`, `Hand Shake`, `Fight`, `Get Out of Car`, `Kiss`, `Hug`, `Answer Phone`, `Run`, `Open Door` and `Drive`. To do so, we collect a corpus of movie scripts different from the set of our 66 movies and train simple text-based action classifiers using linear SVM and a TF-IDF representation of words composed of uni-grams and bi-grams. After retrieving actions in our target movie scripts, we also need to identify who is performing the action. We used spaCy [Honnibal and Johnson, 2015] to parse every sentence classified as describing one of the 13 actions and get every subject for each action verb.

Table 4.1 provides the number of action constraints we extracted from the 66 movie scripts. It also shows the number of ground truth intervals for each action we obtained by an exhaustive manual annotation of human actions in five testing movies.

4.4.3 Person detection and Features

Face tracks. To obtain tracks of faces in the video, we run the multi-view face detector [Mathias et al., 2014] based on the DPM model [Girshick et al.]. We then extract face tracks using the same method as in [Everingham et al., 2006; Sivic et al., 2009]. For each detected face, we compute facial landmarks [Sivic et al., 2009] followed by the face alignment and resizing of face images to 224x224 pixel. We use pre-trained vgg-face features [Parkhi et al., 2015b] to extract descriptors for each face. We kept the features of dimension 4096 computed by the network at the last fully-connected layer that we L_2 normalized. For each face track, we choose the top K (in practice, we choose K=5) faces that have the best facial landmark confidence. Then we represent each track by averaging the features of the top K faces.

Body tracks. To get the person body tracks, we run the Faster-RCNN network with VGG-16 architecture fine-tuned on VOC 07 [Shaoqing et al., 2015]. Then we track bounding boxes using the same tracker as used to obtain face tracks. To get person identity for body tracks, we greedily link each body track to one face track by maximizing a spatio-temporal bounding box overlap measure. However if a body track does not have an associated face track as the actor's face may look away from the camera, we cannot obtain its identity. Such tracks can be originating from any actor in the movie. To capture motion features of each body track, we compute bag-of-visual-words representation of dense trajectory descriptors [Wang and Schmid, 2013] inside the bounding boxes defined by the body track. We use 4000 cluster centers for each of the HOF, MBHx and MBHy channels. In order to capture appearance of each body track we extract ResNet-50 [He et al., 2016] pre-trained on ImageNet. For each body bounding box, we compute the average RoI-pooled [Shaoqing et al., 2015] feature map of the last convolutional layer within the bounding box, which yields a feature vector of dimension 2048 for each box. We extract a feature vector every 10th frame, average extracted feature vectors over the duration of the track and L_2 normalize. Finally, we concatenate the dense trajectory descriptor and the appearance descriptor resulting in a 14028-dimensional descriptor for each body track.

4.4.4 Combining face and body tracks

We describe here, how we linked body tracks to face tracks. Let's denote $a_1, a_2, ..., a_n$, n faces tracks in the current shot and $b_1, b_2, ..., b_m$ the m body tracks in this same shot (we assume $m \geq n$). We want to model that each face track is associated to at most one body track but a body track does not necessary have a face track, as the face of a person may not always be visible. Let's also define the following overlap measure O between a face track a and a body track b. If $\mathcal{A}$ is a set of all frames of

Method	Acc.	Multi-Class AP	BG AP
[Cour et al., 2009]	48	63	-
[Sivic et al., 2009]	49	63	-
[Bojanowski et al., 2013]	57	75	51
[Parkhi et al., 2015a]	74	93	75
Our method	**83**	**94**	**82**

Table 4.2: Comparison on the Casablanca benchmark [Bojanowski et al., 2013]. BG stands for Background.

Episode	1	2	3	4	5
[Sivic et al., 2009]	90	83	70	86	85
[Parkhi et al., 2015a]	**99**	90	94	96	**97**
Ours	98	**98**	**98**	**97**	**97**

Table 4.3: Comparison on the Buffy benchmark [Sivic et al., 2009] using AP.

the track a and $a(t)$, $b(t)$ are bounding boxes of tracks a and b at frame t, we have:

$$O(a,b) = \sum_{t \in \mathcal{A}} \frac{Area(a(t) \cap b(t))}{Area(a(t))}. \tag{4.22}$$

We compute the overlap for all possible pairs $O(a_i, b_j)$, where $i \in [1, n]$ and $j \in [1, m]$. Then we associate each face track a_i with the body track b_j that maximizes $O(a_i, b_j)$. Finally, for each body track b_j we either do not have any associated face track (then the body track won't have a match) or have multiple face tracks a_i associated to it. In the latter case, we match the body track b_j with the face track a_i that maximizes $O(a_i, b_j)$.

4.5 Evaluation

4.5.1 Evaluation of person recognition

We compare our person recognition method to several other methods on the Casablanca benchmark from [Bojanowski et al., 2013] and the Buffy benchmark from [Sivic et al., 2009]. All methods are evaluated on the same inputs: same face tracks, scripts and

α	0	0.1	0.2	0.3	0.4	0.5	0.75	1.0
Accuracy	58	58	70	82	**84**	83	76	55
AP	86	87	90	**94**	**94**	93	85	58

Table 4.4: Sensitivity to hyper-parameter α (4.13) on Casablanca.

characters. Table 4.2 shows the Accuracy (Acc.) and Average Precision (AP) of our approach compared to other methods on the Casablanca benchmark [Bojanowski et al., 2013]. In particular we compare to Parkhi *et al.* [Parkhi et al., 2015a] which is a strong baseline using the same CNN face descriptors as in our method. We also show the AP of classifying the background character class (Background AP). We compare in Table 4.3 our approach to other methods [Parkhi et al., 2015a; Sivic et al., 2009] reporting results on season 5 of the TV series “Buffy the Vampire Slayer". Both of these methods [Parkhi et al., 2015a; Sivic et al., 2009] use speaker detection to mine additional strong (but possibly incorrect) labels from the script, which we also incorporate (as additional bags) to make the comparison fair. Our method demonstrates significant improvement over the previous results. It also outperforms other methods on the task of classifying background characters. Finally, Table 4.4 shows the sensitivity to hyper-parameter α from the background constraint (4.13) on the Casablanca benchmark. Note that in contrast to other methods, our background model does not require supervision for the background class. This clearly demonstrates the advantage of our proposed background model. For all experiments the hyper-parameter α of the background constraint (4.13) was set to 30%. Figure 4-5 illustrates our qualitative results for character recognition in different movies.

4.5.2 Evaluation of action recognition

First, we compare our method to Bojanowski *et al.* 2013 [Bojanowski et al., 2013]. Their evaluation uses different body tracks than ours, we design here an algorithm-independent evaluation setup. We compare our model using the Casablanca movie and the `Sit Down` action. For the purpose of evaluation, we have manually annotated all person tracks in the movie and then manually labeled whether or not they contain

Method	# movies	J-M	St.U.	E.	S.D.	Si.U.	H.S.	F.	G.C.	K.	H.	A.	R.	O.D.	D.	mAP
(a) Random	$\emptyset$	No	0.9	0.1	0.7	0.1	0.1	0.6	0.2	0.3	0.5	0.2	1.8	0.8	0.4	0.5
(b) Script only	$\emptyset$	No	3.0	4.3	5.5	2.8	4.7	2.5	1.6	11.3	4.2	1.4	13.7	3.1	3.0	4.7
(c) Fully-supervised	4	No	21.2	0.2	22.2	0.9	0.6	7.3	1.4	1.9	**4.5**	2.0	33.2	18.5	6.3	9.3
(d) Few training movies	5	Yes	22.6	9.6	15.6	**8.1**	9.7	6.1	1.0	6.0	2.1	4.2	44.0	16.2	15.9	12.4
(e) No Joint Model	66	No	10.7	7.0	17.1	7.3	**18.0**	**12.6**	**2.0**	**14.9**	3.6	**5.8**	24.4	14.2	24.9	12.5
(f) Full setup	66	Yes	**27.0**	**9.8**	**28.2**	6.7	7.8	5.9	1.0	12.9	1.7	5.7	**56.3**	**21.3**	**29.7**	**16.4**

Table 4.5: Average Precision of actions evaluated on 5 movies. (St.U: `Stand Up`, E.: `Eat`, S.D: `Sit Down`, Si.U.: `Sit Up`, H.S: `Hand Shake`, F.: `Fight`, G.C.: `Get out of Car`, K.: `Kiss`, H.: `Hug`, A.: `Answer Phone`, R.: `Run`, O.D.: `Open Door`, D.: `Drive`). J-M stands for Joint-Model.

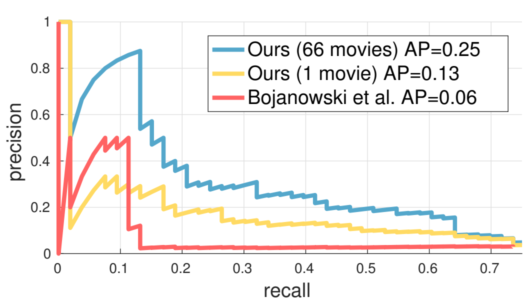


Figure 4-3: PR curves of action Sit Down from Casablanca.

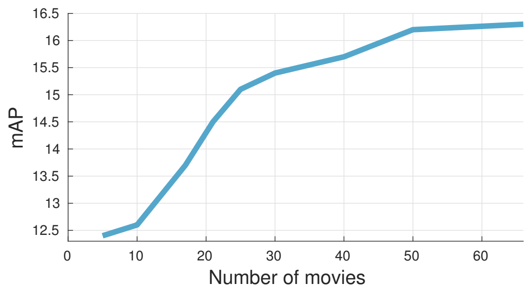


Figure 4-4: Action recognition mAP with increasing number of training movies.

the `Sit Down` action. Given this ground truth, we assess the two models in a similar way as typically done in object detection. Figure 4-3 shows a precision-recall curve evaluating recognition of the `Sit Down` action. We show our method trained on Casablanca only (as done in [Bojanowski et al., 2013]) and then on all 66 movies. Our method trained on Casablanca is already better than [Bojanowski et al., 2013]. The improvement becomes even more evident when training our method on all 66 movies.

To evaluate our method on all 13 action classes, we use five movies (American Beauty, Casablanca, Double Indemnity, Forrest Gump and Fight Club). For each of these movies we have manually annotated all person tracks produced by our tracker according to 13 target action classes and the background action class. We assume that each track corresponds to at most one target action. In rare cases where this assumption is violated, we annotate the track by one of the correct action classes.

In Table 4.5 we compare results of our model to different baselines. The first

β	0	0.1	0.2	0.3	0.4	0.5	0.6	0.75	0.8
mAP	15.0	15.7	15.9	15.8	16.6	16.1	16.2	16.0	15.5

Table 4.6: Influence of the hyper-parameter β (4.13) for action recognition.

baseline (**a**) corresponds to the random assignment of action classes. The second baseline (**b**) Script only uses information extracted from the scripts: each time an action appears in a bag, all person tracks in this bag are then simply annotated with this action. Baseline (**c**) is using our action descriptors but trained in a fully supervised set-up on a small subset of annotated movies. The fourth baseline (**d**) is our method train only *using the five evaluated movies.* The fifth baseline (**e**) is our model without the joint person-action constraint (4.11), but still trained on all 66 movies. Finally, the last result (**f**) is from our model using all the 66 training movies and person-action constraints (4.11). Results demonstrate that optimizing our model on more movies brings the most significant improvement to the final results. We confirm the idea from [Bojanowski et al., 2013] that adding the information of who is performing the action in general helps identifying actions. However we also notice it is not always true for actions with interacting people such as: `Fight`, `Hand Shake`, `Hug` or `Kiss`. Knowing who is doing the action does not seems to help for these actions. Figure 4-4 shows improvements in action recognition when gradually increasing the number of training movies. Figure 4-6 shows qualitative results of our model on different movies. Table 4.6 shows the low sensitivity of the action recognition results to the β (4.15) hyper-parameter on the action recognition results.

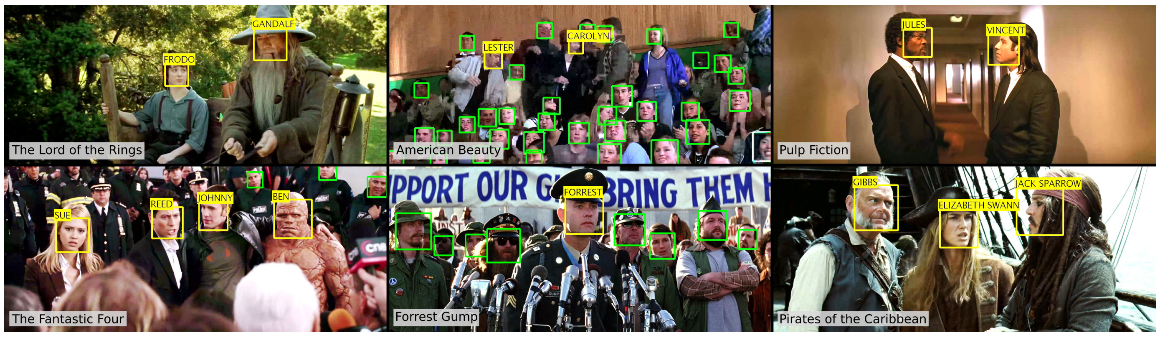


Figure 4-5: Qualitative results for face recognition. Green bounding boxes are face tracks correctly classified as background characters.

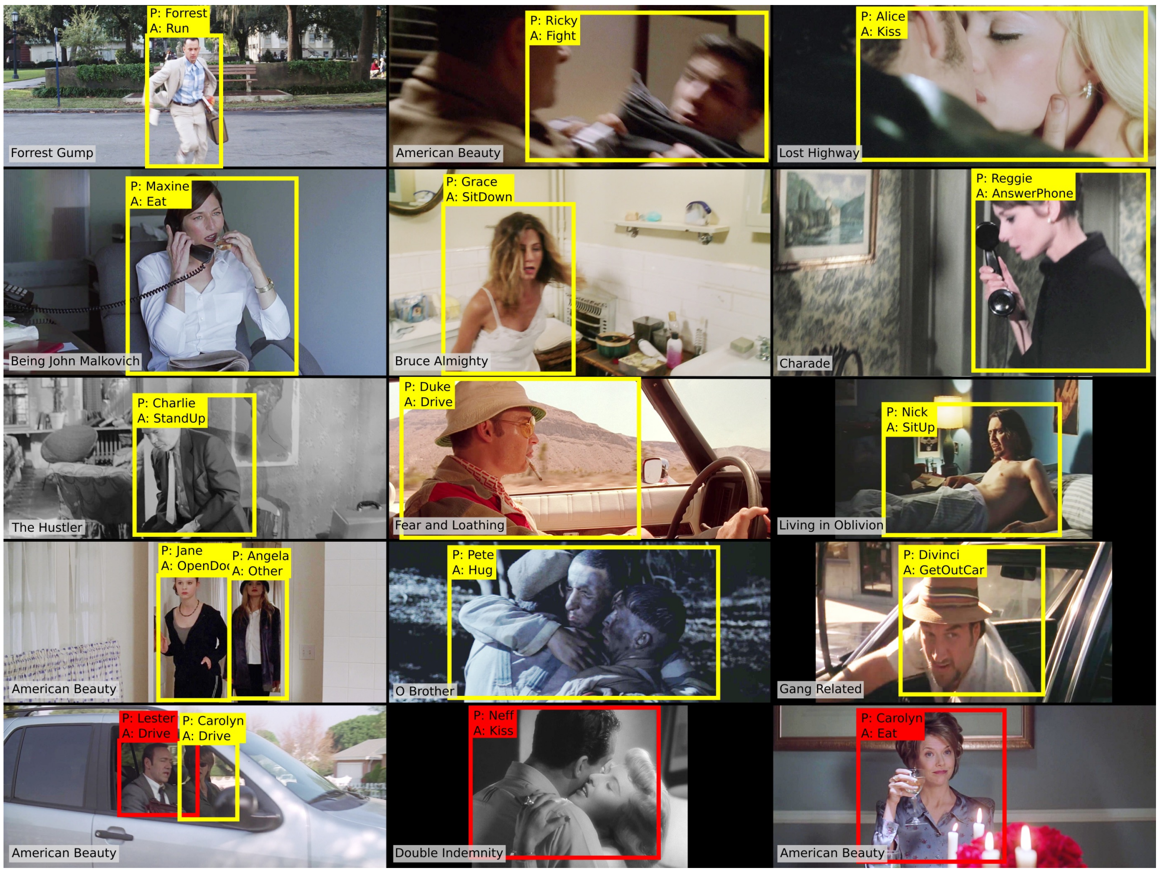


Figure 4-6: Qualitative results for action recognition. P stands for for the name of the character and A for the action performed by P. Last row (in red) shows mislabeled tracks with high confidence (*e.g.* hugging labeled as kissing, sitting in a car labeled as driving).

4.6 Conclusion

In this chapter, we have proposed an efficient online optimization method based on the Block-Coordinate Frank-Wolfe algorithm. We use this new algorithm to scale-up discriminative clustering model in the context of weakly-supervised person and action recognition in feature-length movies. Moreover, we have proposed a novel way of handling the background class, which does not require collecting background class data as required by the previous approaches, and leads to better performance for person recognition. In summary, the proposed model significantly improves action recognition results on 66 feature-length movies. The significance of the technical contribution goes beyond the problem of person-action recognition as the proposed optimization algorithm can scale-up other problems recently tackled by discriminative clustering. Examples include: unsupervised learning from narrated instruction videos [Alayrac et al., 2016], text-to-video alignment [Bojanowski et al., 2015], co-segmentation [Joulin et al., 2010], co-localization in videos and images [Joulin et al., 2014b] or instance-level segmentation [Seguin et al., 2016], which can be now scaled-up to an order of magnitude larger datasets.

Chapter 5

HowTo100M: Learning a Text-Video Embedding from Uncurated Narrated Videos

In the previous chapters, we have introduced an approach for representing videos (Chapter 3) as well as an approach for weakly-supervised learning from readily available supervision in the form of movie scripts (Chapter Chapter 4). In this chapter, we propose to learn a joint text-video embedding from video with readily available natural language annotations in the form of automatically transcribed narrations. We will also reuse the Context Gating module previously introduced in Chapter 3. The contributions of this chapter are three-fold. First, we introduce *HowTo100M*: a large-scale dataset of 136 million video clips sourced from 1.22M narrated instructional web videos depicting humans performing and describing over 23k different visual tasks. Our data collection procedure is fast, scalable and does not require any additional manual annotation. Second, we demonstrate that a text-video embedding trained on this data leads to state-of-the-art results for text-to-video retrieval and action localization in instructional video datasets such as YouCook2 or CrossTask. Finally, we show that this embedding transfers well to other domains: fine-tuning on generic YouTube videos (MSR-VTT dataset) and movies (LSMDC dataset) outperforms models trained on these datasets alone.

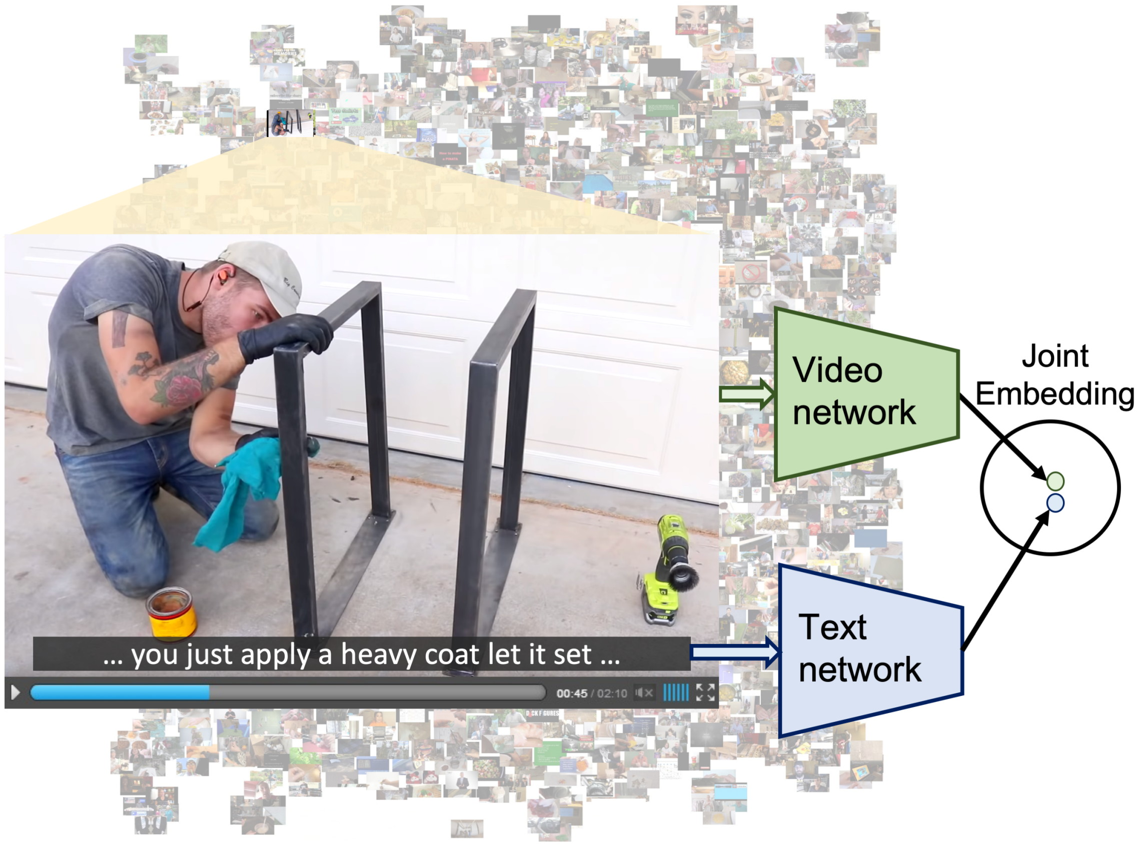


Figure 5-1: We learn a joint text-video embedding by watching millions of narrated video clips of people performing diverse visual tasks. The learned embedding transfers well to other instructional and non-instructional text-video datasets.

5.1 Introduction

Communicating about the visual world using language is a key ability of humans as intelligent beings. A three year old child can manipulate objects, observe its own actions and describe them to others using language; while adults can learn new skills by reading books or watching videos. This interplay between video and language extends naturally to artificial agents that need to understand the visual world and communicate about it with people. Examples of tasks that still represent a significant challenge for current artificial systems include text-to-video retrieval [Klein et al., 2015; Miech et al., 2018; Wang et al., 2018a, 2016b; Yu et al., 2018], text-based action or event localization [Hendricks et al., 2017], video captioning [Pan et al., 2016a; Yu et al., 2016a], and video question answering [Tapaswi et al., 2016; Yu et al., 2018].

Yet, progress on these problems is important for a host of applications from searching video archives to human-robot communication.

A common approach to model visual concepts described with language is to learn a mapping of text and video into a shared embedding space, where related text fragments and video clips are close to each other [Hendricks et al., 2017; Miech et al., 2018; Pan et al., 2016b; Plummer et al., 2017; Xu et al., 2015a]. Learning a good representation often requires a large set of paired video clips and text captions. In fact, given the huge variability of video scenes and their textual descriptions, learning a generic embedding space may require millions of paired video clips and text captions. However, existing datasets (*e.g.* MSR-VTT [Xu et al., 2016], DiDeMo [Hendricks et al., 2017], EPIC-KITCHENS [Damen et al., 2018]), are on the scale of tens to hundreds of thousands of such pairs that have been annotated manually. Manual collection of such datasets is expensive and hard to scale. It is also subjective since video annotation can often be an ill-defined task with low annotator consistency [Xu et al., 2016].

In this chapter, we explore a different source of supervision to obtain paired video clips and text captions for learning joint representations of video and language. We observe that *narrated instructional videos* are available in large quantities (*e.g.* on YouTube) and provide a large amount of visual and language data. In particular, instructional videos [Alayrac et al., 2016; Malmaud et al., 2015; Zhukov et al., 2019] often contain narration with an explicit intention of explaining the visual content on screen. To leverage this rich source of data, we collect a new large-scale dataset containing 136 million video clips sourced from 1.22 million narrated instructional videos depicting humans performing more than 23,000 different tasks. Each clip is paired with a text annotation in the form of an automatically transcribed narration.

Contributions. The contributions of this chapter are three-fold. First, we collect a new dataset of close-captioned video clips, *HowTo100M*, that is orders of magnitude larger than any other existing video-text datasets (Section 5.3). Second, we show that such data can be used to learn powerful video-language representations. Our model

(Section 5.4), trained on HowTo100M, sets a new state-of-the-art for text-based action localization and text-to-video retrieval on existing datasets of instructional videos, YouCook2 [Zhou et al., 2018b] and CrossTask [Zhukov et al., 2019]. Finally, we explore the ability of models trained on our data to transfer to non-instructional videos. In particular, we demonstrate that models pretrained on HowTo100M can be successfully transferred by fine tuning on the MSR-VTT dataset (generic Youtube videos) and the LSMDC dataset (movies).

5.2 Related work

A significant number of computer vision applications rely on a joint understanding of visual and textual cues. These applications include automatic image and video captioning [Johnson et al., 2016; Pan et al., 2016a; You et al., 2016; Yu et al., 2016a], visual question answering [Fukui et al., 2016; Malinowski et al., 2015; Tapaswi et al., 2016; Yu et al., 2018], visual content retrieval based on textual queries [Miech et al., 2018; Wang et al., 2018c; Yu et al., 2018], temporal localization of events in videos using natural language [Hendricks et al., 2017; Krishna et al., 2017] or video summarization with natural language [Plummer et al., 2017].

Vision, language and speech. A common approach to model vision and language is learning a joint embedding space where visual and textual cues are adjacent if and only if they are semantically similar [Chowdhury et al., 2018; Dong et al., 2019; Gong et al., 2014a,b; Klein et al., 2015; Miech et al., 2018; Mithun et al., 2018; Pan et al., 2016b; Plummer et al., 2017; Xu et al., 2015a; Wang et al., 2018a, 2016b; Wu et al., 2017]. Most of these works rely on medium scale well annotated datasets in which descriptive captions are collected for each video clip. This process is costly as it requires considerable human annotation effort making these datasets hard to scale (see Table 5.1). In this chapter, we train a joint video and language model *without a single manually annotated video description* by leveraging automatically transcribed narrated videos. Using the spoken text from narrated videos to supervise vision models has seen some recent interest [Alayrac et al., 2016; Chen et al., 2017a; Harwath

Dataset	Clips	Captions	Videos	Duration	Source	Year
Charades [Sigurdsson et al., 2016b]	10k	16k	10,000	82h	Home	2016
MSR-VTT [Xu et al., 2016]	10k	200k	7,180	40h	Youtube	2016
YouCook2 [Zhou et al., 2018b]	14k	14k	2,000	176h	Youtube	2018
EPIC-KITCHENS [Damen et al., 2018]	40k	40k	432	55h	Home	2018
DiDeMo [Hendricks et al., 2017]	27k	41k	10,464	87h	Flickr	2017
M-VAD [Torabi et al., 2015]	49k	56k	92	84h	Movies	2015
MPII-MD [Rohrbach et al., 2015]	69k	68k	94	41h	Movies	2015
ANet Captions [Krishna et al., 2017]	100k	100k	20,000	849h	Youtube	2017
TGIF [Li et al., 2016]	102k	126k	102,068	103h	Tumblr	2016
LSMDC [Rohrbach et al., 2017]	128k	128k	200	150h	Movies	2017
How2 [Sanabria et al., 2018]	185k	185k	13,168	298h	Youtube	2018
HowTo100M	**136M**	**136M**	**1.221M**	**134,472h**	Youtube	2019

Table 5.1: Comparison of existing video description datasets. The size of our new HowTo100M dataset bypasses the size of largest available datasets by three orders of magnitude. M denotes million while k denotes thousand.

et al., 2018; Malmaud et al., 2015; Sanabria et al., 2018; Yu et al., 2014]. [Harwath et al., 2018] utilize the raw speech waveform to supervise the visual model, however, their method does not scale as annotators were paid to record audio descriptions for thousands of images. [Chen et al., 2017a] use subtitles from documentaries to automatically obtain object labels, but their focus is on learning object detectors rather than text-video embeddings and their dataset contains only 9 documentary movies, compared to about 15 years of video content considered in this work.

Learning from instructional videos. Instructional videos are rising in popularity in the context of learning steps of complex tasks [Alayrac et al., 2016; Huang et al., 2016; Richard et al., 2017, 2018; Sener and Yao, 2018; Zhukov et al., 2019], visual-linguistic reference resolution [Huang et al., 2017, 2018], action segmentation in long untrimmed videos [Zhou et al., 2018a] and joint learning of object states and actions [Alayrac et al., 2017]. Related to this chapter, [Alayrac et al., 2016; Malmaud et al., 2015; Yu et al., 2014] also consider automatically generated transcription of narrated instructional videos as a source of supervision. However as opposed to this chapter, these works typically extract from transcriptions only a small number of predefined labels.

Numerous datasets of web instructional videos were proposed over the past years

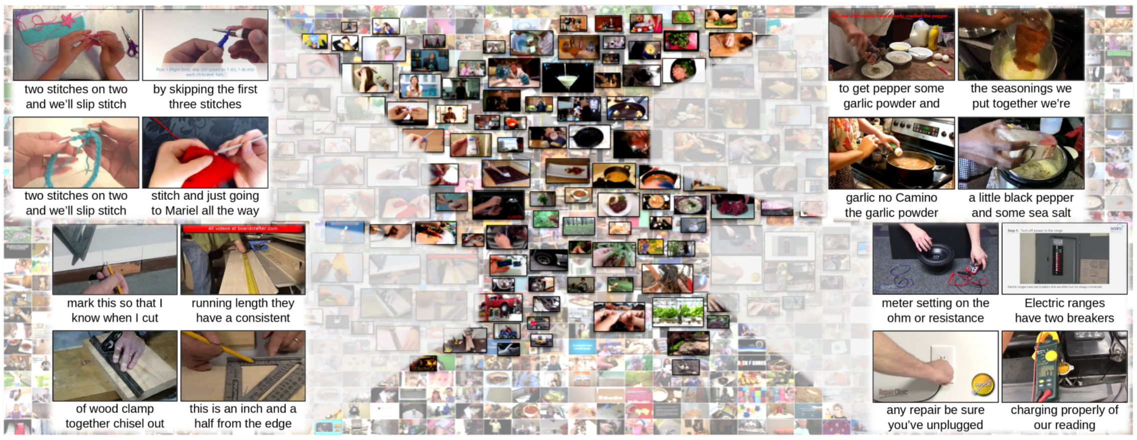

two stitches on two and we'll slip stitch
by skipping the first three stitches
two stitches on two and we'll slip stitch
stitch and just going to Mariel all the way
mark this so that I know when I cut
running length they have a consistent
of wood clamp together chisel out
this is an inch and a half from the edge
to get pepper some garlic powder and
the seasonings we put together we're
garlic no Camino the garlic powder
a little black pepper and some sea salt
meter setting on the ohm or resistance
Electric ranges have two breakers
any repair be sure you've unplugged
charging properly of our reading

Figure 5-2: Examples of clip-caption pairs retrieved with the help of our joint embedding. Pairs are selected based on the similarity between visual appearance and corresponding narration, while they are arranged based on linguistic similarity across pairs. Examples are taken from 4 distinct clusters, corresponding to *Knitting*, *Woodwork/Measuring*, *Cooking/Seasoning* and *Electric maintenance*.

[Alayrac et al., 2016; Malmaud et al., 2015; Sanabria et al., 2018; Sener et al., 2015; Tang et al., 2019; Zhou et al., 2018b; Zhukov et al., 2019]. Among the first to harvest instructional videos, [Sener et al., 2015] use WikiHow, an encyclopedia of *how to* articles, to collect 17 popular physical tasks, and obtain videos by querying these tasks on YouTube. In a similar vein, COIN [Tang et al., 2019] and CrossTask [Zhukov et al., 2019] datasets are collected by first searching for tasks on WikiHow and then videos for each task on YouTube. We use the same approach for collecting HowTo100M. The main distinction between our dataset and previous efforts is the unprecedented scale both in terms of variety (more than 23,000 tasks from 12 different domains) and size (136 million clips sourced from 1.2 million instructional videos).

Large-scale data for model pretraining. The use of large-scale and potentially noisy data from the web is an exciting prospect to pretrain language and vision models. In natural language processing, BERT [Devlin et al., 2018], GPT [Radford et al., 2018], and GPT-2 [Radford et al., 2019] are examples of language models trained on large-scale data that achieve state-of-the-art for many tasks. In fact, training GPT-2 on WebText [Radford et al., 2019] a dataset of 40GB of text from Reddit achieves state-of-the-art even in *zero-shot* settings. In vision, [Mahajan et al., 2018a; Sun et al., 2017] explore the use of image metadata such as Instagram hashtags to pretrain image classifiers.

We are inspired by these works and focus our efforts on learning a strong embedding for joint understanding of video and language. We demonstrate that our video-language embedding learned from millions of YouTube videos not only outperforms previous work on tasks related to instructional videos without fine-tuning, but also generalizes well to non-instructional videos with some fine-tuning. We release our dataset, feature extraction pipeline, and model parameters as a resource that the video and language community can build on.

5.3 The HowTo100M dataset

We collect a new dataset of narrated videos with an emphasis on instructional videos where content creators teach complex tasks. This ensures that most narrations describe the observed visual content. HowTo100M features 1.22 million videos from YouTube, with activities from domains such as cooking, hand crafting, personal care or gardening. Each video is associated with a narration available as subtitles that are either written manually or are the output of an Automatic Speech Recognition (ASR) system.

5.3.1 Data collection

Visual tasks. With an aim to obtain instructional videos that describe how to perform certain activities, we first start by acquiring a large list of activities using *WikiHow*[1] – an online resource that contains 120,000 articles on *How to ...* for a variety of domains ranging from cooking to human relationships structured in a hierarchy. We are primarily interested in "visual tasks" that involve some interaction with the physical world (*e.g. Making peanut butter*, *Pruning a tree*) as compared to others that are more abstract (*e.g. Ending a toxic relationship*, *Choosing a gift*). To obtain predominantly visual tasks, we limit them to one of 12 categories (listed in

[1]`https://www.wikihow.com`

Table 5.2). We exclude categories such as *Relationships* and *Finance and Business*, that may be more abstract.

We further refine the set of tasks, by filtering them in a semi-automatic way. In particular, we restrict the primary verb to physical actions, such as *make*, *build* and *change*, and discard non-physical verbs, such as *be*, *accept* and *feel.* This procedure yields 23,611 visual tasks in total.

Instructional videos. We search for YouTube videos related to the task by forming a query with *how to* preceding the task name (*e.g.* *how to paint furniture*). We choose videos that have English subtitles - either uploaded manually, generated automatically by YouTube ASR, or generated automatically after translation from a different language by YouTube API.

We improve the quality and consistency of the dataset, by adopting the following criteria. We restrict to the top 200 search results, as the latter ones may not be related to the query task. Videos with less than 100 views are removed as they are often of poor quality or are amateurish. We also ignore videos that have less than 100 words as that may be insufficient text to learn a good video-language embedding. Finally, we remove videos longer than 2,000 seconds.

As some videos may appear in several tasks, we de-duplicate videos based on YouTube IDs. However, note that the dataset may still contain duplicates if a video was uploaded several times or edited and re-uploaded. Nevertheless, this is not a concern at our scale.

5.3.2 Paired video clips and captions

Subtitles are often organized as a list of text chunks (lines), and need not form complete sentences. Each line is associated with a time interval in the video, typically the duration in which the line is uttered. We select each line of the subtitles as a caption, and pair it with the video clip from the time interval corresponding to the line. We show some examples from our clip-caption pairs in Figure 5-2.

Different from other datasets with clip-caption pairs (*e.g.* MSR-VTT), our cap-

Category	Tasks	Videos	Clips
Food and Entertaining	11504	497k	54.4M
Home and Garden	5068	270k	29.5M
Hobbies and Crafts	4273	251k	29.8M
Cars & Other Vehicles	810	68k	7.8M
Pets and Animals	552	31k	3.5M
Holidays and Traditions	411	27k	3.0M
Personal Care and Style	181	16k	1.6M
Sports and Fitness	205	16k	2.0M
Health	172	15k	1.7M
Education and Communications	239	15k	1.6M
Arts and Entertainment	138	10k	1.2M
Computers and Electronics	58	5k	0.6M
Total	23.6k	1.22M	136.6M

Table 5.2: Number of tasks, videos and clips within each category.

tions are *not* manually annotated, but automatically obtained through the narration. Thus, they can be thought of as *weakly paired.* Typical examples of incoherence include the content producer asking viewers to subscribe to their channel, talking about something unrelated to the video, or describing something before or after it happens. Furthermore, our captions are often incomplete, lack punctuation, or are grammatically incorrect sentences, as they come from continuous narration and often ASR. We have manually inspected 400 randomly sampled clip-caption pairs and found that in 51 %, at least one object or action mention in the caption is visually seen in the video clip.

Statistics. The initial set of visual tasks are obtained by focusing on 12 WikiHow categories. Table 5.2 shows the number of collected WikiHow tasks and corresponding videos and clips per category. In Figure 5-8, we show the first two levels of the WikiHow hierarchy: the twelve categories and their subcategories along with the number of chosen tasks and corresponding videos in our dataset. We compare the sizes of existing clip-caption paired datasets in Table 5.1. HowTo100M is several orders of magnitude larger than existing datasets and contains an unprecedented duration (15 years) of video data. However, unlike previous datasets, HowTo100M does not have clean annotated captions. As the videos contain complex activities, they are relatively long with an average duration of 6.5 minutes. On average, a video produces 110 clip-caption pairs, with an average duration of 4 seconds per clip and 4 words (after excluding stop-words) per caption. Figure 5-9 shows frequencies of nouns and verbs in transcribed video narrations. We used the MaxEnt Treebank POS Tagger to obtain the nouns and verbs. Please refer to the Figure 5-9 caption for additional analysis. Our data collection procedure assumes that searching with *How to* queries on YouTube would result in mostly instructional videos. We verify this by randomly selecting 100 videos and labeling their type. 71% of the videos are found to be instructional, 12% are vlogs, and another 7% are product reviews or advertisements. Note that vlogs, reviews and ads may also contain correspondences between visual content and narration. In particular, we noticed that objects shown on screen are often mentioned in narration. We do not discard such non-instructional videos, as they may still be useful for the learning the joint embedding.

5.4 Text-video joint embedding model

We now present our model to learn a joint text-video embedding from the automatically paired video clips and captions in our dataset. More formally, we are given a set of n video clips and associated captions $\{(V_i, C_i)\}_{i=1}^n$. We denote by $\mathbf{v} \in \mathbb{R}^{d_v}$ and $\mathbf{c} \in \mathbb{R}^{d_c}$ the d_v and d_c dimensional feature representation of a video clip V and caption C, respectively. Given this, our goal is to learn two mapping functions: $f : \mathbb{R}^{d_v} \to \mathbb{R}^d$ and $g : \mathbb{R}^{d_c} \to \mathbb{R}^d$ that respectively embed video and caption features into a common d-dimensional space, such that the cosine similarity

$$s(V, C) = \frac{\langle f(\mathbf{v}), g(\mathbf{c}) \rangle}{\|f(\mathbf{v})\|_2 \|g(\mathbf{c})\|_2} \tag{5.1}$$

is high when caption C describes the video clip V, and low otherwise.

In this chapter, we use the class of non-linear embedding functions used in [Miech et al., 2018], which are given by:

$$f(\mathbf{v}) = (W_1^v \mathbf{v} + b_1^v) \circ \sigma(W_2^v (W_1^v \mathbf{v} + b_1^v) + b_2^v) \tag{5.2}$$

$$\text{and } g(\mathbf{c}) = (W_1^c \mathbf{c} + b_1^c) \circ \sigma(W_2^c (W_1^c \mathbf{c} + b_1^c) + b_2^c), \tag{5.3}$$

where $W_1^v \in \mathbb{R}^{d \times d_v}$, $W_1^c \in \mathbb{R}^{d \times d_c}$, $W_2^v, W_2^c \in \mathbb{R}^{d \times d}$, $b_1^v, b_1^c, b_2^v, b_2^c \in \mathbb{R}^d$ are learnable parameters, σ is an element-wise sigmoid activation and $\circ$ is the element-wise multiplication (Hadamard product). In practice, $d_v = 4,096$, $d_c = 4,096$ and $d = 4,096$ resulting in a model composed of 67M parameters. Note that the first term on the right-hand side in Equations (5.2) and (5.3) is a linear fully-connected layer and the second term corresponds to a context gating function [Miech et al., 2017b] with an output ranging between 0 and 1, which role is to modulate the output of the linear layer. As a result, this embedding function can model non-linear multiplicative interactions between the dimensions of the input feature vector which has proven effective in other text-video embedding applications [Miech et al., 2018].

Loss. We train our embedding model using the max-margin ranking loss [Karpathy et al., 2014a; Miech et al., 2018; Wang et al., 2018a, 2016b; Yu et al., 2016b]. At each iteration of our training algorithm, we sample a mini-batch $\mathcal{B} = \{i_1, ..., i_b\} \subset \{1, \ldots, n\}$ of caption-clip training pairs $(V_i, C_i)_{i \in \mathcal{B}}$, and update the model parameters with a gradient step of the following loss:

$$\sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{N}(i)} \max(0, \delta + s_{i,j} - s_{i,i}) + \max(0, \delta + s_{j,i} - s_{i,i}), \tag{5.4}$$

where $s_{i,j} = s(V_i, C_j)$ is the similarity score (5.1) between video clip V_i and caption C_j, $\mathcal{N}(i)$ is a set of negative pairs for caption-clip i and δ is the margin. The first term in Equation (5.4) corresponds to the ranking loss when sampling a negative caption, while the second term corresponds to sampling a negative video clip. We fix $\delta = 0.1$ in practice. Our model parameters are updated using Adam [Kingma and Ba, 2015] with a learning rate of 10^{-4}.

Sampling strategy. Similar to [Hendricks et al., 2017], we apply an intra-video negative sampling strategy to define $\mathcal{N}(i)$. We show in Section 5.5.3 that this approach is *critical* for good performance. More precisely, half of our negative pairs $\{(V_i, C_j) : i \neq j\}$, are selected such that the video clip V_i and the caption C_j belong to the same original YouTube video (as (V_i, C_i)), while the other half are sampled from other YouTube videos. We apply intra-negative sampling to ensure that the learned embedding focuses on relevant aspects of the video clip (*e.g.* the hands of the person showing how to knead dough) rather than irrelevant background features (*e.g.* the kitchen). Later in section 5.5.7, we also provide an empirical analysis of the positive pair sampling strategy. We show that even though the training data is noisy, our attempts to automatically select correct positive pairs during training did not yield improvements so far. We think this could be attributed to the fact our model is shallow and is trained on a large amount of data.

Clip and caption representation. The clip feature $\mathbf{v}$ consists of temporally max-pooled pre-extracted CNN features. The caption feature $\mathbf{c}$ is the output of a shallow 1D-CNN on top of pre-computed word embeddings. More details are given in Section 5.5.1.

5.4.1 Loss implementation

We explain next how $\mathcal{N}(i)$ is constructed to improve computational efficiency.

At each training iteration, we first sample v unique YouTube video ids. We then sample with replacement a number k of clip-caption pairs from each of these videos. Therefore, we are left with a mini-batch containing $b = kv$ clip-caption pairs, with $v = 32$ and $k = 64$ in practice. In order to not waste computation efforts, we use every sampled mini-batch pair as a negative anchor, *i.e.* $\mathcal{N}(i) = \mathcal{B} \setminus \{i\}, \forall i$.

Doing so, the proportion of negative examples coming from the same video (*intra-video*) is $\frac{k-1}{kv-1}$ while the proportion of negatives from different videos (*inter-video*) is $\frac{k(v-1)}{kv-1}$. A problem with this is that the ratio between *intra* and *inter* video negative examples depends on the number of unique videos sampled and the amount of clip-caption pairs collected per video (respectively v and k). To address this, we follow Hendricks et al. [2017] by re-weighting the inter-video and intra-video contributions inside the triplet loss. For example, in order to sample intra-video triplets with probability $p \in [0,1]$ (and inter-video triplets with probability $1-p$), one can equivalently weight the intra-video triplet losses by: $\alpha = \frac{pk(v-1)}{(1-p)(k-1)}$ (thus ensuring a ratio between intra-video and inter-video negative examples of $\frac{p}{1-p}$). This allows us to fix the intra-video to inter-video negative sampling ratio regardless of v and k. Formally, we define the following weighting function:

$$\alpha_{i,j} = \begin{cases} \frac{pk(v-1)}{(1-p)(k-1)} & \text{if } i \text{ and } j \text{ are from same video,} \\ 1, & \text{otherwise.} \end{cases} \tag{5.5}$$

We then use this weighing function to define the loss:

$$\sum_{i\in\mathcal{B},j\in\mathcal{N}(i)} \alpha_{i,j}\Big[\max(0,\delta + s_{i,j} - s_{i,i}) + \max(0,\delta + s_{j,i} - s_{i,i})\Big].$$

5.5 Experiments

In this section, we demonstrate that a strong joint representation for video and text can be learned from our unlabeled HowTo100M dataset. We provide experimental results for a variety of domains ranging from instructional videos in CrossTask [Zhukov et al., 2019], cooking videos in YouCook2 [Zhou et al., 2018b], generic YouTube videos in MSR-VTT [Xu et al., 2016] to movie video clips in LSMDC [Rohrbach et al., 2017]. Specifically, we evaluate our learned embedding on the tasks of localizing steps in instructional videos of CrossTask [Zhukov et al., 2019] and text-based video retrieval on YouCook2 [Zhou et al., 2018b], MSR-VTT [Xu et al., 2016] and LSMDC [Rohrbach et al., 2017] datasets.

Our *key* findings are the following: **(i)** For instructional video datasets, such as CrossTask [Zhukov et al., 2019] and YouCook2 [Zhou et al., 2018b], our off-the-shelf embedding trained on HowTo100M significantly outperforms state-of-the-art models trained on much smaller and manually-annotated datasets. **(ii)** On generic YouTube videos (MSR-VTT [Xu et al., 2016]), our HowTo100M embedding provides competitive retrieval performance compared to state-of-the-art methods trained on

MSR-VTT. Moreover, we show that fine-tuning our pre-trained embedding model on just a fifth of annotated videos from MSR-VTT outperforms state-of-the-art. **(iii)** We show that fine-tuning our embedding on LSMDC enables generalization to movie videos and scripts despite the large domain gap. **(iv)** Finally, we demonstrate the importance of scale in HowTo100M to learn better joint video-text embeddings.

5.5.1 Implementation details

Video features. We extract frame-level and video-level features with pre-trained 2D and 3D CNNs. 2D features are extracted with the ImageNet pre-trained Resnet-152 [He et al., 2016] at the rate of one frame per second. 3D features are extracted with the Kinetics [Carreira and Zisserman, 2017] pre-trained ResNeXt-101 16-frames model [Hara et al., 2018] to obtain 1.5 features per second. We aggregate features from longer video clips by the temporal max-pooling and concatenate 2D and 3D features to form a single 4096 dimensional vector for each video clip.

Text pre-processing. We preprocess transcribed video narrations by discarding common English stop-words. For the word representations, we use the GoogleNews pre-trained word2vec embedding model [Mikolov et al., 2013].

Training time. Once the video and text features are extracted, training our embedding model on the full HowTo100M dataset is relatively fast and takes less than three days on a single Tesla P100 GPU.

5.5.2 Datasets and evaluation setups

Action step localization. We evaluate localization of action steps in instructional videos on the recent CrossTask dataset [Zhukov et al., 2019]. CrossTask includes 18 tasks and 2.7k instructional videos with manually annotated action segments. Each video may contain multiple segments, corresponding to different actions. It also provides an ordered list of action steps with short natural language descriptions for each task. We apply our model trained only on HowTo100M to the problem of step localization by computing similarity between every frame in the video and the action label names of CrossTask. In order to compare to [Zhukov et al., 2019], we follow a similar inference procedure. We use the same recall metric as in [Zhukov et al., 2019], which is defined by the number of step assignments that fall into the correct ground truth interval, divided by the total number of steps in the video. Videos from the test set of CrossTask are removed from the HowTo100M training set to ensure that they are not observed at training time.

Text-based video retrieval. We also evaluate our learned embedding on the task of video clip retrieval using natural language queries. Given a textual description, the goal is to retrieve representative video clips from a large pool of videos. We evaluate our learned embedding using the standard recall metrics R@1, R@5, R@10 and the median rank (Median R). We provide experimental results for the following domain-specific video description datasets.

YouCook2 [Zhou et al., 2018b] is a cooking video dataset collected from YouTube. It features 89 different recipes and 14k video clips all annotated with textual descriptions collected from paid human workers. Since no descriptions are provided for the test set clips, we evaluate YouCook2 clip retrieval task on the validation clips (3.5k in total). Note that we have taken care to remove the few validation YouCook2 videos that are also present in HowTo100M.

MSR-VTT [Xu et al., 2016] is a dataset of generic videos collected from 257 popular video queries depicting 20 categories (including music, sports or movie) from YouTube. It contains 200k unique video clip-caption pairs, all annotated by paid human workers. We evaluate our model on the MSR-VTT clip retrieval test set used in [Yu et al., 2018] as performance of several other methods is reported on it.

LSMDC [Rohrbach et al., 2017] is a dataset of movie clips. It features 101k unique video clip-caption pairs. All clips are associated with a description that either comes from the movie script or the audio description. We evaluate our model on the official LSMDC test set[2] that contains 1000 video-caption pairs.

[2]`https://sites.google.com/site/describingmovies/lsmdc-2016/movieretrieval`

Negative sampling	M (R@10)	L (R@10)	Y (R@10)	C (AVG Recall)
No intra-negative	**30.1**	12.3	18.1	25.7
With intra-negative	29.6	**14.0**	**24.8**	**33.6**

Table 5.3: Impact of intra-video negative pairs during training. M: MSR-VTT, L: LSMDC, Y: YouCook2, C: CrossTask.

5.5.3 Study of negative pair sampling strategy

We first study the effect of alternative strategies for sampling negative caption-video clip pairs when training our embedding. Table 5.3 shows that using negatives from the same video (intra-negatives) is beneficial as compared to randomly sampling them from other YouTube videos. The improvement is particularly significant on YouCook2 and CrossTask which are more fine-grained datasets than MSR-VTT and LSMDC. For the rest of the chapter, we report numbers using our model trained with the intra-negative sampling strategy.

5.5.4 Scale matters

A natural question is whether the large scale of our dataset is truly required to achieve high performance. To answer this, we train our embedding model on smaller subsets of our dataset. These smaller subsets of HowTo100M are created by gradually decreasing the allowed Youtube search rank (see the paragraph on data collection in Section 5.3.1 for more details) for training videos. We experiment with the following rank thresholds: top 2 (15k videos), top 3 (28k videos), top 5 (52k videos), top 10 (104k videos), top 20 (197k videos), top 40 (364k videos), top 80 (648k videos) and top 200 (entire HowTo100M dataset). This process ensures that we subsample training videos that are more likely to be relevant to the queried task as we reduce the size of the training dataset. Figure 5-3 shows average recall on CrossTask and the R@10 clip retrieval results on LSMDC, MSR-VTT and YouCook2 when varying the size of the training dataset. There is a clear improvement over all evaluated tasks with the gradual increase in the amount of training data. Interestingly, we do not observe any saturation, hence we can expect further improvements by collecting even more

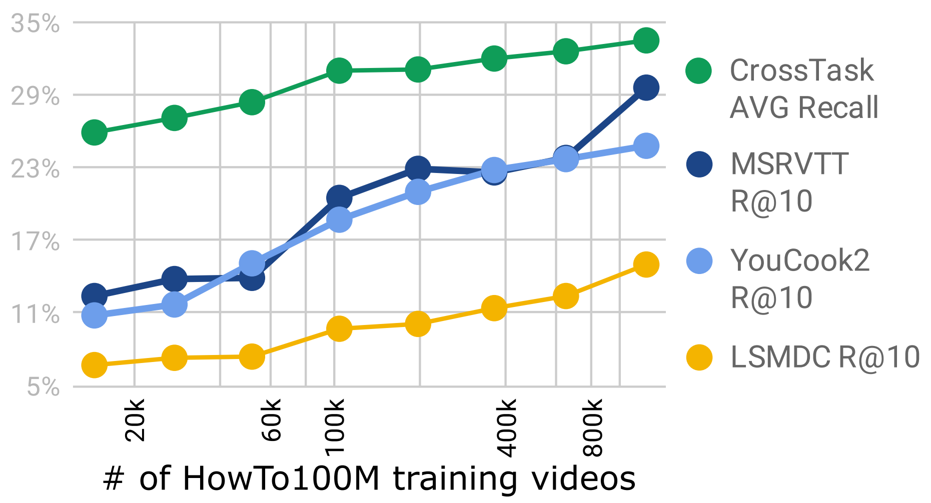


Figure 5-3: Retrieval and step localization results when varying the training size of our HowTo100M dataset.

readily-available and unlabeled video data.

5.5.5 Comparison with state-of-the-art

CrossTask. We compare our off-the-shelf embedding trained on HowTo100M against methods proposed by [Alayrac et al., 2016] and [Zhukov et al., 2019] which is the current state-of-the-art on CrossTask for weakly supervised methods. Note that [Zhukov et al., 2019] have access to the ordered list of action labels at the task level and narrations are the only form of supervision during training. We also report the fully-supervised upper-bound from [Zhukov et al., 2019] obtained with a model that has been trained on action segments with ground truth annotation. The results are shown in Table 5.4. Our approach significantly outperforms the state-of-the-art, even though it has not been specifically designed for the task of step localization in videos. The improvement made by our method is consistent across all tasks (with the exception of *Make Meringue*), showing that the trained model is not biased towards any specific domain. The recall is above 30% for most tasks with the significant improvement observed for the "*Add Oil to a Car*" task (6.4% to 30.7% boost in recall). Note that our method also outperforms the fully-supervised upper bound [Zhukov et al., 2019]

	Make Kimchi Rice	Pickle Cucumber	Make Banana Ice Cream	Grill Steak	Jack Up Car	Make Jello Shots	Change Tire	Make Lemonade	Add Oil to Car	Make Latte	Build Shelves	Make Taco Salad	Make French Toast	Make Irish Coffee	Make Strawberry Cake	Make Pancakes	Make Meringue	Make Fish Curry	Average
[Zhukov et al., 2019] (Supervised)	19.1	25.3	38.0	37.5	25.7	28.2	54.3	25.8	18.3	31.2	47.7	12.0	39.5	23.4	30.9	41.1	53.4	17.3	31.6
[Alayrac et al., 2016]	15.6	10.6	7.5	14.2	9.3	11.8	17.3	13.1	6.4	12.9	27.2	9.2	15.7	8.6	16.3	13.0	23.2	7.4	13.3
[Zhukov et al., 2019]	13.3	18.0	23.4	23.1	16.9	16.5	30.7	21.6	4.6	19.5	35.3	10.0	32.3	13.8	29.5	37.6	**43.0**	13.3	22.4
Ours (HowTo100M only trained)	**33.5**	**27.1**	**36.6**	**37.9**	**24.1**	**35.6**	**32.7**	**35.1**	**30.7**	**28.5**	**43.2**	**19.8**	**34.7**	**33.6**	**40.4**	**41.6**	41.9	**27.4**	**33.6**

Table 5.4: Step localization results on CrossTask [Zhukov et al., 2019] instructional video dataset.

Method	Trainset	R@1	R@5	R@10	Median R
Random	None	0.03	0.15	0.3	1675
HGLMM FV CCA [Klein et al., 2015]	YouCook2	4.6	14.3	21.6	75
Ours	YouCook2	4.2	13.7	21.5	65
Ours	HowTo100M	6.1	17.3	24.8	46
Ours	PT: HowTo100M FT: YouCook2	**8.2**	**24.5**	**35.3**	**24**

Table 5.5: YouCook2 clip retrieval results. PT denotes: pre-trained, while FT denotes: fine-tuned.

on average. Thus, we conclude that training on a large amount of narrated videos is better than training a step localization model on a small but carefully annotated training set.

YouCook2 [Zhou et al., 2018b] does not provide an official benchmark nor any reported number for clip retrieval. As a consequence, we have applied a state-of-the-art text-video embedding model from [Klein et al., 2015] (HGLMM FV CCA) on YouCook2 using our features. We also report results of our model trained on YouCook2 instead of HowTo100M in Table 5.5. First, we notice that our off-the-shelf model trained on HowTo100M significantly outperforms both the exact same model directly trained on YouCook2 and [Klein et al., 2015]. Furthermore, fine-tuning our model pre-trained on HowTo100M on YouCook2 results in a significant improvement of 13.7 % in R@10 against [Klein et al., 2015]. In conclusion, we show that the off-the-shelf HowTo100M trained model can outperform state-of-the-art on this domain specific instructional video dataset. Moreover, we demonstrate that our model can get further benefits from fine-tuning.

Method	Trainset	R@1	R@5	R@10	Median R
Random	None	0.1	0.5	1.0	500
C+LSTM+SA+FC7 [Torabi et al., 2016]	MSR-VTT	4.2	12.9	19.9	55
VSE-LSTM [Kiros et al., 2014]	MSR-VTT	3.8	12.7	17.1	66
SNUVL [Yu et al., 2016b]	MSR-VTT	3.5	15.9	23.8	44
[Kauman et al., 2017]	MSR-VTT	4.7	16.6	24.1	41
CT-SAN [Yu et al., 2017b]	MSR-VTT	4.4	16.6	22.3	35
JSFusion [Yu et al., 2018]	MSR-VTT	10.2	31.2	43.2	13
Ours	HowTo100M	7.5	21.2	29.6	38
Ours	MSR-VTT	12.1	35.0	48.0	12
Ours	PT: HowTo100M FT: MSR-VTT	**14.9**	**40.2**	**52.8**	**9**

Table 5.6: MSR-VTT clip retrieval results. PT denotes: pre-trained, while FT denotes: fine-tuned.

MSR-VTT. We compare our model trained on **(i)** HowTo100M only, **(ii)** MSR-VTT only and **(iii)** pre-trained on HowTo100M and then fine-tuned on MSR-VTT against prior work that directly uses MSR-VTT for training (reproduced in [Yu et al., 2018]) in Table 5.6. Our off-the-shelf HowTo100M model outperforms [Kauman et al., 2017; Kiros et al., 2014; Torabi et al., 2016; Yu et al., 2016b, 2017b] that are *directly trained* on MSR-VTT. Here again, after fine-tuning the HowTo100M pre-trained model on MSR-VTT, we observe a significant improvement over the state-of-the-art JSFusion [Yu et al., 2018] trained on MSR-VTT. However, as opposed to instructional videos (CrossTask) and cooking videos (YouCook2), training our model directly on MSR-VTT performs better than our off-the-shelf model trained on HowTo100M. We believe this is due to MSR-VTT videos being generic Youtube videos that are different from the instructional or VLOG type of videos that dominate HowTo100M. In Figure 5-4, we also investigate the impact on performance at various amounts of supervision when fine-tuning our pre-trained model. It shows that state-of-the-art performance [Yu et al., 2018] can be attained with *only* 20% of MSR-VTT samples. This has great practical implications as comparable performance can be obtained using significantly reduced annotation.

LSMDC. Finally, we compare to state-of-the-art on LSMDC in Table 5.7. This dataset is even more challenging as movie clips are quite distinct from HowTo100M

Method	Trainset	R@1	R@5	R@10	Median R
Random	None	0.1	0.5	1.0	500
C+LSTM+SA+FC7 [Torabi et al., 2016]	LSMDC	4.3	12.6	18.9	98
VSE-LSTM [Kiros et al., 2014]	LSMDC	3.1	10.4	16.5	79
SNUVL [Yu et al., 2016b]	LSMDC	3.6	14.7	23.9	50
[Kauman et al., 2017]	LSMDC	4.7	15.9	23.4	64
CT-SAN [Yu et al., 2017b]	LSMDC	4.5	14.1	20.9	67
JSFusion [Yu et al., 2018]	LSMDC	**9.1**	**21.2**	**34.1**	**36**
Ours	HowTo100M	4.0	9.8	14.0	137
Ours	LSMDC	**7.2**	18.3	25.0	44
Ours	PT: HowTo100M FT: LSMDC	7.1	**19.6**	**27.9**	**40**

Table 5.7: LSMDC clip retrieval results. PT denotes: pre-trained, while FT denotes: fine-tuned.

videos. We compare against several other prior works that have been reproduced in [Yu et al., 2018] and are trained directly on LSMDC. Here again, we see that pre-training our model on HowTo100M and fine-tuning it on LSMDC also provides improvements upon a model directly trained on LSMDC. This finding is interesting and shows that a HowTo100M pre-trained model can still be useful when fine-tuned on videos from a different domain.

5.5.6 Cross-dataset fine-tuning evaluation

In this section, we evaluate the advantage of HowTo100M for pre-training compared to pre-training on other smaller datasets. Figure 5-5 shows evaluation on YouCook2, MSR-VTT and LSMDC clip retrieval (R@10) using no pre-training (No PT), using pre-training on YouCook2, MSR-VTT, LSMDC and HowTo100M datasets while fine-tuning to the target dataset. For all evaluated datasets, pre-training on HowTo100M prior to fine-tuning on the target dataset consistently yields best results.

5.5.7 Sampling strategy for positive pairs

As discussed in this chapter, narrations need not necessarily describe what is seen in the video. As a consequence, some captions from HowTo100M do not correlate with

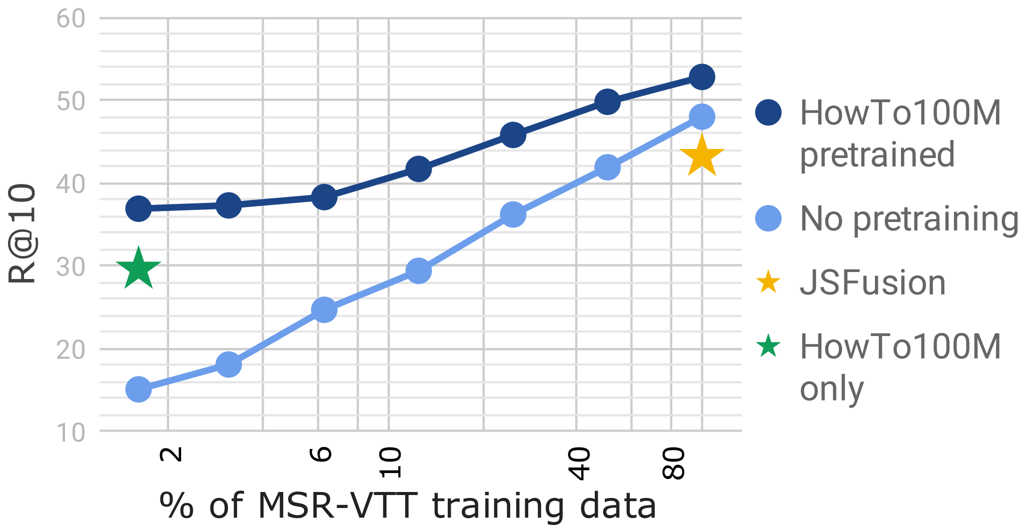


Figure 5-4: Evaluation of fine-tuning a HowTo100M pre-trained model with varying amounts of MSR-VTT supervision for text-to-video clip retrieval.

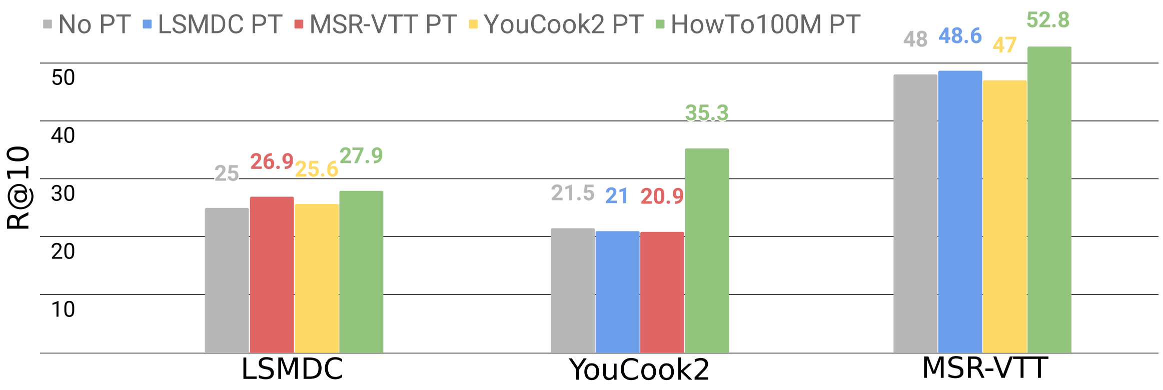


Figure 5-5: Results of clip retrieval by pre-training models on different datasets. Evaluation on LSMDC, YouCook2 and MSR-VTT.

their corresponding video clips (see Figure 5-6). To deal with this noisy data, we tried a sampling strategy for positive pairs that aims to discard non-relevant video-caption pairs during training. Inspired by multiple instance learning, our idea is to select a subset of top scoring clip-caption training pairs within each video.

In particular, given a video with N video clip-caption pairs $\{(V_i, C_i)\}_{i\in[1,N]}$, we first compute the similarity scores of all the N pairs: $s(V_i, C_i)$ using the current model parameters. We then use a pre-defined max-pool rate $r \in [0, 1]$ of the highest scoring positive training pairs $\{(V_i, C_i)\}_{i\in[1,N]}$ within each video. For example, at $r = 0.5$ we retain the high scoring half of all N pairs for training.

Max pool rate (r)	M (R@10)	L (R@10)	Y (R@10)
0.2	21.9	13.9	19.7
0.5	25.2	12.6	23.5
0.9	27.3	12.6	23.9
1.0 (no max pool)	**29.6**	**14.0**	**24.8**

Table 5.8: Study of positive pair sampling. When max pool rate r is below 1.0 only the proportion r of top scoring clip-caption pairs are used for learning. We report R@10 retrieval results from M: MSR-VTT, L: LSMDC, Y: YouCook2.

MP rate	RS rate	M (R@10)	L (R@10)	Y (R@10)
1.0	0.5	**28.8**	**14.3**	**24.2**
0.5	1.0	25.2	12.6	23.5

Table 5.9: Study of Random Sampling (RS) vs. Max Pool (MP) sampling of positive clip-caption pairs. We report R@10 retrieval results from M: MSR-VTT, L: LSMDC, Y: YouCook2.

Table 5.8 shows results of our positive sampling strategy when varying the max pool rate r with evaluation on video clip retrieval. For example, $r = 1.0$ means that no sampling strategy is applied as we keep all N pairs as potential candidates. Interestingly, in our case, carefully selecting the positive pairs does not improve our model as the best results are obtained with $r = 1.0$. Note that decreasing the max pool rate also decreases the number of triplet losses computed within a mini-batch by the same rate. To show that the number of triplet losses computed for each mini-batch does not impact the overall performance, we have performed a sanity check experiment in Table 5.9 in which we also replaced the max pool sampling by random sampling of pairs for $r = 0.5$. The results with random sampling at $r = 0.5$ are very similar to the results obtained with no max pool sampling (r=1.0) as shown in Table 5.8, which confirms our finding that our model is relatively robust to the noisy positive pairs. We think this could be attributed to the fact our model is shallow and is trained on a large amount of data.

5.5.8 Qualitative results

Figure 5-7 illustrates examples of retrieved video clips from HowTo100M using our trained joint text-video embedding. For example, our learned representation can correctly distinguish between queries *Cut paper* and *Cut wood*. A demo of the retrieval system is available online [icc, 2019].

5.6 Conclusion

We have introduced introduced in this chapter: HowTo100M, a video dataset with more than 130M video clips, extracted from 1.2M narrated web videos of people performing complex visual tasks. Our data collection method is fast, scalable and *does not require any manual annotation*. We use this dataset to learn a joint text-video embedding by leveraging more than 130M video clip-caption pairs. We have shown through various experiments that our learned embedding can perform better compared to models trained on existing carefully annotated but smaller video description datasets.

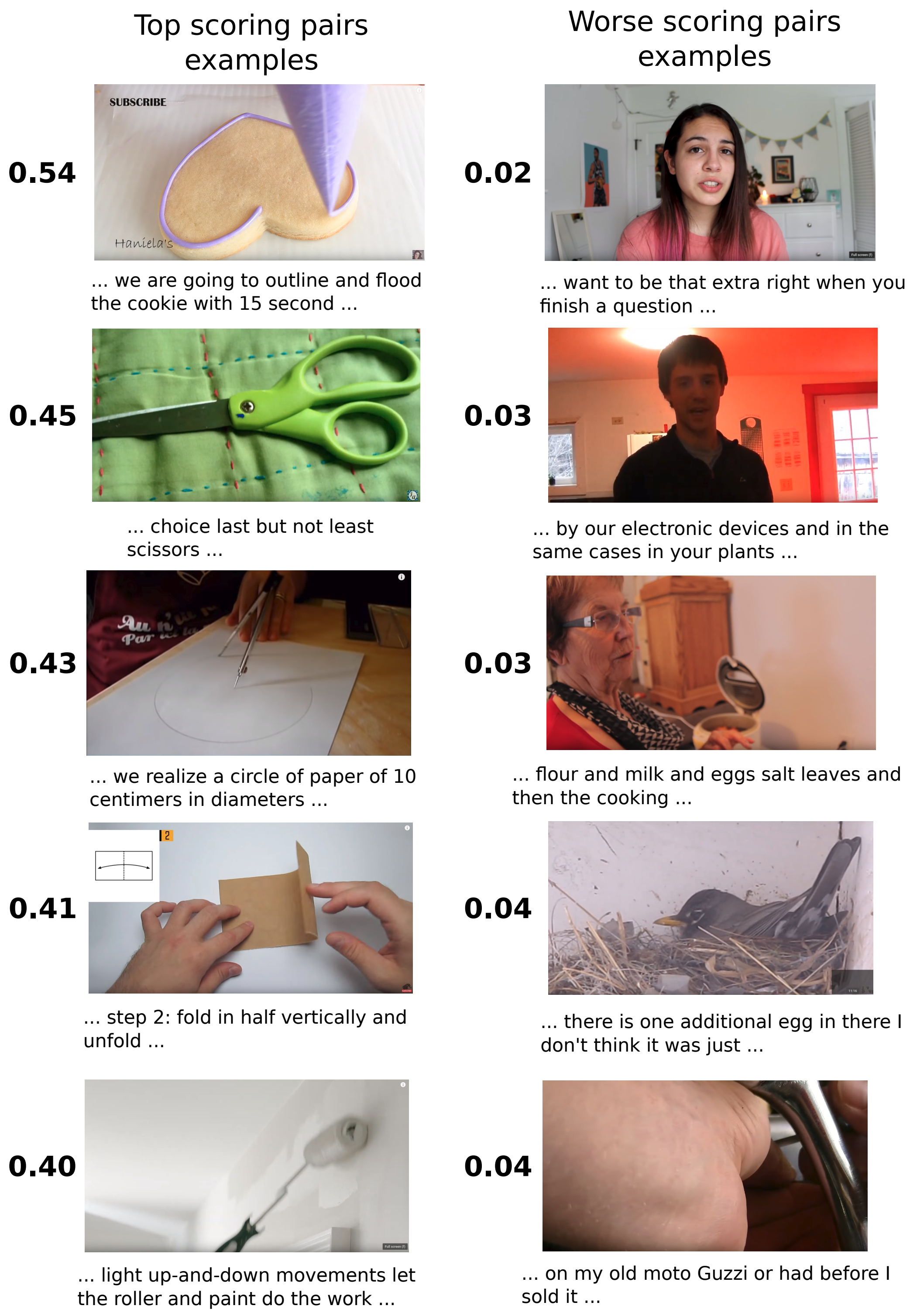

Top scoring pairs examples
Worse scoring pairs examples
0.54
... we are going to outline and flood the cookie with 15 second ...
0.02
... want to be that extra right when you finish a question ...
0.45
... choice last but not least scissors ...
0.03
... by our electronic devices and in the same cases in your plants ...
0.43
... we realize a circle of paper of 10 centimers in diameters ...
0.03
... flour and milk and eggs salt leaves and then the cooking ...
0.41
... step 2: fold in half vertically and unfold ...
0.04
... there is one additional egg in there I don't think it was just ...
0.40
... light up-and-down movements let the roller and paint do the work ...
0.04
... on my old moto Guzzi or had before I sold it ...

Figure 5-6: We illustrate examples of high and low scoring clip-caption pairs. Examples from the left column show pairs where the caption visually describes what is seen in the corresponding video clip. On the other hand, low scoring pairs from the right column have captions that do not match visual content.

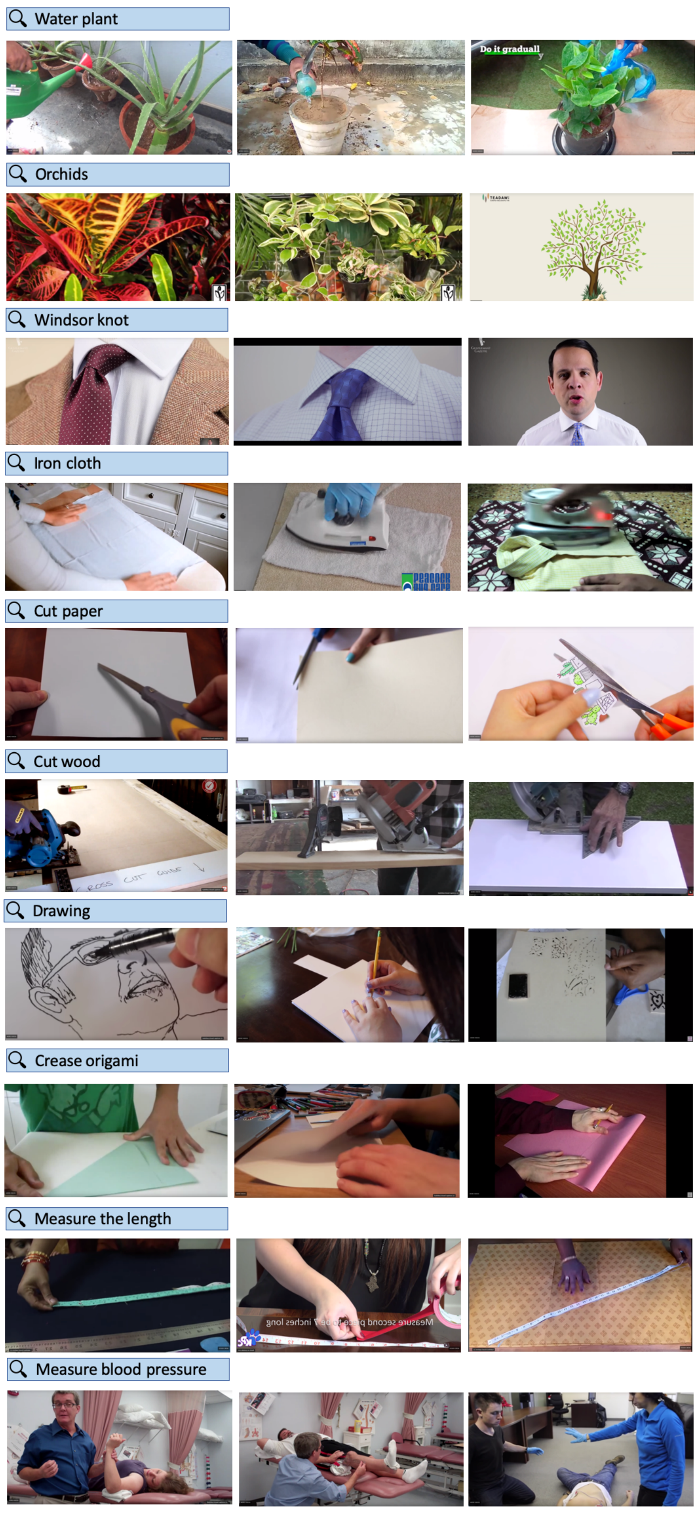


Figure 5-7: Example video-clip retrieval results on HowTo100M using our trained joint embedding.

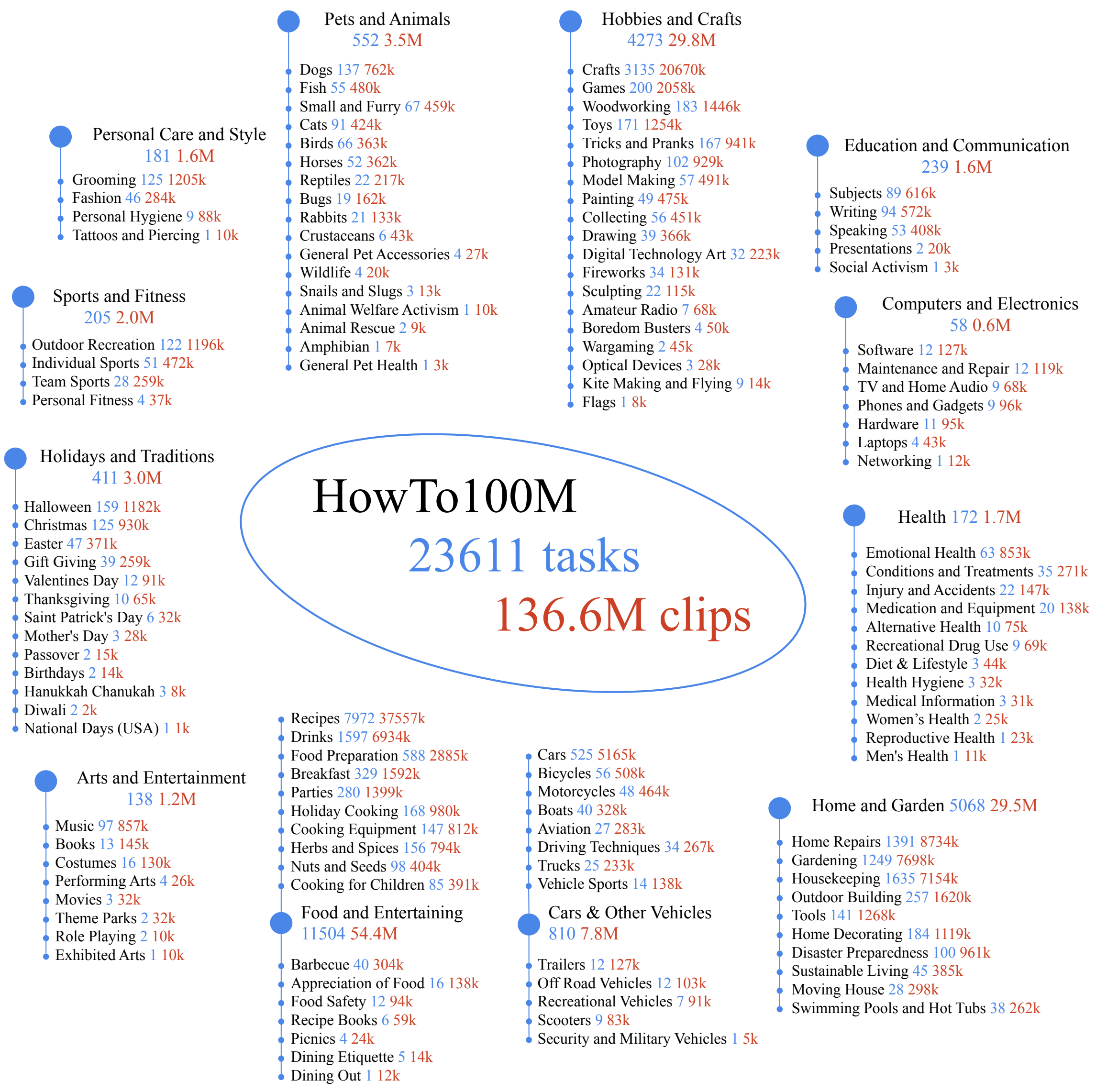


Figure 5-8: The first two levels of hierarchy of tasks in the HowTo100M dataset. Our dataset includes 12 categories from WikiHow containing 129 subcategories. For each (sub)category we show the total number of collected tasks and clips. This hierarchy of tasks in our dataset follows the WikiHow structure. Please recall that abstract tasks such as Choosing a gift or Meeting new friends, were not considered and were removed from the WikiHow hierarchy semi-automatically by verb analysis, as described in the chapter. As a result, the category tree is imbalanced. For example, the *Dining Out* subcategory includes only one physical task (*Fix a Shaky Table at a Restaurant*), while *Recipes* subcategory from the same level of the hierarchy includes a large number of tasks and clips.

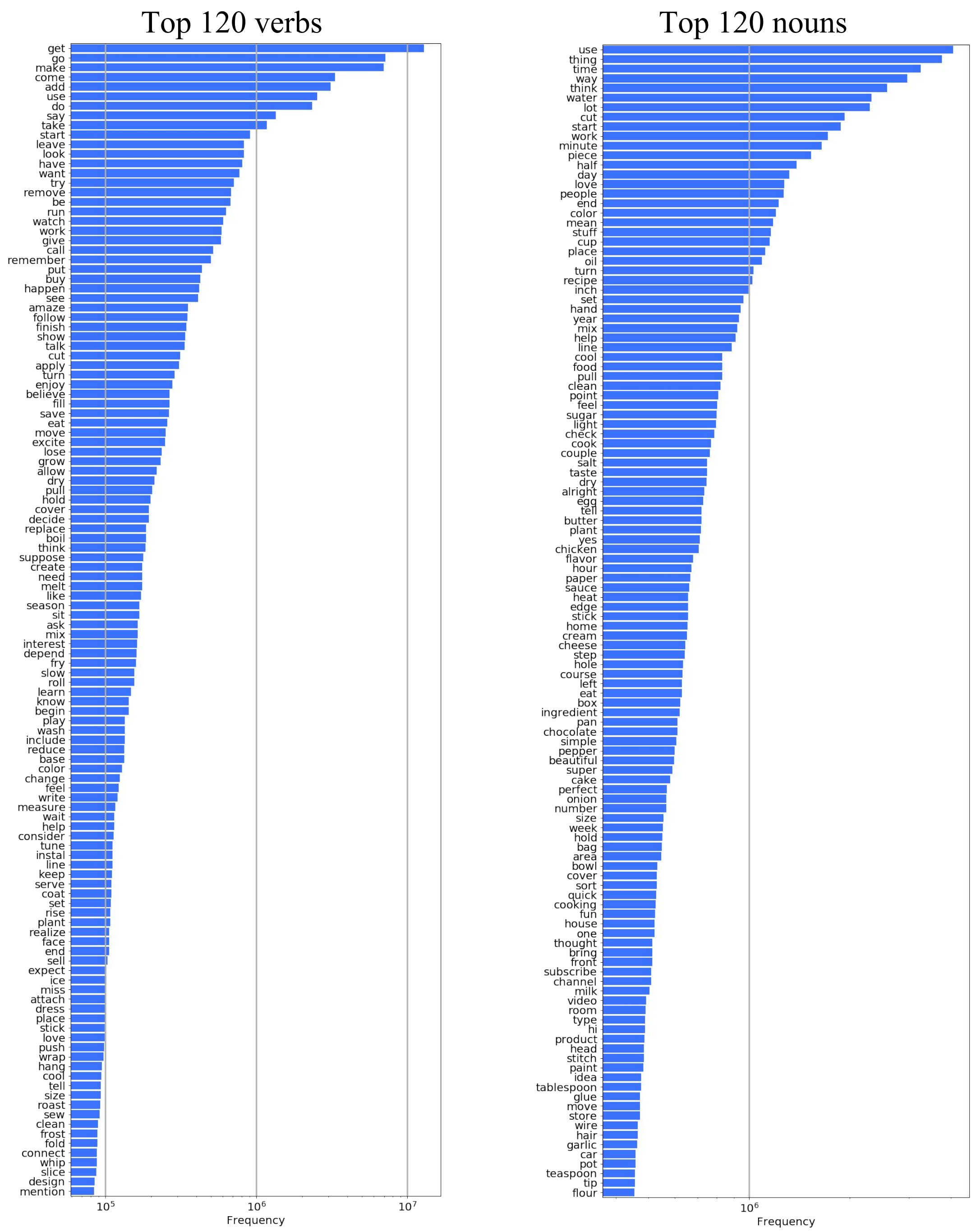


Figure 5-9: Frequencies of the top 120 most commonly occurring nouns and verbs in our dataset. Note that our dataset is biased towards physical actions, with verbs such as *get*, *go* and *make* being the most frequent, while verbs, such as *be*, *know* and *think* are less frequent than in common English. Top nouns show the dominant topics in our instructional videos. In particular, many cooking-related words, such as *water*, *oil* and *sugar* occur with high frequency.

Chapter 6

End-to-End Learning of Visual Representations from Uncurated Instructional Videos

In the previous chapters, we have proposed different approaches for weakly supervised learning of video models from readily available metadata in the form of natural language. Most representations build on CNN image and video representations pretrain on manually annotated visual datasets such as ImageNet or Kinetics. This is problematic as it means the developed approaches still rely on a large amount of manually annotated images or videos. In this chapter, we push the limit of learning from readily available data by training a model *from scratch* on the weakly annotated HowTo100M dataset introduced in Chapter 5. In particular, we propose a new learning approach, *MIL-NCE*, capable of addressing misalignment between the video and automatically transcribed narration. With this approach we are able to learn strong video representations from scratch, without the need for any manual annotation. We evaluate our representations on a wide range of four downstream tasks over eight datasets: action recognition (HMDB-51, UCF-101, Kinetics-700), text-to-video retrieval (YouCook2, MSR-VTT), action localization (YouTube-8M Segments, CrossTask) and action segmentation (COIN). Our method outperforms all published self-supervised approaches for these tasks as well as several fully supervised baselines.

6.1 Introduction

Vision and language play an important role in the way humans learn to associate visual entities to abstract concepts and vice versa. This has also become the *de facto* way to successfully train computer vision models. Indeed, from *classification* where images are categorized based on a fixed list of words to the recent *captioning* tasks where images or videos are annotated with rich language descriptions, this interplay is one of the driving forces behind recent progress in the field. However, one of the main limitations of this approach is that it requires manually annotating large collections of visual data.

Manual annotation is both cumbersome and expensive. Moreover, for videos, which are the main focus of this chapter, annotation is also even more challenging due to the ambiguities of choosing the right vocabulary of actions and annotating action intervals in video. This significantly limits the scale at which fully supervised video datasets can be obtained and hence slows down the quest to improve visual representations. Recent work has proposed a promising alternative to this fully supervised approach: leveraging narrated videos that are readily available at scale on the web.

Of particular interest, the recent HowTo100M dataset [Miech et al., 2019b] contains more than 100 million pairs of video clips and associated narrations. It was automatically collected by querying YouTube for instructional videos. Such videos usually depict someone explaining orally how to perform a complex human activity, *e.g.* preparing a particular meal or repairing a car. Our objective in this chapter is to learn strong video representations using *only* this narrated material.

End-to-end learning from instructional videos is a highly challenging task. Indeed, these videos are made in general with the goal of maximizing the number of views, and with no specific intention to provide a training signal for machine learning algorithms. This means that the supervision present in the narration is only weak and noisy. Among typical sources of noise, the prominent one by far is the weak *alignment* between the video and the language: although for the most part the spoken words

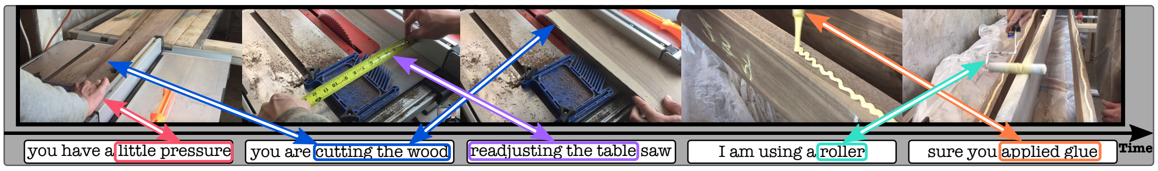
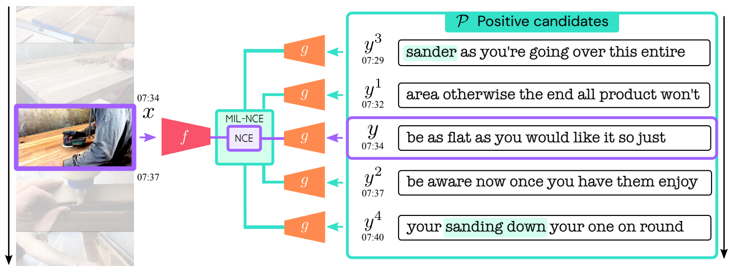


Figure 6-1: We describe an efficient approach to learn visual representations from highly misaligned and noisy narrations automatically extracted from instructional videos. Our video representations are learnt *from scratch without relying on any manually annotated visual dataset* yet outperform all self-supervised and many fully-supervised methods on several video recognition benchmarks.

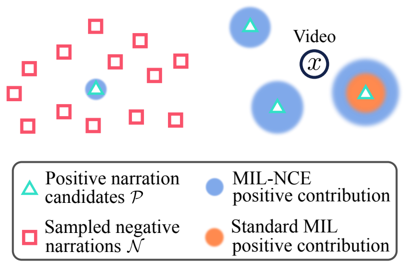
correlate with what is happening in these videos, this alignment is far from perfect. People might talk about something *before* actually demonstrating it, but they might also *omit* to talk about something that is happening because it is clear enough visually. Conversely they could only mention an action without showing it in the case where the step is not essential or trivial to convey with language alone. This is without even considering the *irrelevant* information given throughout the video (*e.g.* jokes or credits) as well as the general difficulty of working with spoken language obtained from potentially *erroneous* speech recognition algorithm as opposed to written text.

In this chapter, we propose a bespoke training loss, dubbed *MIL-NCE* as it inherits from Multiple Instance Learning (MIL) *and* Noise Contrastive Estimation (NCE). Our method is capable of addressing visually misaligned narrations from uncurated instructional videos as illustrated in Figure 6-1. Equipped with this novel training scheme and a simple joint video and text embedding model, we show that we can successfully train video representations *from scratch* directly from pixels on the HowTo100M [Miech et al., 2019b] dataset. To demonstrate the quality of the learnt representations, we employ an extensive set of evaluation benchmarks on a wide variety of video understanding tasks: action recognition (HMDB-51, UCF-101, Kinetics-700), text-to-video retrieval (YouCook2, MSR-VTT), action localization (YouTube-8M Segments, CrossTask) and action segmentation (COIN). Notably, our learnt video representations outperform fully supervised baselines trained on Kinetics or ImageNet for several of the tasks. We also show improvements over other self-supervised approaches on HMDB51 and UCF101 even without fine-tuning the learnt representa-



$\mathcal{P}$ Positive candidates
y^3 07:29 sander as you're going over this entire
y^1 07:32 area otherwise the end all product won't
y 07:34 be as flat as you would like it so just
y^2 07:37 be aware now once you have them enjoy
y^4 07:40 your sanding down your one on round
07:34
x
07:37
f
MIL-NCE
NCE
g

(a) **Examples of positive candidates**



Video
x
Positive narration candidates $\mathcal{P}$
MIL-NCE positive contribution
Sampled negative narrations $\mathcal{N}$
Standard MIL positive contribution

(b) **Illustration of *MIL-NCE***

Figure 6-2: **Left.** Our MIL-NCE makes it possible to consider a set of multiple positive candidate pairs $\{(x, y), (x, y^1), \dots, (x, y^4)\}$ while the standard NCE approach would only consider the single (x, y) training pair and miss the visually grounded object description `sander` from pair (x, y^3) or the action description `sanding down` from (x, y^4). **Right.** Given a video x and an associated set of positive narration candidates $\mathcal{P}$ (green triangles) that may or may not be correct, our *MIL-NCE* selects *multiple* correct positives (large blue areas) while downweighting incorrect positives (smaller blue areas) based on a discriminative ratio against negatives $\mathcal{N}$ (red squares). In contrast, traditional MIL considers only one positive (orange circle) while discarding the rest.

tions. Finally, by leveraging the joint video and text representations, our off-the-shelf trained model also reaches state-of-the-art results on YouCook2 and CrossTask, without any training on the target datasets.

Contributions. The contributions of this chapter are threefold. **(i)** We propose a method to learn a joint text video embedding in an end-to-end fashion from unlabelled, uncurated narrated videos using the recently introduced HowTo100M [Miech et al., 2019b] dataset. In particular, we introduce a specific loss, dubbed *MIL-NCE* for Multiple Instance Learning Noise Contrastive Estimation, that enables the learning to cope with the highly misaligned narration descriptions. **(ii)** We provide a thorough ablation study to quantitatively assess the importance of the different design choices of the approach. **(iii)** Finally, we demonstrate that the representations thus obtained are competitive with their strongly supervised counterparts on four downstream tasks over eight video datasets.

6.2 Related work

Learning visual representations from unlabeled videos. As labeling videos is cumbersome, expensive and not scalable, a significant number of prior works have studied the task of learning visual representations from unlabeled videos. Currently, the most effective approach is to collect a large amount of data from social media and use the available metadata as supervision [Abu-El-Haija et al., 2016; Ghadiyaram et al., 2019]. However, this metadata is often in the form of keywords or tags, rather than (spoken) natural language considered in this chapter. In addition, the meta data is often platform dependent and rarely publicly available. Self-supervised approaches do not suffer from these issues as the idea is to define a supervised proxy task using labels directly generated from videos. Some of these tasks include: temporal ordering of video clips or frames [Fernando et al., 2017; Lee et al., 2017; Misra et al., 2016; Xu et al., 2019], predicting geometric transformations [Jing and Tian, 2018], maximizing the mutual information of multiple views [Tian et al., 2019], predicting motion and appearance [Wang et al., 2019a], predicting the future, the past or a portion of masked input in the feature space [Han et al., 2019; Sun et al., 2019a; Vondrick et al., 2016], colorizing videos [Vondrick et al., 2018], predicting 3D geometry from synthetic data [Gan et al., 2019], predicting the audio in a feature space [Arandjelović and Zisserman, 2018; Korbar et al., 2018] or tasks leveraging temporal cycle consistency [Dwibedi et al., 2019; Wang and Gupta, 2015]. In this chapter, our proxy task is supervised by the automatic speech recognition (ASR) applied to narrated instructional videos. The nature of this supervision has the potential to also provide semantic information [Miech et al., 2019b; Sanabria et al., 2018], which is often missing in works that only exploit pixel-wise cues. Moreover, most of the top performing prior works only study their method on curated video datasets (*e.g.* Kinetics [Carreira and Zisserman, 2017]) where labels have been removed. However, this is not truly learning from unlabeled data as these videos have been carefully selected and verified to belong to classes of interests. [Caron et al., 2019] further explain the performance gap between training on such curated data versus uncurated ones, truly available at scale. Instead, our approach focuses on the learning of representations *only from uncurated videos.*

Vision, speech and language. A common alternative to training visual models using manually defined sets of labels is to exploit semantic supervision from natural language or speech. Numerous prior works [Chowdhury et al., 2018; Dong et al., 2019; Gong et al., 2014a,b; Klein et al., 2015; Miech et al., 2018; Mithun et al., 2018; Pan et al., 2016b; Plummer et al., 2017; Xu et al., 2015a; Wang et al., 2018a, 2016b; Wray et al., 2019; Wu et al., 2017] have used image / video description datasets [Lin et al., 2014; Plummer et al., 2015; Rohrbach et al., 2017; Xu et al., 2016; Zhou et al., 2018b] to learn an embedding space where visual and textual data are close only if they are semantically similar. These methods either rely on manually annotated image / video description datasets, or leverage representations already pre-trained on manually labelled datasets (*e.g.* ImageNet [Russakovsky et al., 2015] or Kinetics [Carreira and Zisserman, 2017]). In contrast, in this chapter *no manually annotated visual data is involved at any stage of our approach.* To avoid labeling visual data, several approaches have leveraged audio transcripts obtained from narrated videos using automatic speech recognition (ASR) as a way to supervise video models for object detection [Amrani et al., 2019; Chen et al., 2017a; Moriya et al., 2019], captioning [Hessel et al., 2019; Sun et al., 2019b], classification [Alayrac et al., 2016; Kuehne et al., 2019; Malmaud et al., 2015; Yu et al., 2014], summarization [Palaskar et al., 2019] or retrieval [Miech et al., 2019b] using large-scale narrated video datasets such as How2 [Sanabria et al., 2018] or HowTo100M [Miech et al., 2019b]. Others [Boggust et al., 2019; Harwath et al., 2018] have investigated learning from narrated videos by directly using the raw speech waveform instead of generating transcriptions. Most related to us is the work of [Miech et al., 2019b] who trained a joint video and text embedding from uncurated instructional videos [Miech et al., 2019b]. However, as opposed to this chapter, they do not model any misalignment issue encountered when training on such videos and rely on visual representations pretrained on Kinetics-400 and ImageNet. Building on this work, [Sun et al., 2019a] have used a contrastive bidirectional transformer (CBT) to learn long term contextual video representations from instructional videos. All these works use a visual representation pre-trained on either Kinetics or ImageNet when training on such narrated videos. In contrast, the key innovation in this chapter is that we demonstrate learning a generic video representation as well as a joint video-text embedding *from scratch*, without pre-training on manually annotated video or image datasets.

Multiple instance learning for video understanding. Multiple instance learning methods have been employed in many weakly-supervised video understanding problems including: person recognition in movies using scripts [Bojanowski et al., 2013; Miech et al., 2017a; Parkhi et al., 2015a], anomaly detection [Sultani et al., 2018], weakly supervised action classification [Leung et al., 2011; Shapovalova et al., 2012] and localization [Chéron et al., 2018; Duchenne et al., 2009; Weinzaepfel et al., 2016], co-reference resolution of characters in TV series [Ramanathan et al., 2014] or object tracking [Babenko et al., 2009]. These methods often rely on some form of max-pooling (*i.e.* MIL-SVM [Andrews et al., 2003]) or discriminative clustering (*i.e.* DIFFRAC [Bach and Harchaoui, 2007]) to resolve the label ambiguities, and have used mostly linear (or shallow) models. In this chapter, we present MIL-NCE, a new approach marrying the noise contrastive estimation (NCE) framework [Gutmann and Hyvärinen, 2010] with multiple instance learning [Dietterich et al., 1997]. We show that MIL-NCE is well-suited to learn deep visual representations from scratch using weak and noisy training signals available in uncurated instructional videos.

6.3 Leveraging Uncurated Instructional Videos

This section describes the proposed approach to train joint video and text embeddings from unlabeled narrated videos in an end-to-end fashion. To start with, we are given n pairs of video clips and associated narrations. In practice, a pair is composed of a short 3.2 seconds video clip (32 frames at 10 FPS) together with a small number of words (not exceeding 16) that correspond to what the person is saying in the video. For example, someone might be *sanding wood* while mentioning the action "`sanding down`" or the object "`sander`" as illustrated in Figure 6-2a. Given this input, our goal is to *learn a joint embedding space* where similarity between the narration and video embedding is high when the text and visual content are semantically similar and low otherwise, and we wish to learn this starting from raw pixels in the video and text descriptions. As illustrated in Figure 6-1, this is a very challenging problem due to the often severely misaligned visual descriptions.

In this chapter, we address this issue by introducing the *MIL-NCE* objective:

$$\max_{f,g} \sum_{i=1}^{n} \log \left(\frac{\sum\limits_{(x,y)\in\mathcal{P}_i} e^{f(x)^\top g(y)}}{\sum\limits_{(x,y)\in\mathcal{P}_i} e^{f(x)^\top g(y)} + \sum\limits_{(x',y')\sim\mathcal{N}_i} e^{f(x')^\top g(y')}} \right) \tag{6.1}$$

where x represents a video clip and y a narration. f and g are the two embedding functions that respectively operate over video and text. Given a specific sample i-th, we construct $\mathcal{P}_i$ to be a valid set of *positive* video/narration candidate pairs (see Figure 6-2) while $\mathcal{N}_i$ conversely refers to an associated set of negative video/narration pairs. This objective function implies *maximizing the ratio* of the sum of the positive candidate scores from $\mathcal{P}_i$ to the sum of the scores of *all* negatives sampled from $\mathcal{N}_i$, where the score is measured by the exponentiated dot product of the corresponding video and language embeddings, $f(x)$ and $g(y)$.

In the following, we describe more precisely the motivation behind the *MIL-NCE* objective (6.1). First, Section 6.3.1 introduces the chosen probabilistic model for joint text and video embedding. Given that model, Section 6.3.2 details the choice behind the training objective (6.1) explaining how it is specifically adapted to handle the misalignment noise inherent in narrated videos in comparison with existing approaches.

6.3.1 A simple joint probabilistic model

In the following, $x \in \mathcal{X}$ stands for a video clip and $y \in \mathcal{Y}$ for a narration. Given a set of n pairs of video clips and associated narrations $\{(x_i, y_i)\}_{i=1}^{n} \in (\mathcal{X} \times \mathcal{Y})^n$ sampled from the joint data distribution $P(\mathcal{X} \times \mathcal{Y})$, our goal is to learn a joint embedding space where semantically related videos and texts are close and far away otherwise.

Formally, we learn two parametrized mappings: $f : \mathcal{X} \to \mathbb{R}^d$ maps a video clip x into a d-dimensional vector $f(x) \in \mathbb{R}^d$, and $g : \mathcal{Y} \to \mathbb{R}^d$ maps a narration y into the same d-dimensional vector space, $g(y) \in \mathbb{R}^d$. We assume that we can estimate up to a constant factor the joint probability of a pair of video and narration (x, y) by exponentiating the dot product of the two embeddings:

$$p(x, y; f, g) \propto e^{f(x)^\top g(y)}. \tag{6.2}$$

In this chapter, f takes the form of a CNN that runs over a fixed-length clip. For g, we consider simple sentence based models that transform a set of words into a single vector. Note, for simplicity and with a slight abuse of notation, we refer to f (or g) as both a function *and* the parameters that define it. Also, we will refer to (6.2) as simply $p(x, y)$, *i.e.* we keep the dependence in f and g *implicit* for the clarity of simpler equations. More details about the exact architecture of the models are provided in Section 6.4.

6.3.2 Learning from uncurated data: MIL-NCE

Recall that our goal is to learn a joint video and text representation only from *uncurated* narrated videos. In this section, we start by detailing why this is a highly challenging endeavor due to misalignments present in that data. Next, we explain how the introduced *MIL-NCE* objective (6.1) enables to learn despite that noise. Finally, we contrast our proposed approach to similar works in the self-supervised domain.

Misalignment in narrated videos. In [Miech et al., 2019b], the authors estimate that around 50% of clip-narration pairs from the HowTo100M dataset are not aligned. In fact, people are likely to describe an event after or before performing it in the video as illustrated in Figure 6-1. This visual misalignment makes it more challenging to learn video representations than with manually annotated and aligned labels.

How to learn despite noisy supervision ? To address the aforementioned issues, we propose to consider multiple options for matching a video and a narration instead of *only* comparing a single video x with a single narration y as done in [Miech et al., 2019b]. Let's consider the example illustrated in Figure 6-2a. Given a clip x, K narrations $\{y_k\}_{k=1}^K$ that happen close in time within the same video can be considered as positive candidates. By doing so, the chance that spoken words correlate with what is happening in the video increases. In that case, we would like to match *at least one* of the narrations $\{y_k\}_{k=1}^K$ with video x. Given the probabilistic model (6.2), a natural way to express this is by computing the joint probability of x happening with any of the y_k. Because we can make the assumption that y_k's are mutually exclusive (*i.e.* $(x, y_i) \neq (x, y_j), \forall i \neq j$), this can be expressed mathematically by (6.2) as follows:

$$p(\cup_k\{(x, y_k)\}) = \sum_k p(x, y_k) \propto \sum_k e^{f(x)^\top g(y_k)}. \tag{6.3}$$

This is a MIL like extension which as opposed to MIL-SVM, do not explicitly select a single positive sample per bag at training. More generally, and symmetrically, the case where several video clips are candidates for a given narration can also be envisioned. Hence, for generality, we assume that instead of having a single pair (x, y), we have a set of candidate positive pairs $\mathcal{P} = (x^k, y^k)_{k=1}^K$, and we can simply repurpose (6.3) as $p(\mathcal{P}) \propto \sum_{(x,y)\in\mathcal{P}} e^{f(x)^\top g(y)}$. We denote by $\{\mathcal{P}_i\}_{i=1}^n$ the training set of candidate positives deduced from the original training set $\{(x_i, y_i)\}_{i=1}^n$. With this extension, we have the tools to address misalignments. Details about how to construct $\mathcal{P}_i$ are given in Section 6.4.1

How to train this model? ***MIL-NCE.*** We wish to learn a video representation based on the previously described probabilistic model $p(\mathcal{P})$. However, this is challenging as one cannot directly apply standard *generative* techniques such as maximum likelihood due to the intractability of computing the normalization constant over all possible pairs of videos and narrations. Instead, we rely on a *discriminative* technique, namely the noise-contrastive estimation (*NCE*) approach [Gutmann and Hyvärinen, 2010; Jozefowicz et al., 2016], that has recently been shown to be effective in the context of feature learning [Hénaff et al., 2019; Oord et al., 2018]. The core idea is to directly optimize the *unnormalized* probabilistic model (6.3) to discriminate between data obtained from the true joint distribution $P(\mathcal{X} \times \mathcal{Y})$ and some artificially generated noise data, a.k.a. "negatives". In this chapter, we use the softmax version of NCE [Jozefowicz et al., 2016]:

$$\max_{f,g} \sum_{i=1}^{n} \log \left(\frac{e^{f(x_i)^\top g(y_i)}}{e^{f(x_i)^\top g(y_i)} + \sum\limits_{(x',y') \sim \mathcal{N}_i} e^{f(x')^\top g(y')}} \right) \tag{6.4}$$

and replacing the probability of a single positive match, $e^{f(x_i)^\top g(y_i)}$, with our MIL like extension, $\sum_{(x,y)\in\mathcal{P}_i} e^{f(x)^\top g(y)}$, gives our proposed *MIL-NCE* training objective (6.1). Given this, we can simply estimate the parameters of our model by maximizing the objective (6.1), where $\mathcal{N}_i$ is a specific set of negatives for the i-th sample. Next, we discuss how our approach differs from prior work.

NCE objectives for self-supervised learning. NCE has recently been successfully applied to self-supervision. In particular, CPC [Hénaff et al., 2019; Oord et al., 2018] introduces the *InfoNCE* loss to enforce the model to maximize the conditional probability of some targets (*e.g.* the bottom part of the image) conditioned on some context (*e.g.* the top part of the image). Differently from CPC, which creates an *asymmetric* set of negatives by fixing the context and only sampling negative targets, we instead use NCE to model the symmetric joint probability between text and video (6.2). Thus, we construct $\mathcal{N}_i$ so that it contains both negatives for video x_i *and* narration y_i. In Section 6.4, we describe precisely how $\mathcal{N}_i$ is obtained as well as evaluate the benefit of this symmetric approach.

6.4 Experiments

We first describe implementation details of our method in Section 6.4.1. The datasets used in our evaluation are outlined in Section 6.4.2. We present an ablation study emphasizing key ingredients of our approach in Section 6.4.3. Finally, we compare our learnt representations to previous self and fully-supervised methods in Section 6.4.4.

Operation	output size
Input video	32×200×200×3
I3D / S3D → Mixed_5c	4×6×6×1024
Global avg pool	1×1×1×1024
Linear	1×1×1×512

Operation	output size
Embedding	16×300
Linear + ReLU	16×2048
Max pool	1×2048
Linear	1×512

Table 6.1: Video (left) and text (right) model architectures.

6.4.1 Implementation details

Model and Inputs. For the 3D CNN backbone, we use the standard I3D implementation from [Carreira and Zisserman, 2017] for all ablation studies and for the comparison to state-of-the-art, we report result on both I3D and S3D [Xie et al., 2018]. We use the Google News self-supervised pre-trained word2vec (d=300) embedding from [Mikolov et al., 2013] for our word representation. Each video clip at training contains 32 frames sampled at 10 fps (3.2 seconds) with a 200x200 resolution (224x224 at test time). For each narration, we take a maximum of 16 words. More details about the model architecture and input dimensions are provided in Table 6.1. A detailed illustration of the architecture is also given in Figure 6-4.

Visual representations evaluation. We evaluate our visual representations at two different semantic levels. First, we use the output of the I3D (or S3D) *Global avg pool* (see Table 6.1), to evaluate our representation for action recognition, action segmentation and action localization. Next, the output of the last I3D (or S3D) *Linear* layer (see Table 6.1), which maps the video to the joint text-video semantic space, is used in conjunction with the output of the language model for the text-video retrieval tasks.

Training dataset. We train our model using the HowTo100M [Miech et al., 2019b] narrated video dataset. It consists of more than 1.2M videos accompanied with automatically generated speech transcription. We use the provided transcription to create pairs of video / caption defined by each caption time stamp. Note that while the original dataset [Miech et al., 2019b] consists of 136M pairs, we only used 120M pairs to comply with the YouTube wipe out policy. Each video shorter than 5 seconds is extended symmetrically in time so that the duration is at least 5 seconds.

Then we randomly sample, a fixed length clip of 3.2 seconds within each video at training. For each clip-narration training pair (x, y) sampled, we construct the bag of positive candidate pairs $\mathcal{P}$ by considering the nearest captions in time to y as depicted in Figure 6-2a. For example, if we set the number of positive candidate pairs to 3, we would have $\mathcal{P} = \{(x, y), (x, y^1), (x, y^2)\}$ where y^1 and y^2 are the 2 closest narrations in time to y. We work with batch containing B positive video-narration pairs $\{(x_i, y_i)\}_{i\in[1,B]}$. We construct the set $\mathcal{N}$ by simply creating negative pairs from this batch by combining $\{(x_i, y_j)\}_{i\neq j}$. Since all representations are already computed, computing negative scores is cheap and efficient.

Optimization. We use the ADAM [Kingma and Ba, 2015] optimizer with an initial learning rate of 10^{-3} with linear warm up of 5k steps. The learning rate is decayed twice by a factor of 10. We train our model using Cloud TPUs v3 [1], each Cloud TPU having a batch size of 128 videos. Given the high computational load required for training on HowTo100M, we run ablation studies on 4 Cloud TPUs and train our model for 500k steps ($\sim$ 3 days). For our final evaluation in Section 6.4.4, we pick the best parameters based on our ablation study and then use 64 Cloud TPUs for 400k steps (also $\sim$ 3 days) as we observed that training on bigger batch size, and thus more epochs, had a positive impact on performance.

6.4.2 Downstream tasks

To show the generality of our learnt representations, we perform evaluation on five diverse downstream tasks using eight datasets described below.

Action Recognition: *HMDB-51 [Kuehne et al., 2011], UCF-101 [Soomro et al., 2012], Kinetics-700 [Carreira et al., 2019].* We evaluate our video-only representation on the traditional HMDB-51 / UCF-101 as well as the recent Kinetics-700 action recognition tasks.

Text-to-Video retrieval: *YouCook2 [Zhou et al., 2018b], MSR-VTT [Xu et al., 2016].* We use the YouCook2 and MSR-VTT text-to-video retrieval benchmarks to evaluate our off-the-shelf learnt joint text-video representation. We follow the same

[1] `https://cloud.google.com/tpu/`

evaluation protocol as described in [Miech et al., 2019b]. We report the retrieval performance using the recall at K (R@K) metric (with K=1,5,10) which measures the percentage of clips retrieved at the top K (the higher the better). We also report the median rank (MedR) of videos to be retrieved (the lower the better). Note from [Miech et al., 2019b] that there is no intersection between YouCook2 testing and HowTo100M training videos.

Action Localization: *YouTube-8M [Abu-El-Haija et al., 2016] Segments.* We evaluate our video representation on YouTube-8M Segments[2], a subset of the YouTube-8M [Abu-El-Haija et al., 2016] with precise temporal annotation. We follow the YouTube-8M Segments challenge evaluation protocol and report the mAP metric.[3]

Action Step Localization: *CrossTask [Zhukov et al., 2019].* We use the recently released CrossTask instructional video dataset to evaluate our off-the-shelf learnt joint text-video representation on the task of action step localization. We perform the same evaluation protocol as in [Zhukov et al., 2019] and report the average recall (CTR) metric for the localization task.

Action Segmentation: *COIN [Tang et al., 2019].* We evaluate our video-only representation on the COIN action segmentation task and follow the evaluation protocol of [Sun et al., 2019a] by reporting the frame-wise accuracy (FA).

6.4.3 Ablation studies

We perform the ablation studies on the following downstream tasks: MSR-VTT R@10 (MR10), YouCook2 R@10 (YR10), HMDB-51 and UCF-101 recognition accuracy on split 1 and CrossTask average recall (CTR). This subset of downstream tasks has been chosen for their simplicity of evaluation and because they cover a wide range of tasks.

Which loss is better for learning the joint embedding ?

In this ablation study (Table 6.2a), we compare different losses for matching the text and video embeddings in the standard single-instance learning setting where we pair

[2]`https://research.google.com/youtube8m`

[3]`https://www.kaggle.com/c/youtube8m-2019/overview/evaluation`

(a) **Training loss**

Loss	YR10	MR10	CTR	HMDB	UCF
Binary-Classif	18.5	23.1	32.6	44.2	68.5
Max margin	16.3	24.1	29.3	**56.2**	76.6
NCE	**29.1**	**27.0**	**35.6**	55.4	**77.5**

(b) **Negatives per positive**

$\|\mathcal{N}\|$	YR10	MR10	CTR	HMDB	UCF
64	26.0	25.5	33.1	56.1	76.0
128	27.1	26.4	33.3	**57.2**	76.2
256	28.7	28.7	**36.5**	56.5	**77.5**
512	**28.8**	**29.0**	35.6	55.4	77.4

(c) **Number of positive pair**

	NCE	MIL-NCE				
$\|\mathcal{P}\| \rightarrow$	1	3	5	9	17	33
YR10	29.1	33.6	**35.0**	33.1	32.4	28.3
MR10	27.0	30.2	**31.8**	30.5	29.2	30.4
CTR	35.6	**37.3**	34.2	31.8	25.0	25.0
HMDB	55.4	**57.8**	56.7	55.7	54.8	51.4
UCF	77.5	79.7	**80.4**	79.5	78.5	77.9

(d) **MIL strategy**

Method	YR10	MR10	CTR	HMDB	UCF
Cat+NCE	31.9	30.8	**35.2**	56.3	78.9
Max+NCE	32.3	31.3	32.2	55.3	79.2
Attn+NCE	32.4	30.2	33.4	55.2	78.4
MIL-NCE	**35.0**	**31.8**	34.2	**56.7**	**80.4**

(e) **Symmetric vs asymmetric negatives**

Negatives	YR10	MR10	CTR	HMDB	UCF
$(x\|y)$	34.4	29.0	33.9	55.1	78.1
$(y\|x)$	19.3	19.4	28.2	**57.1**	79.2
(x, y)	**35.0**	**31.8**	**34.2**	56.7	**80.4**

(f) **Language models**

Text model	YR10	MR10	CTR	HMDB	UCF
LSTM	16.6	15.6	23.8	53.1	80.1
GRU	16.8	16.9	22.2	54.7	**82.8**
Transformer	26.7	26.5	32.7	53.4	78.4
NetVLAD	33.4	29.2	**35.5**	51.8	79.3
Ours	**35.0**	**31.8**	34.2	**56.7**	80.4

Table 6.2: Ablation studies

each video clip to its closest narration in time. We compare the NCE based approach (ours) to the frequently used max margin ranking loss [Chowdhury et al., 2018; Dong et al., 2019; Hendricks et al., 2017; Karpathy et al., 2014a; Miech et al., 2018; Mithun et al., 2018; Wang et al., 2018a, 2016b; Wray et al., 2019; Wu et al., 2017] and a binary classification loss (*i.e.* sigmoid cross entropy loss) that has shown to be effective in video-audio matching [Arandjelović and Zisserman, 2017, 2018]. Overall, the NCE loss outperforms other losses or works similarly on all five tested datasets.

The more negatives, the better. We keep the same single-instance learning setting and assess the quality of our representations trained with different number of sampled negative examples per positive pair in Table 6.2b. We can see that the overall performance increases with the number of negatives. For the rest of the ablation studies, we use 512 negative samples per positive.

How many positive candidates pairs to consider ? We evaluate the benefit of going from the single-instance learning approach to the proposed multiple-instance

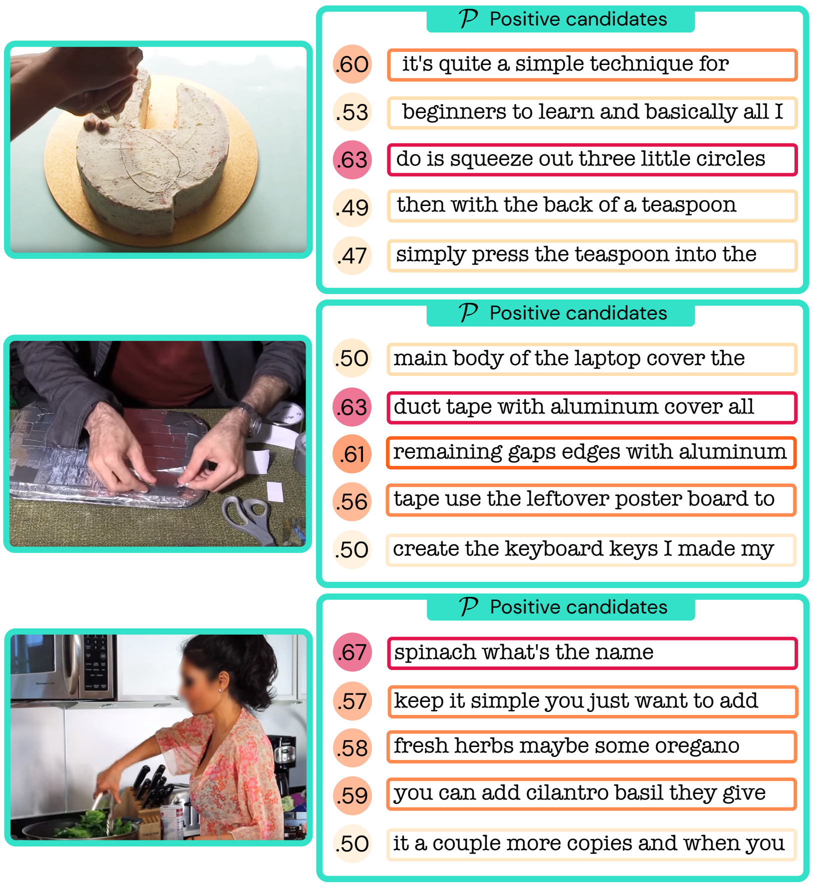

$\mathcal{P}$ Positive candidates
.60 it's quite a simple technique for
.53 beginners to learn and basically all I
.63 do is squeeze out three little circles
.49 then with the back of a teaspoon
.47 simply press the teaspoon into the
$\mathcal{P}$ Positive candidates
.50 main body of the laptop cover the
.63 duct tape with aluminum cover all
.61 remaining gaps edges with aluminum
.56 tape use the leftover poster board to
.50 create the keyboard keys I made my
$\mathcal{P}$ Positive candidates
.67 spinach what's the name
.57 keep it simple you just want to add
.58 fresh herbs maybe some oregano
.59 you can add cilantro basil they give
.50 it a couple more copies and when you

Figure 6-3: Selected video and narration pairs from five positive candidates on HowTo100M held-out samples using MIL-NCE..

based approach in Table 6.2c. In this experiment, we vary the number of positive candidate training pairs $\|\mathcal{P}\|$ for each video clip from 1 (*i.e.* single-instance learning setting) up to 33 candidates. Adding candidates significantly improves the performance upon the single-instance learning baseline. Moreover, we observe a trade-off between having too many candidates and not having enough of them, as we reach the best results by considering 3 to 5 positive candidates. We believe that adding too many contextual narrations increases the chance for irrelevant ones as they are sampled further in time from the considered video clip. For the rest of the chapter we fix the number of positive candidate pairs to 5.

MIL-NCE vs other MIL based approaches. In Table 6.2d, we compare our MIL-NCE approach with methods that can also handle multiple possible candidate captions at training time. The max-pool based approach [Andrews et al., 2003; Arandjelović and Zisserman, 2018; Oquab et al., 2015] (Max+NCE) only optimizes over the clip-caption pair with the highest similarity score among the positive candidates. On the other hand, the attention-based approach [Ilse et al., 2018] (Attn+NCE) computes cross-modal attention weights between all the clip-caption pairs and perform a weighted average of the similarity scores in order to consider the most relevant positive candidate pairs. More details about these baselines are provided in section 6.4.5. Finally, we also compare to the single-instance learning baseline where we concatenate all of the candidate narrations as one longer narration (Cat+NCE). Our proposed MIL-NCE method outperforms these two standard approaches on five out of six tasks. The qualitative figure from Figure 6-3 illustrates examples of selected pairs from a hold-out set of HowTo100M videos, using MIL-NCE.

Symmetric or asymmetric negative sampling ? Recall that given a pair of video/narration (x, y), we create $\mathcal{N}$ in a *symmetric* manner by sampling negative narrations for the video x *and* negative videos for the narration y. Table 6.2e compares that approach (x, y) to *asymmetric* alternatives: (i) by fixing the video x and *only* sampling negative captions $(y|x)$ and (ii) by fixing the narration y and *only* sampling negative videos $(x|y)$. Overall, the best results are achieved when sampling *jointly* the negatives (x, y), *i.e.* when we equally sample both video and narration negatives.

Which language model ? Finally, we also experiment with different language models (1 layer LSTM [Hochreiter and Schmidhuber, 1997] or GRU [Cho et al., 2014], 1 layer and 8 attention heads Transformer [Vaswani et al., 2017] and NetVLAD with 32 clusters [Arandjelović et al., 2016]) and compare them to our simple model (see Table 6.1) in Table 6.2f. Even though our language model is similar to simple bag-of-word approach, it performs better on average and is more consistent over the five tasks than the other models. In particular, our model significantly outperforms the other language models on the text-to-video retrieval tasks (YR10 and MR10), where language plays the most important role. We believe that a sophisticated language understanding is not key for our learning task. Instead, most of the time, detecting and matching the main keywords in each narration is enough.

Moreover, we provide an ablation study where we replace the input text vectors

Input word vectors	YR10	MR10
BERT wo. stop words	25.1	24.2
BERT w. stop words	24.2	26.0
Word2Vec	**35.0**	**31.8**

Table 6.3: Results when using BERT vectors as inputs instead of Word2vec.

coming from Word2Vec with more advanced contextual word embedding vectors from a BERT model [Devlin et al., 2018]. In details, we use the BERT base model [4] to replace the Word2Vec module in our model (see Figure 6-4): we process the sequence of 16 input words to obtain 16 output vectors of dimension 768. Apart from the fact that the input dimension of word vectors is increased from 300 to 768, the rest of the text model is kept the same. For a fair comparison in terms of number of parameters, we do not finetune the BERT model. For this experiment, we use an I3D model under the same training setting used in the ablation study 6.4.3. Also, as BERT may be sensitive to the absence of stop words, we also run an experiment where we do not remove the stop words during the text preprocessing phase. Results are given in Table 6.3. We observe a strong degradation in performance when using BERT pretrained vectors. We believe this is due to the domain gap between web text corpus and ASR outputs.

6.4.4 Comparison to the state-of-the-art

Video only representation. In Table 6.4, we evaluate our learnt representations on the HMDB-51 [Kuehne et al., 2011] and UCF-101 [Soomro et al., 2012] action recognition benchmarks by extracting averaged pooled Mixed_5c features from the HowTo100M pretrained backbone. More specifically, we compare to self-supervised approaches, which similarly to this chapter, do not make use of any annotated video nor image dataset when training the visual representation. We outperform state-of-the-art on UCF-101 and perform on par with AVTS [Korbar et al., 2018] on HMDB-51. Most importantly, our learnt representation significantly outperforms many prior

[4]`https://tfhub.dev/google/bert_cased_L-12_H-768_A-12/1`

Method	Dataset	MM	Model	Frozen	HMDB	UCF
OPN [Lee et al., 2017]	UCF	✗	VGG	✗	23.8	59.6
Shuffle & Learn [Misra et al., 2016]*	K600	✗	S3D	✗	35.8	68.7
[Wang et al., 2019a]	K400	Flow	C3D	✗	33.4	61.2
CMC [Tian et al., 2019]	UCF	Flow	CaffeNet	✗	26.7	59.1
Geometry [Gan et al., 2019]	FC	Flow	FlowNet	✗	23.3	55.1
[Fernando et al., 2017]	UCF	✗	AlexNet	✗	32.5	60.3
ClipOrder [Xu et al., 2019]	UCF	✗	R(2+1)D	✗	30.9	72.4
3DRotNet [Jing and Tian, 2018]*	K600	✗	S3D	✗	40.0	75.3
DPC [Han et al., 2019]	K400	✗	3D-R34	✗	35.7	75.7
CBT [Sun et al., 2019a]	K600	✗	S3D	✓	29.5	54.0
CBT [Sun et al., 2019a]	K600	✗	S3D	✗	44.6	79.5
AVTS [Korbar et al., 2018]	K600	Audio	I3D	✗	53.0	83.7
AVTS [Korbar et al., 2018]	Audioset	Audio	MC3	✗	**61.6**	89.0
Ours	HTM	Text	I3D	✓	**54.8**	**83.4**
				✗	59.2	**89.1**
			S3D	✓	**53.1**	**82.7**
				✗	61.0	**91.3**
Fully-supervised SOTA [Xie et al., 2018]			S3D	✗	75.9	96.8

Table 6.4: **Comparison to self-supervised methods on HMDB/UCF**. Results are reported by averaging the accuracy over the 3 splits for both datasets. *Shuffle & Learn and 3DRotNet reported numbers are reimplemented in [Sun et al., 2019a] by using a better backbone (S3D). The *MM* column indicates whether or not other modalities than the video frames have been used for the learning of the visual features. FC: FlyingChairs.

approaches even without fine-tuning. This result is significant as it demonstrates the generalization of our representation to diverse sets of actions despite being trained on uncurated instructional videos.

Next, we evaluate our visual representation on COIN [Tang et al., 2019] action segmentation task in Table 6.5a. We split videos in subsequent clips of 1.5 second and represent them by concatenating three features: the local representation from I3D (or S3D), the global average pooled representation across the entire video and the relative positional embedding of the video clip. We train a logistic regression to predict the label for each clip. We compare our HowTo100M pretrained I3D network to an I3D fully-supervised on Kinetics-400, Kinetics-700 as well as a ResNet-50 fully supervised

Method	Net	Pretraining Dataset	Pretraining Labels	FA
	R50	ImNet	✓	52.0
Ours	I3D	K400	✓	52.9
	I3D	K700	✓	54.2
CBT [Sun et al., 2019a]	S3D	K600+HTM	✓	53.9
Ours	I3D	HTM	✗	**59.4**
Ours	S3D	HTM	✗	**61.0**

(a) **COIN**

Net	Pretraining Dataset	Pretraining Labels	mAP
I3D	K400	✓	73.7
I3D	K700	✓	74.0
R50	ImNet	✓	75.0
I3D	HTM	✗	**77.1**

(b) **YT8M-S**

Method	Labels used	CTR
[Alayrac et al., 2016]	ImNet+K400	13.3
CrossTask [Zhukov et al., 2019]	ImNet+K400	22.4
CrossTask [Zhukov et al., 2019]	ImNet+K400+CT	31.6
[Miech et al., 2019b]	ImNet+K400	33.6
Ours (I3D)	**None**	**36.4**
Ours (S3D)	**None**	**40.5**

(c) **CrossTask** (CT)

Init	Net	Top1 val	Top1 test
Scratch	I3D	57.0	55.4
ImNet	I3D	59.9	58.2
Ours	I3D	**61.1**	**59.6**

(d) **K700**

Table 6.5: Evaluation on action segmentation (a), localization (b, c) and recognition (d) benchmarks. K400: Kinetics-400, K600: Kinetics-600, K700: Kinetics-700, HTM: HowTo100M, ImNet: ImageNet, YT8M-S: YouTube-8M Segments, R50: 2D ResNet-50.

on ImageNet. We also compare to the state-of-the-art approach on COIN, CBT [Sun et al., 2019a], which relies on a fully supervised S3D [Xie et al., 2018] trained on Kinetics-600. Our learnt representation performs better than representations trained on Kinetics-400, Kinetics-700 or ImageNet. Moreover, our method also significantly outperforms the state-of-the-art CBT [Sun et al., 2019a] despite their use of fully-supervised representation trained on Kinetics-600 and a Transformer model. We believe this improved localization ability is due to the fact that our model was trained on narrative descriptions with precise timestamps as opposed to coarse video-level annotations.

We also report performance on the recently released YouTube-8M Segments dataset in Table 6.5b. Since no results have been published for this benchmark yet, we only compare the classification performance using different fully-supervised representations (*i.e.* I3D trained on Kinetics-400 / Kinetics-700 or ResNet-50 trained on ImageNet). Here again, our learnt representation outperforms all of the fully-supervised counterparts despite the domain gap between YouTube-8M and uncurated instructional

Method	Labeled dataset used	R@1↑	R@5↑	R@10↑	MedR↓
Random	None	0.03	0.15	0.3	1675
HGLMM FV CCA [Klein et al., 2015]	ImNet + K400 + YouCook2	4.6	14.3	21.6	75
[Miech et al., 2019b]	ImNet + K400	6.1	17.3	24.8	46
[Miech et al., 2019b]	ImNet + K400 + YouCook2	8.2	24.5	35.3	24
Ours (I3D)	**None**	**11.4**	**30.6**	**42.0**	**16**
Ours (S3D)	**None**	**15.1**	**38.0**	**51.2**	**10**

(a) **YouCook2**

Method	Labeled dataset used	R@1↑	R@5↑	R@10↑	MedR↓
Random	None	0.01	0.05	0.1	500
[Miech et al., 2019b]	ImNet + K400	7.5	21.2	29.6	38
Ours (I3D)	**None**	**9.4**	**22.2**	**30.0**	**35**
Ours (S3D)	**None**	**9.9**	**24.0**	**32.4**	**29.5**

(b) **MSRVTT**

Table 6.6: Zero-shot evaluation on text-to-video retrieval.

videos.

Finally, in Table 6.5d we investigate the benefit of initializing an I3D model with our learned weights for a large-scale action recognition dataset (*i.e.* Kinetics-700). We compare to a randomly initialized I3D and one inflated from an Inception network pretrained on ImageNet [Carreira and Zisserman, 2017]. We obtain a 4% improvement over the randomly initialized I3D and 1.4% over the ImageNet pretrained I3D [Carreira and Zisserman, 2017].

Joint text-video representation. We report text-to-video retrieval results on the YouCook2 (Table 6.6a) and MSR-VTT (Table 6.6b) using our off-the-shelf model trained on HowTo100M. Note that our model has not seen any YouCook2 nor MSR-VTT annotated videos, hence for fair comparison on the MSR-VTT dataset we only compare to prior work [Miech et al., 2019b] that did not finetune on MSR-VTT. On YouCook2, our model significantly outperforms all prior work. More specifically, it performs better than [Miech et al., 2019b] which uses visual representation trained on Kinetics-400 + ImageNet and trains the joint text-video representation on both HowTo100M and *YouCook2*. On MSR-VTT, our method performs also better than [Miech et al., 2019b], yet without using any manually annotated dataset. Finally, we also evaluate our off-the-shelf model trained on HowTo100M on the CrossTask [Zhukov et al., 2019] action localization benchmark in Table 6.5c. We perform localization via a video-to-text retrieval approach similarly to [Miech et al., 2019b]. Our method outperforms state-of-the-art approaches on this benchmark, here again, without using manual supervision.

6.4.5 Max+NCE and Attn+NCE baselines

We provide here, more details formulations of the Max+NCE and Attn+NCE baselines from the ablation study.

Max+NCE. This baseline aims at reproducing the standard max-pool based approach often used in multiple instance learning, but here combined with the NCE loss. Formally, this can be written as maximizing the following objective:

$$\max_{f,g} \sum_{i=1}^{n} \log\left(\mathrm{MaxNCE}_i\right), \tag{6.5}$$

where:

$$\mathrm{MaxNCE}_i = \frac{\max\limits_{(x,y)\in\mathcal{P}_i} e^{f(x)^\top g(y)}}{\max\limits_{(x,y)\in\mathcal{P}_i} e^{f(x)^\top g(y)} + \sum\limits_{(x',y')\sim\mathcal{N}_i} e^{f(x')^\top g(y')}}. \tag{6.6}$$

Intuitively, this corresponds to choosing the *best* positive candidate pair among all pairs $\mathcal{P}_i$ according to the model.

Attn+NCE. This other baseline aims at selecting best candidate pairs via a cross-modal soft-attention mechanism between the clips and narrations. The cross-modal attention mechanism a is defined as follows:

$$a(x,y,\mathcal{P}_i) = \frac{e^{f_a(x)^\top g_a(y)}}{\sum\limits_{(x',y')\in\mathcal{P}_i} e^{f_a(x')^\top g_a(y')}}, \tag{6.7}$$

where f_a and g_a are two parametrized functions. In practice f_a and g_a are sharing parameters with f (respectively g) except for the last 'Linear' layer (see Figure 6-4).

Given that cross-modal attention mechansim, the Attn+NCE objective is:

$$\max_{f,g,a} \sum_{i=1}^{n} \log\left(\text{AttnNCE}_i\right), \tag{6.8}$$

where:

$$\text{AttnNCE}_i = \frac{e^{\sum\limits_{(x,y)\in\mathcal{P}_i} a(x,y,\mathcal{P}_i)\ f(x)^\top g(y)}}{e^{\sum\limits_{(x,y)\in\mathcal{P}_i} a(x,y,\mathcal{P}_i)\ f(x)^\top g(y)} + \sum\limits_{(x',y')\sim\mathcal{N}_i} e^{f(x')^\top g(y')}}. \tag{6.9}$$

The intuition behind this approach is to allow the model to have a separate selection mechanism for the positive candidate pairs.

6.5 Conclusion

In this final contribution chapter, we have addressed the challenging task of learning visual representations from scratch using uncurated instructional videos. Our approach *did not rely on any manually annotated video nor image dataset.* To deal with highly misaligned narrations and videos, we have introduced MIL-NCE, a multiple instance learning approach derived from the noise contrastive estimation framework. We have applied MIL-NCE to the uncurated HowTo100M dataset and obtained strong visual representations that outperformed self-supervised as well as fully-supervised representations on many downstream tasks. More generally, we believe MIL-NCE can be applicable in many multiple instance learning problems where representation learning is key.

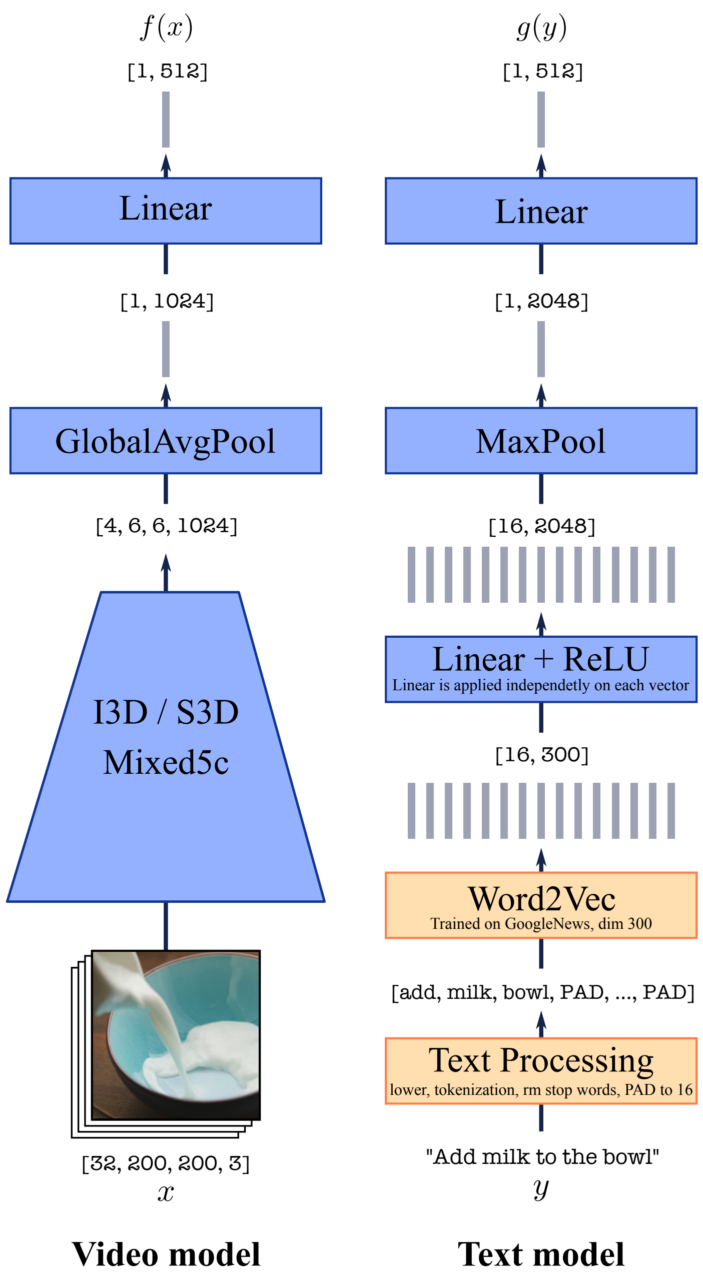


Figure 6-4: **Model architecture**. In this figure, we provide a visualization of the video embedding network f (**left**) and the text embedding network g (**right**). Modules displayed in blue are trained *from scratch* on the challenging uncurated HowTo100M dataset using the MIL-NCE loss. The word embeddings are learned in an unsupervised fashion using Word2Vec trained on GoogleNews and are kept fixed during training. Finally, the dimensions of the *outputs* of each layer are specified in brackets, *e.g.* the output of the 'Word2Vec' layer is of size $[16, 300]$ corresponding to the 16 word embedding vectors of dimension 300 (one vector for each word, also materialized by the 16 grey rectangles).

Chapter 7

Discussion and perspectives

In this last chapter, we summarize the thesis contributions before concluding with perspectives on future research directions.

7.1 Summary of contributions

Learnable Pooling with Context Gating for Video Understanding. In Chapter 3, we tackled the problem of learning a video feature aggregation module that can reliably aggregate information from videos that are several minutes long. We proposed a neural network architecture that combines a differentiable clustering-based layer for aggregating frame-level video features with a novel learnable non-linear unit, named Context Gating, aiming to model interdependencies among network activations. We have applied our approach on the large-scale and weakly-labeled YouTube-8M video classification dataset [Abu-El-Haija et al., 2016], which contains millions of minut-long videos annotated with weak labels generated from metadata. Our approach especially reached the first place at the YouTube-8M challenge [1] among 655 participating teams, which demonstrated the efficiency of our approach on long videos and weak labels.

[1]`https://www.kaggle.com/c/youtube8m`

Learning from Video and Text via Large-Scale Discriminative Clustering. In Chapter 4, we studied large-scale weakly-supervised learning from readily available supervision in the form of movie scripts for actor and action recognition. In particular, we followed a multiple-instance learning approach based on discriminative clustering, which has been previously successfully applied to a number of weakly-supervised learning tasks in video understanding. One drawback of discriminative clustering, however, is its limited scalability at training. We have addressed this issue by proposing an online optimization algorithm based on the Block-Coordinate Frank-Wolfe algorithm [Lacoste-Julien et al., 2013]. Another contribution of the chapter was the introduction of a new constraint, namely the background constraint, aiming to model the background class within the discriminative cluster formulation in an unsupervised manner. We have applied our multiple instance learning method to the problem of weakly-supervised learning of actions and actors from movies together with their corresponding movie scripts. The scaling up of the learning problem to 66 feature-length movies enabled us to significantly improve weakly-supervised action recognition performance compared to prior work [Bojanowski et al., 2013].

HowTo100M: Learning a Text-Video Embedding from Uncurated Narrated Videos. In Chapter 5, we have introduced a method for learning a joint video and text embedding without using manually annotated videos with descriptions. In particular, we have proposed to learn a joint video and text embedding from uncurated videos with readily available natural language annotations in the form of automatically transcribed narrations. We have introduced *HowTo100M*: a large-scale uncurated dataset of 136 million video clips sourced from 1.22 million narrated instructional web videos depicting humans performing and describing over 23k different visual tasks. Our data collection procedure is fast, scalable and does not require any additional manual annotation. We have demonstrated that a joint video and text embedding trained on this data leads to state-of-the-art results for text-to-video retrieval and action localization on instructional video datasets such as YouCook2 or CrossTask. Finally, we have shown that this embedding transfers well to other domains: fine-tuning on generic YouTube videos (MSR-VTT dataset) and movies (LSMDC dataset) outperforms models trained on these datasets alone.

End-to-End Learning of Visual Representations from Uncurated Instructional Videos. Finally in Chapter 6, we have proposed an approach for learning a video representation *from scratch* by only using uncurated narrated videos from the HowTo100M dataset introduced in the preceding Chapter 5. We have introduced a novel training objective combining Multiple Instance Learning with Contrastive Learning, namely *MIL-NCE*, for addressing weak correspondence between the video and the transcribed narration. With this approach, we have been able to learn strong video representations from scratch, without the need for any manual annotation. We have evaluated our representations on a wide range of four downstream tasks over eight datasets: action recognition (HMDB-51, UCF-101, Kinetics-700), text-to-video retrieval (YouCook2, MSR-VTT), action localization (YouTube-8M Segments, CrossTask) and action segmentation (COIN). Our method has outperformed all published self-supervised approaches for these tasks as well as several fully supervised baselines.

7.2 Perspectives

Next, we provide several future research directions that stem from this thesis.

7.2.1 Leveraging narrated videos

We have shown in this thesis that by leveraging readily available data in the form of natural language, it is possible to outperform video models trained in a fully supervised manner with videos manually annotated for various tasks (see Chapter 5 and Chapter 6). However, there are still many domains and tasks where weakly-supervised video models perform far below fully-supervised ones. Examples include tasks such as human action recognition on Kinetics or AVA for action detection.

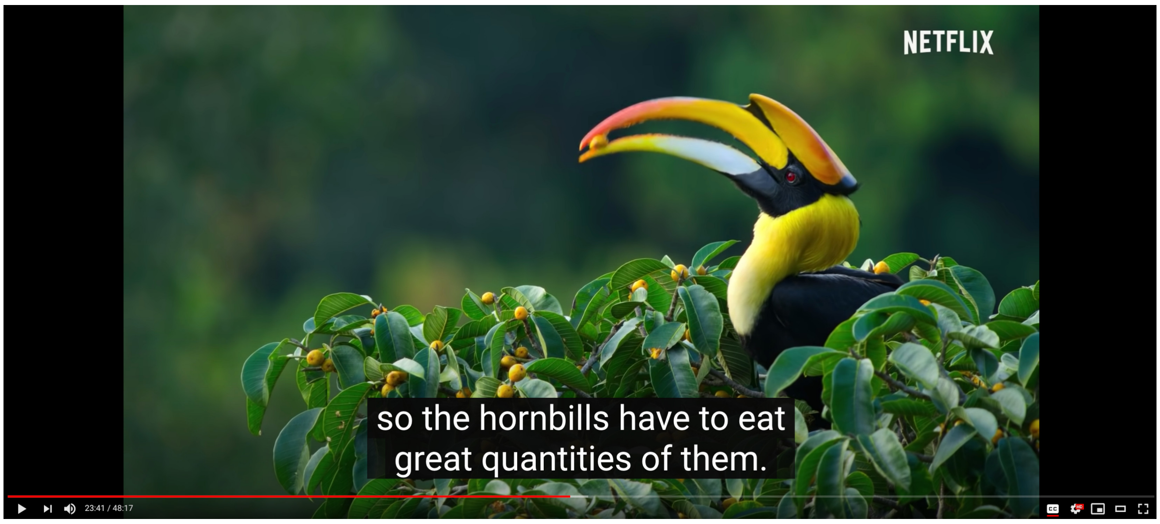


Figure 7-1: An example of wildlife documentary video together with the narration transcript. YouTube video: `https://www.youtube.com/watch?v=JkaxUblCGz0`

One potential future research direction is to keep pursuing our initial research efforts on leveraging narrated videos. Especially, we believe leveraging narrated videos is one of the most promising directions in self-supervised learning given our initial work from Chapter 5 and Chapter 6 using the HowTo100M dataset. However, one drawback with the HowTo100M dataset is the bias towards instructional videos which makes it inadequate for many other video domains. As an example, we have noticed that models trained on the HowTo100M dataset are quite weak at recognizing different types of wild animals such as lions or elephants, which are rarely seen in instructional videos. In contrast, these animals are easily recognizable by CNNs trained on ImageNet. We believe this issue could be limited if our training set is enhanced with narrated wildlife documentary videos. In fact, many of them contain wildlife documentaries with accurate narrated descriptions of animals [Chen et al., 2017a] (See Figure 7-1 for an illustration). This is only an example but in general, there are other interesting video domains that have available and useful narration. This opens up the general problem of learning from narrated videos in different domains. Can we find an automatic approach for mining useful narrated videos from the Internet? For instance, we can instead use machine learning to automatically select useful narrated videos, *i.e.* where the narration describes the visual content depicted in the video.

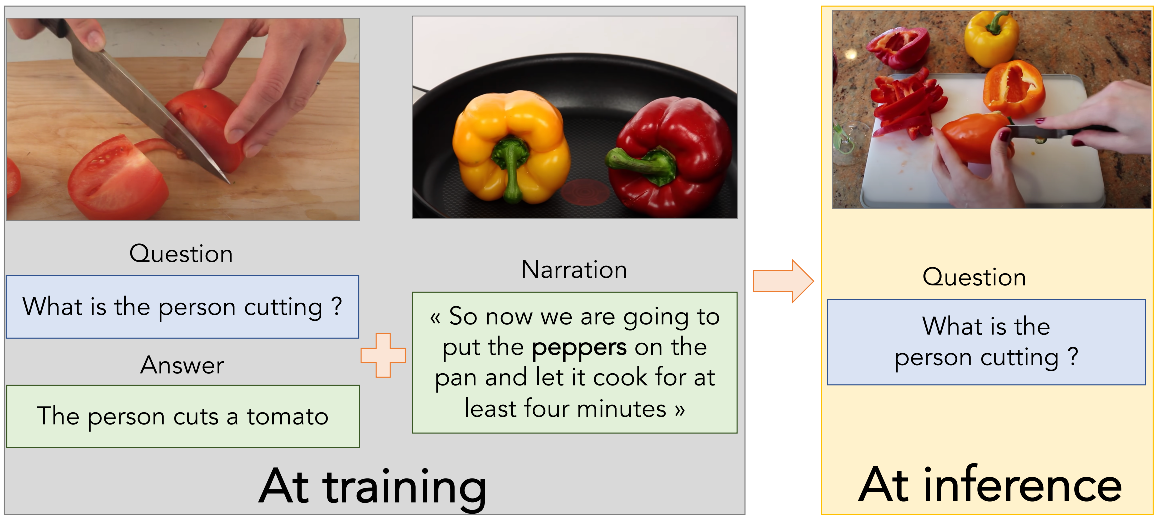

Question
What is the person cutting ?
Answer
The person cuts a tomato
Narration
« So now we are going to put the **peppers** on the pan and let it cook for at least four minutes »
At training
Question
What is the person cutting ?
At inference

Figure 7-2: One future direction to explore is the ability to also leverage narrated videos for improving video question answering. Can we design a model able to answer the question on the video of someone cutting a pepper (right) only by acquiring the knowledge of *pepper* from narrated videos ?

Another issue is that end-to-end training on narrated videos requires a large computational resources. For example, our best model from Chapter 6 was trained using 64 Cloud TPUs V3 (which is roughly equivalent to a cluster of 256 Tesla V100 32Gb GPUs) for 3 days. This is problematic as it means it is challenging to reproduce such results in an academic setup. The high computational load is partly due to the fact that (as estimated in Chapter 5) more than 50M of video clips from HowTo100M are not corresponding to their narration. Thus, another research direction would be to design a learning approach that can be efficiently trained with fewer resources on a large amount of narrated videos despite the noise in the training data.

Finally, in our current work on learning from narrated videos, we did not exploit the spatio-temporal localization of objects/actions in video. It would be exciting to study whether or not we can also leverage this weak form of supervision to train an open-vocabulary object detection model for video.

7.2.2 Tackling more complex video and language tasks

In this thesis, we have explored leveraging readily available natural language data for simple tasks such as video classification or retrieval. However, we did not explore using such readily available data for tasks requiring more complex language modeling such as video question answering, long video summarization, vision and language based navigation or building visual agents that can engage in a discussion about a video or an image (visual dialog). Studying how we could use readily available language data for these more challenging video and language tasks is thus a natural future direction. More especially, given the recent breakthroughs in natural language processing with self-supervised pretraining using Transformer [Vaswani et al., 2017] based methods such as BERT [Devlin et al., 2018], we now have powerful tools for tackling language-based tasks such as question answering or agents that can dialog as humans.

For example, the video question answering field suffers from a lack of manually annotated and large-scale video question answering datasets as underlined in Section 2.2.5. One possible extension of our initial work from Chapter 5 and Chapter 6 could be to explore leveraging narrated videos for improving video question answering models. In fact, this intuitively makes sense as humans are capable of acquiring new knowledge from a narrated video and answering questions about it without requiring supervision in the form of a video with a set of questions and answers, as illustrated in Figure 7-2. The same idea in the field of visual dialog would also be as much as interesting. It would be exciting to show it is possible to leverage millions of unlabeled narrated videos to improve visual dialog systems trained on a few annotated samples. This would be a step towards closing the gap between humans and machines as we are capable of learning and accumulating a large wealth of knowledge only by watching others perform different tasks and talking about it.

Bibliography

Project webpage. `https://www.di.ens.fr/willow/research/howto100m/`, 2019. 114

Nayyer Aafaq, Ajmal Mian, Wei Liu, Syed Zulqarnain Gilani, and Mubarak Shah. Video description: A survey of methods, datasets, and evaluation metrics. *ACM Computing Surveys (CSUR)*, 52(6):1–37, 2019. 27, 29

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2015. 58, 60

Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. YouTube-8M: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 7, 8, 12, 13, 25, 26, 41, 42, 45, 47, 48, 50, 55, 58, 60, 69, 123, 132, 143

Jean-Baptiste Alayrac, Piotr Bojanowski, Nishant Agrawal, Ivan Laptev, Josef Sivic, and Simon Lacoste-Julien. Unsupervised learning from narrated instruction videos. In *CVPR*, 2016. 32, 37, 38, 72, 74, 75, 91, 94, 95, 96, 97, 108, 109, 124, 138

Jean-Baptiste Alayrac, Josef Sivic, Ivan Laptev, and Simon Lacoste-Julien. Joint discovery of object states and manipulation actions. In *ICCV*, 2017. 96

Humam Alwassel, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *arXiv preprint arXiv:1911.12667*, 2019. 37

Elad Amrani, Rami Ben-Ari, Tal Hakim, and Alex Bronstein. Toward self-supervised object detection in unlabeled videos. *arXiv preprint arXiv:1905.11137*, 2019. 124

Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2003. 125, 134

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 34, 35

Relja Arandjelović and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017. 37, 133

Relja Arandjelović and Andrew Zisserman. Objects that sound. In *ECCV*, 2018. 123, 133, 134

Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*, 2016. 26, 46, 49, 55, 56, 60, 135

Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *CVPR*, 2009. 125

Moez Baccouche, Franck Mamalet, Christian Wolf, Christohe Garcia, and Atilla Baskurt. Sequential deep learning for human action recognition. *Human Behavior Understanding*, pages 29–39, 2011. 23, 45, 47, 48

Francis Bach and Zaïd Harchaoui. DIFFRAC: a discriminative and flexible framework for clustering. In *NIPS*, 2007. 9, 72, 73, 74, 75, 76, 78, 125

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005. 28

Fernando Basura, Efstratios Gavves, José M. Oramas, Amir Ghodrati, and Tinne Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015. 26, 48

Tamara L Berg, Alexander C Berg, Jaety Edwards, Michael Maire, Ryan White, Yee-Whye Teh, Erik Learned-Miller, and David A Forsyth. Names and faces in the news. In *CVPR*, volume 2, pages II–848, 2004. 70

Angie Boggust, Kartik Audhkhasi, Dhiraj Joshi, David Harwath, Samuel Thomas, Rogerio Feris, Dan Gutfreund, Yang Zhang, Antonio Torralba, Michael Picheny, et al. Grounding spoken words in unlabeled video. In *CVPRW*, 2019. 124

Piotr Bojanowski, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Finding Actors and Actions in Movies. In *ICCV*, 2013. 39, 70, 72, 73, 74, 75, 80, 86, 87, 88, 89, 125, 144

Piotr Bojanowski, Rémi Lajugie, Francis Bach, Ivan Laptev, Jean Ponce, Cordelia Schmid, and Josef Sivic. Weakly supervised action labeling in videos under ordering constraints. In *ECCV*, 2014. 39, 70, 72, 74, 75

Piotr Bojanowski, Rémi Lajugie, Edouard Grave, Francis Bach, Ivan Laptev, Jean Ponce, and Cordelia Schmid. Weakly-supervised alignment of video with text. In *ICCV*, 2015. 32, 72, 74, 75, 91

Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 18, 70

Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. Unsupervised pre-training of image features on non-curated data. In *ICCV*, 2019. 123

João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 4, 10, 19, 23, 45, 47, 48, 61, 105, 123, 124, 130, 139

João Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 19, 131

Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 19

David L Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *ACL*, 2011. 29, 35

Kai Chen, Hang Song, Chen Change Loy, and Dahua Lin. Discover and learn new objects from documentaries. In *CVPR*, 2017a. 39, 95, 96, 124, 146

Shaoxiang Chen, Xi Wang, Yongyi Tang, Xinpeng Chen, Zuxuan Wu, and Yu-Gang Jiang. Aggregating frame-level features for large-scale video classification. *arXiv preprint arXiv:1707.00803*, 2017b. 66

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 37

Xinlei Chen and C Lawrence Zitnick. Learning a recurrent visual representation for image caption generation. *arXiv preprint arXiv:1411.5654*, 2014. 28

Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *ICCV*, 2019. 24

Guilhem Chéron, Jean-Baptiste Alayrac, Ivan Laptev, and Cordelia Schmid. A flexible model for training action localization with varying levels of supervision. In *NeurIPS*, 2018. 125

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv preprint arXiv:1409.1259*, 2014. 26, 45, 48, 49, 52, 61, 135

Mithun Chowdhury, Panda Rameswar, Evangelos Papalexakis, and Amit Roy-Chowdhury. Webly supervised joint embedding for cross-modal image-text retrieval. In *ACM International Conference on Multimedia*, 2018. 30, 95, 124, 133

Timothee Cour, Chris Jordan, Eleni Miltsakaki, and Ben Taskar. Movie/script: Alignment and parsing of video and text transcription. In *ECCV*, 2008. 32

Timothée Cour, Benjamin Sapp, Chris Jordan, and Benjamin Taskar. Learning from ambiguously labeled images. In *CVPR*, 2009. 73, 86

Gabriela Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *ECCV Workshop*, 2004. 25, 46, 48, 56

Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *CVPR*, 2005. 16

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018. 20, 94, 96

Yann N. Dauphin, Fan Angela, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *arXiv preprint arXiv:1612.08083*, 2016. 49, 51, 63

César Roberto de Souza, Adrien Gaidon, Eleonora Vig, and Antonio Manuel López. Sympathy for the details: Dense trajectories and hybrid classification architectures for action recognition. In *ECCV*, 2016. 47

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 97, 136, 148

Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89 (1-2):31–71, 1997. 125

Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *arXiv preprint arXiv:1411.4389*, 2014. 26, 28, 45, 48, 55

Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *CVPR*, 2019. 30, 95, 124, 133

Matthis Douze, Jérôme Revaud, Cordelia Schmid, and Hervé Jégou. Stable hyper-pooling and query expansion for event detection. In *ICCV*, 2013. 57

Olivier Duchenne, Ivan Laptev, Josef Sivic, Francis Bach, and Jean Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009. 39, 70, 72, 125

Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *CVPR*, 2019. 123

Mark Everingham, Josef Sivic, and Andrew Zisserman. "Hello! My name is... Buffy" – Automatic Naming of Characters in TV Video. In *BMVC*, 2006. 39, 70, 72, 84

Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollar, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, et al. From captions to visual concepts and back. In *CVPR*, 2015. 28

Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 24

Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 45, 47, 48

Christoph Feichtenhofer, Axel Pinz, and Richard P. Wildes. Spatiotemporal multiplier networks for video action recognition. In *CVPR*, 2017. 23, 45, 47, 48

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 24

Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 2017. 123, 137

David F Fouhey, Wei-cheng Kuo, Alexei A Efros, and Jitendra Malik. From lifestyle vlogs to everyday interactions. In *CVPR*, 2018. 19

Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 1956. 72

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, pages 457–468, 2016. 34, 35, 95

Chuang Gan, Boqing Gong, Kun Liu, Hao Su, and Leonidas J Guibas. Geometry guided convolutional neural networks for self-supervised video representation learning. In *CVPR*, 2019. 123, 137

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. Tall: Temporal activity localization via language query. In *ICCV*, 2017. 31

Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019. 42, 123

Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *CVPR*, 2017. 26, 47, 48, 49, 55

R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Discriminatively trained deformable part models, release 5. 84

Yunchao Gong, Qifa Ke, Michael Isard, and Svetlana Lazebnik. A multi-view embedding space for modeling internet images, tags, and their semantics. *IJCV*, 2014a. 30, 95, 124

Yunchao Gong, Liwei Wang, Micah Hodosh, Julia Hockenmaier, and Svetlana Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *ECCV*, 2014b. 30, 95, 124

Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 20, 24

Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 19

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, 2010. 125, 128

Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. *arXiv preprint arXiv:1909.04656*, 2019. 37, 123, 137

Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, 2018. 24, 105

David Harwath, Adrià Recasens, Dídac Surís, Galen Chuang, Antonio Torralba, and James Glass. Jointly discovering visual objects and spoken words from raw sensory input. In *ECCV*, 2018. 95, 96, 124

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016. 16, 45, 51, 52, 60, 85, 105

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 37

Michael Heilman and Noah A Smith. Good question! statistical ranking for question generation. In *ACL*, 2010. 35

Olivier J Hénaff, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. *arXiv preprint arXiv:1905.09272*, 2019. 128, 129

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. *ICCV*, 2017. 31, 93, 94, 95, 96, 102, 103, 133

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with temporal language. In *EMNLP*, 2018. 31

Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. CNN architectures for large-scale audio classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. 60

Jack Hessel, Bo Pang, Zhenhai Zhu, and Radu Soricut. A case study on combining asr and visual features for generating instructional video captions. *arXiv preprint arXiv:1910.02930*, 2019. 124

Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. In *Neural Computing*, 1997. 26, 45, 48, 49, 52, 61, 135

Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *JAIR*, 2013. 28

Honnibal and Johnson. An improved non-monotonic transition system for dependency pars- ing. In *EMNLP*, 2015. 84

Harold Hotelling. Relations between two sets of variates. In *Breakthroughs in statistics*, pages 162–190. Springer, 1992. 30

Hexiang Hu, Wei-Lun Chao, and Fei Sha. Learning answer embeddings for visual question answering. In *CVPR*, 2018. 35

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017. 49

De-An Huang, Li Fei-Fei, and Juan Carlos Niebles. Connectionist temporal modeling for weakly supervised action labeling. In *ECCV*, 2016. 32, 96

De-An Huang, Joseph J. Lim, Li Fei-Fei, and Juan Carlos Niebles. Unsupervised visual-linguistic reference resolution in instructional videos. In *CVPR*, 2017. 96

De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. Finding "it": Weakly-supervised reference-aware visual grounding in instructional video. In *CVPR*, 2018. 96

Noureldien Hussein, Efstratios Gavves, and Arnold WM Smeulders. Timeception for complex action recognition. In *CVPR*, 2019. 26

Moustafa Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Mori Greg. A Hierarchical Deep Temporal Model for Group Activity Recognition. In *CVPR*, 2016. 26, 48

Maximilian Ilse, Jakub M Tomczak, and Max Welling. Attention-based deep multiple instance learning. *arXiv preprint arXiv:1802.04712*, 2018. 135

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate. *arXiv preprint arXiv:1502.03167*, 2015. 60

Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML*, 2013. 72

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017. 34

Hervé Jegou, Matthis Douze, Cordelia Schmid, and Patrick Perez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010. 25, 45, 46, 48, 55

S. Ji, W. Xu, M. Yang, and K. Yu. 3D Convolutional Neural Networks for Human Action Recognition. In *PAMI*, 2013. 23, 45, 47, 48

Longlong Jing and Yingli Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2018. 37, 123, 137

Justin Johnson, Andrej Karpathy, and Li Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*, 2016. 28, 95

M. I. Jordan. Hierarchical mixtures of experts and the em algorithm. *Neural Computation*, 1994. 50

A. Joulin, K. Tang, and L. Fei-Fei. Efficient image and video co-localization with Frank-Wolfe algorithm. In *ECCV*, 2014a. 75

Armand Joulin, Francis Bach, and Jean Ponce. Discriminative Clustering for Image Co-segmentation. In *CVPR*, 2010. 74, 91

Armand Joulin, Francis Bach, and Jean Ponce. Multi-class cosegmentation. In *CVPR*, 2012. 74

Armand Joulin, Kevin Tang, and Li Fei-Fei. Efficient image and video co-localization with frank-wolfe algorithm. In *ECCV*, 2014b. 72, 91

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016. 128, 129

Andrej Karpathy, Armand Joulin, and Fei Fei F Li. Deep fragment embeddings for bidirectional image sentence mapping. In *NIPS*, 2014a. 30, 102, 133

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014b. 18, 22, 42, 47, 70

Dotan Kauman, Gil Levi, Tal Hassner, and Lior Wolf. Temporal tessellation: A unified approach for video analysis. In *ICCV*, 2017. 40, 110, 111

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 60, 102, 131

Ryan Kiros, Ruslan Salakhutdinov, and Richard S Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *TACL*, 2014. 110, 111

Benjamin Klein, Guy Lev, Gil Sadeh, and Lior Wolf. Associating neural word embeddings with deep image representations using fisher vectors. In *CVPR*, 2015. 61, 93, 95, 109, 124, 139

Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018. 37, 123, 136, 137

Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *ICCV*, 2019. 26

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, Michael Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. 2016. URL `https://arxiv.org/abs/1602.07332`. 28, 43

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *ICCV*, 2017. 28, 29, 31, 95, 96

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 16, 45

Hilde Kuehne, Ahsan Iqbal, Alexander Richard, and Juergen Gall. Mining youtube-a dataset for learning fine-grained action concepts from webly supervised video data. *arXiv preprint arXiv:1906.01012*, 2019. 124

Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011. 18, 22, 131, 136

S Lacoste-Julien, M Jaggi, M Schmidt, and P Pletscher. Block-coordinate frank-wolfe optimization for structural svms. In *ICML*, 2013. 69, 70, 72, 75, 76, 144

Tian Lan, Yang Wang, and Greg Mori. Discriminative figure-centric models for joint action localization and recognition. In *ICCV*, 2011. 72

Ivan Laptev and Tony Lindeberg. Space-time Interest Points. In *ICCV*, 2003. 21

Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 17, 21, 25, 32, 39, 40, 45, 47, 48, 70, 83

Rémi Lebret, Pedro O Pinheiro, and Ronan Collobert. Phrase-based image captioning. *arXiv preprint arXiv:1502.03671*, 2015. 28

Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, 2017. 37, 123, 137

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L Berg. Tvqa: Localized, compositional video question answering. *arXiv preprint arXiv:1809.01696*, 2018. 34

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. Tvr: A large-scale dataset for video-subtitle moment retrieval. *arXiv preprint arXiv:2001.09099*, 2020. 40

Thomas Leung, Yang Song, and John Zhang. Handling label noise in video classification via multiple instance learning. In *ICCV*, 2011. 125

Guy Lev, Gil Sadeh, Benjamin Klein, and Lior Wolf. Rnn fisher vectors for action recognition and image annotation. In *ECCV*, 2016. 26, 48

Fu Li, Chuang Gan, Xiao Liu, Yunlong Bian, Xiang Long, Yandong Li, Zhichao Li, Jie Zhou, and Shilei Wen. Temporal modeling approaches for large-scale Youtube-8M video understanding. *arXiv preprint arXiv:1707.04555*, 2017. 66

Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. TGIF: A New Dataset and Benchmark on Animated GIF Description. In *CVPR*, 2016. 96

Chin-Yew Lin. Rouge: a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS)*, 2004. 28

Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 24

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. 28, 34, 43, 124

Jingen Liu, Jiebo Luo, and Mubarak Shah. Recognizing realistic actions from videos "in the wild". In *CVPR*, 2009. 18, 22

Jingzhou Liu, Wenhu Chen, Yu Cheng, Zhe Gan, Licheng Yu, Yiming Yang, and Jingjing Liu. Violin: A large-scale dataset for video-and-language inference. *arXiv preprint arXiv:2003.11618*, 2020. 40

Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. *arXiv preprint arXiv:1907.13487*, 2019. 30

David G Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 16

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019a. 28, 43

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 12-in-1: Multi-task vision and language representation learning. *arXiv preprint arXiv:1912.02315*, 2019b. 43

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018a. 97

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018b. 42

Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. In *NIPS*, 2014. 34, 35

Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *ICCV*, 2015. 34, 95

Jonathan Malmaud, Jonathan Huang, Vivek Rathod, Nick Johnston, Andrew Rabinovich, and Kevin Murphy. What's cookin'? interpreting cooking videos using text, speech and vision. *NAACL*, 2015. 37, 38, 94, 96, 97, 124

Marcin Marszalek, Ivan Laptev, and Cordelia Schmid. Actions in context. In *CVPR*, 2009. 18, 22, 25

Markus Mathias, Rodrigo Benenson, Marco Pedersoli, and Luc Van Gool. Face detection without bells and whistles. In *ECCV*, 2014. 84

Antoine Miech, Jean-Baptiste Alayrac, Piotr Bojanowski, Ivan Laptev, and Josef Sivic. Learning from Video and Text via Large-Scale Discriminative Clustering. In *ICCV*, 2017a. 11, 125

Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017b. 11, 12, 49, 102

Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a Text-Video Embedding from Incomplete and Heterogeneous Data. *arXiv preprint arXiv:1804.02516*, 2018. 11, 93, 94, 95, 101, 102, 124, 133

Antoine Miech, Ivan Laptev, Josef Sivic, Heng Wang, Lorenzo Torresani, and Du Tran. Leveraging the present to anticipate the future in videos. In *CVPRW Precognition workshop*, 2019a. 12, 25

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100M: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, 2019b. 11, 12, 120, 121, 122, 123, 124, 127, 128, 130, 132, 138, 139, 140

Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*, 2020. 11, 13

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 105, 130

Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016. 37, 123, 137

Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *ICMR*. ACM, 2018. 30, 95, 124, 133

Niluthpol Chowdhury Mithun, Sujoy Paul, and Amit K Roy-Chowdhury. Weakly supervised video moment retrieval from text queries. In *CVPR*, 2019. 31

Yasufumi Moriya, Ramon Sanabria, Florian Metze, and Gareth JF Jones. Grounding object detections with transcriptions. *arXiv preprint arXiv:1906.06147*, 2019. 124

Arsha Nagrani, Chen Sun, David Ross, Rahul Sukthankar, Cordelia Schmid, and Andrew Zisserman. Speech2action: Cross-modal supervision for action recognition. *arXiv preprint arXiv:2003.13594*, 2020. 39, 40

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 128, 129

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Is object localization for free? - weakly-supervised learning with convolutional neural networks. In *CVPR*, June 2015. 135

Vicente Ordonez, Girish Kulkarni, and Tamara L. Berg. Im2text: Describing images using 1 million captioned photographs. In *NIPS*, 2011. 28, 41

Anton Osokin, Jean-Baptiste Alayrac, Isabella Lukasewitz, Puneet Dokania, and Simon Lacoste-Julien. Minding the gaps for block frank-wolfe optimization of structured svms. In *ICML*, 2016. 9, 78

Shruti Palaskar, Jindrich Libovickỳ, Spandana Gella, and Florian Metze. Multimodal abstractive summarization for how2 videos. *arXiv preprint arXiv:1906.07901*, 2019. 124

Pingbo Pan, Zhongwen Xu, Yi Yang, Fei Wu, and Yueting Zhuang. Hierarchical recurrent neural encoder for video representation with application to captioning. In *CVPR*, pages 1029–1038, 2016a. 28, 93, 95

Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. Jointly modeling embedding and translation to bridge video and language. In *CVPR*, 2016b. 30, 94, 95, 124

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002. 28

Omkar Parkhi, Esa Rahtu, and Andrew Zisserman. It's in the bag: Stronger supervision for automated face labelling. In *ICCV Workshop*, 2015a. 39, 73, 80, 86, 87, 125

Omkar Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep Face Recognition, British Machine Vision Conference. In *BMVC*, 2015b. 84

Nikolaos Passalis and Anastasios Tefas. Learning neural bag-of-features for large scale image retrieval. *IEEE Trans. Cybernetics*, 2017. 56

Xiaojiang Peng, Limin Wang, Yu Qiao, and Qiang Peng. Boosting VLAD with Supervised Dictionary Learning and High-Order Statistics. In *ECCV*, 2014a. 26, 48, 49

Xiaojiang Peng, Changqing Zou, Yu Qiao, and Qiang Peng. Action recognition with stacked fisher vectors. In *ECCV*, 2014b. 26, 49

Florent Perronnin and Christopher Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007. 25, 46, 48, 56, 57

Florent Perronnin and Diane Larlus. Fisher Vectors Meet Neural Networks: A Hybrid Classification Architecture. In *CVPR*, 2015. 48

James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008. 56

AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Evolving losses for unsupervised video representation learning. *arXiv preprint arXiv:2002.12177*, 2020. 37

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015. 124

Bryan A Plummer, Matthew Brown, and Svetlana Lazebnik. Enhancing video summarization via vision-language embedding. In *CVPR*, 2017. 32, 94, 95, 124

Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. Imagebert: Cross-modal pre-training with large-scale weak-supervised image-text data. *arXiv preprint arXiv:2001.07966*, 2020. 42, 43

Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017. 23, 24

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understandingby Generative Pre-Training. *preprint*, 2018. 97

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *preprint*, 2019. 97

Vignesh Ramanathan, Armand Joulin, Percy Liang, and Li Fei-Fei. Linking people in videos with "their" names using coreference resolution. In *ECCV*, 2014. 74, 125

Alexander Richard and Juergen Gall. A bag-of-words equivalent recurrent neural network for action recognition. In *BMVC*, 2015. 56

Alexander Richard, Hilde Kuehne, and Juergen Gall. Weakly supervised action learning with rnn based fine-to-coarse modeling. In *CVPR*, 2017. 32, 96

Alexander Richard, Hilde Kuehne, and Juergen Gall. Action sets: Weakly supervised action segmentation without ordering constraints. In *CVPR*, 2018. 96

Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *CVPR*, 2015. 40, 47, 58, 59, 71, 96

Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *IJCV*, 2017. 96, 104, 106, 124

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 2015. 4, 10, 16, 124

Ramon Sanabria, Ozan Caglayan, Shruti Palaskar, Desmond Elliott, Loïc Barrault, Lucia Specia, and Florian Metze. How2: a large-scale dataset for multimodal language understanding. In *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*. NeurIPS, 2018. 33, 37, 38, 96, 97, 123, 124

Christian Schüldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: a local svm approach. In *ICPR*, 2004. 17, 22, 23, 45, 47, 48

G Seguin, P Bojanowski, R Lajugie, and I Laptev. Instance-level video segmentation from object tracks. In *CVPR*, 2016. 72, 74, 75, 91

Fadime Sener and Angela Yao. Unsupervised learning and segmentation of complex activities from video. In *CVPR*, 2018. 96

Ozan Sener, Amir R. Zamir, Silvio Savarese, and Ashutosh Saxena. Unsupervised semantic parsing of video collections. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 37, 38, 97

Ren Shaoqing, He Kaiming, Girshick Ross, and Sun Jian. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 85

Nataliya Shapovalova, Arash Vahdat, Kevin Cannons, Tian Lan, and Greg Mori. Similarity constrained latent support vector machine: An application to weakly supervised action classification. In *ECCV*, 2012. 72, 125

Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 41, 42, 43

Zhiqiang Shen, Jianguo Li, Zhou Su, Minjun Li, Yurong Chen, Yu-Gang Jiang, and Xiangyang Xue. Weakly supervised dense video captioning. In *CVPR*, 2017. 28

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016a. 19

Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, 2016b. 96

Gunnar A Sigurdsson, Jean-Baptiste Alayrac, Aida Nematzadeh, Lucas Smaira, Mateusz Malinowski, João Carreira, Phil Blunsom, and Andrew Zisserman. Visual grounding in video for unsupervised word translation. In *CVPR*, 2020. 32, 33, 39

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *ICLR*, pages 568–576, 2014. 22, 48, 70

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 16, 45

Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep fisher networks for large-scale image classification. In *NIPS*, 2013. 57

Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 25, 46, 48, 56

Josef Sivic, Mark Everingham, and Andrew Zisserman. "Who are you?" - Learning person specific classifiers from video. In *CVPR*, 2009. 39, 70, 72, 84, 86, 87

Miha Skalic, Marcin Pekalski, and Xingguo E. Pan. Deep learning methods for efficient large scale video labeling. *arXiv preprint arXiv:1706.04572*, 2017. 66

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 18, 23, 70, 131, 136

Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *CVPR*, 2018. 125

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017. 42, 98

Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*, 2019a. 25, 123, 124, 132, 137, 138

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019b. 37, 38, 124

Vladyslav Sydorov, Mayu Sakurada, and Christoph H Lampert. Deep fisher kernels and end to end learning of the Fisher kernel GMM parameters. In *CVPR*, 2014. 49, 57

Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Edouard Grave, Tatiana Likhomanenko, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, and Ronan Collobert. End-to-end asr: from supervised to semi-supervised learning with modern architectures. *arXiv preprint arXiv:1911.08460*, 2019. 37

Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv:1602.07261v1*, 2016. 45

Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, 2019. 97, 132, 137

Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. "Knock! Knock! Who is it?" probabilistic person identification in tv-series. In *CVPR*, 2012. 70

Makarand Tapaswi, Martin Bäuml, and Rainer Stiefelhagen. Book2movie: Aligning video scenes with book chapters. In *CVPR*, 2015. 32

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *CVPR*, 2016. 34, 93, 95

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 2016. 41

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 37, 123, 137

Atousa Torabi, Christopher Pal, Hugo Larochelle, and Aaron Courville. Using descriptive video services to create a large data source for video annotation research. *arXiv preprint arXiv:1503.01070*, 2015. 40, 96

Atousa Torabi, Niket Tandon, and Leonid Sigal. Learning language-visual embedding for movie understanding with natural-language. *arXiv preprint arXiv:1609.08124*, 2016. 40, 61, 110, 111

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 23, 45, 47, 48

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 24

Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *ICCV*, 2019. 24

Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term Temporal Convolutions for Action Recognition. *PAMI*, 2017. 23, 45, 47, 48

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 28, 38, 135, 148

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *CVPR*, 2015. 28

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015. 28

Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Anticipating visual representations from unlabeled video. In *CVPR*, 2016. 123

Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018. 123

He-Da Wang, Teng Zhang, and Ji Wu. The monkeytyping solution to the YouTube-8M video understanding challenge. *arXiv preprint arXiv:1706.05150*, 2017. 66

Heng Wang and Cordelia Schmid. Action Recognition with Improved Trajectories. In *ICCV*, 2013. 22, 25, 45, 47, 48, 70, 85

Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *CVPR*, 2011. 22, 25

Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *CVPR*, 2019a. 123, 137

Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, pages 4305–4314, 2015. 47

Limin Wang, Yuanjun Xiong, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016a. 23, 25, 48

Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016b. 30, 93, 95, 102, 124, 133

Liwei Wang, Yin Li, Jing Huang, and Svetlana Lazebnik. Learning two-branch neural networks for image-text matching tasks. *PAMI*, 2018a. 30, 93, 95, 102, 124, 133

Xiaolong Wang and Abhinav Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, 2015. 123

Xin Wang, Wenhu Chen, Jiawei Wu, Yuan-Fang Wang, and William Yang Wang. Video captioning via hierarchical reinforcement learning. In *CVPR*, 2018b. 28

Xin Wang, Jiawei Wu, Da Zhang, Yu Su, and William Yang Wang. Learning to compose topic-aware mixture of experts for zero-shot video captioning. In *AAAI*, 2018c. 28, 95

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*, 2019b. 29, 33

Yongqiang Wang, Abdelrahman Mohamed, Due Le, Chunxi Liu, Alex Xiao, Jay Mahadeokar, Hongzhao Huang, Andros Tjandra, Xiaohui Zhang, Frank Zhang, et al. Transformer-based acoustic modeling for hybrid speech recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6874–6878. IEEE, 2020. 37

Philippe Weinzaepfel, Xavier Martin, and Cordelia Schmid. Towards weakly-supervised action localization. *arXiv preprint arXiv:1605.05197*, 2016. 72, 125

Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *ICCV*, 2019. 124, 133

Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *CVPR*, 2019. 26

Chao-Yuan Wu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. *ICCV*, 2017. 30, 95, 124, 133

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 24, 130, 137, 138

Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM Multimedia*, 2017. 34, 35

Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *CVPR*, 2019. 37, 123, 137

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 29, 35, 94, 96, 104, 106, 124, 131

Ran Xu, Caiming Xiong, Wei Chen, and Jason J Corso. Jointly modeling deep video and compositional text to bridge vision and language in a unified framework. In *AAAI*, 2015a. 30, 94, 95, 124

Zhongwen Xu, Yi Yang, and Alexander G. Hauptmann. A Discriminative CNN Video Representation for Event Detection. In *CVPR*, 2015b. 48

Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *CVPR*, pages 4651–4659, 2016. 28, 95

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *ACL*, 2014. 28, 43

Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. In *CVPR*, pages 4584–4593, 2016a. 28, 93, 95

Shoou-I Yu, Lu Jiang, and Alexander Hauptmann. Instructional videos for unsupervised harvesting and learning of action examples. In *ACM*, 2014. 37, 38, 96, 124

Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. Video captioning and retrieval models with semantic attention. In *ECCV LSMDC2016 Workshop*, 2016b. 40, 61, 102, 110, 111

Youngjae Yu, Jongseok Kim, and Gunhee Kim. Joint sequence fusion model for video question-answering and retrieval. `https://drive.google.com/file/d/0B9n0ObAFqKC9MGNDTkJWejM4dE0/view`, 2017a. Presented at ICCV17 LSMDC17 workshop. 50, 61, 62

Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *CVPR*, 2017b. 28, 40, 61, 110, 111

Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *ECCV*, 2018. 30, 40, 93, 95, 106, 110, 111

Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015. 26, 48

Kuo-Hao Zeng, Tseng-Hung Chen, Juan Carlos Niebles, and Min Sun. Title generation for user generated videos. In *ECCV*, 2016. 28, 42

Kuo-Hao Zeng, Tseng-Hung Chen, Ching-Yao Chuang, Yuan-Hong Liao, Juan Carlos Niebles, and Min Sun. Leveraging video descriptions to learn video question answering. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 34

Da Zhang, Xiyang Dai, Xin Wang, Yuan-Fang Wang, and Larry S Davis. Man: Moment alignment network for natural language moment retrieval via iterative graph adjustment. In *CVPR*, 2019. 31

Luowei Zhou, Xu Chenliang, and Jason J. Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018a. 96

Luowei Zhou, Chenliang Xu, and Jason J Corso. Towards automatic learning of procedures from web instructional videos. In *AAAI*, 2018b. 13, 29, 95, 96, 97, 104, 106, 109, 124, 131

Luowei Zhou, Yingbo Zhou, Jason J. Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *CVPR*, 2018c. 28

Yuke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In *CVPR*, 2016. 34, 35

Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019. 94, 95, 96, 97, 104, 105, 106, 108, 109, 132, 138, 140

RÉSUMÉ

Nous nous intéressons à l'apprentissage automatique d'algorithmes pour la compréhension automatique de vidéos. Une majorité des approaches en compréhension de vidéos dépend de large base de données de vidéos manuellement annotées pour l'entraînement. Cependant, la collection et l'annotation de telles base de données est fastidieuse, coûte cher et prend du temps. Pour palier à ce problème, cette thèse se concentre sur l'exploitation de large quantité d'annotations publiquement disponible, cependant bruitées, sous forme de language naturel. En particulier, nous nous intéressons à un corpus divers de métadonnées textuelles incluant des scripts de films, des titres et descriptions de vidéos internet ou encore des transcriptions de paroles. L'usage de ce type de données publiquement disponibles est difficile car l'annotation y est faible. Pour cela, nous introduisons différentes approches d'apprentissage telles que de nouvelles fonctions de coûts ou architectures de réseaux de neurones, adaptées à de faibles annotations.

MOTS CLÉS

Vision par ordinateur, Analyse de vidéo, Vidéo et language, Apprentissage faiblement supervisé, Apprentissage profond, Apprentissage machine.

ABSTRACT

The goal of this thesis is to build and train machine learning models capable of understanding the content of videos. Current video understanding approaches mainly rely on large-scale manually annotated video datasets for training. However, collecting and annotating such dataset is cumbersome, expensive and time-consuming. To address this issue, this thesis focuses on leveraging large amounts of readily-available, but noisy annotations in the form of natural language. In particular, we exploit a diverse corpus of textual metadata such as movie scripts, web video titles and descriptions or automatically transcribed speech obtained from narrated videos. Training video models on such readily-available textual data is challenging as such annotation is often imprecise or wrong. In this thesis, we introduce learning approaches to deal with weak annotation and design specialized training objectives and neural network architectures.

KEYWORDS

Computer vision, Video analysis, Video and language, Weakly-supervised learning, Deep learning, Machine learning.