THÈSE DE DOCTORAT DE L'UNIVERSITÉ DE LYON
opérée au sein de
INSA LYON

École Doctorale 512
Informatique et Mathématique de Lyon
(INFOMATHS)

Spécialité
Informatique

Présentée par

Fabien Baradel

Pour obtenir le grade de
DOCTEUR de L'UNIVERSITÉ DE LYON

Sujet de la thèse :

Structured deep learning for video analysis

Apprentissage profond structuré pour l'analyse de vidéos

Soutenue publiquement le 4 juin 2020, devant le jury composé de :

M. Ivan Laptev	INRIA	Rapporteur
M. Jakob Verbeek	Facebook AI Research	Rapporteur
M. David Picard	École des Ponts ParisTech	Présdient
Mme. Diane Larlus	Naver Labs Europe	Examinateur
M. Julien Mille	INSA Centre Val de Loire - LIFAT	Co-encadrant de thèse
M. Christian Wolf	INSA Lyon - LIRIS	Directeur de thèse
Mme. Natalia Neverova	Facebook AI Research	Invitée
Mme. Cordelia Schmid	INRIA - Google	Invitée

ABSTRACT

With the massive increase of video content on Internet and beyond, the automatic understanding of visual content could impact many different application fields such as robotics, health care, content search or filtering. The goal of this thesis is to provide methodological contributions in Computer Vision (CV) and Machine Learning (ML) for automatic content understanding from videos. We emphasis on problems, namely fine-grained human action recognition and visual reasoning from object-level interactions.

In the first part of this manuscript, we tackle the problem of fine-grained human action recognition. We introduce two different trained attention mechanisms on the visual content from articulated human pose. The first method is able to automatically draw attention to important pre-selected points of the video conditioned on learned features extracted from the articulated human pose. We show that such mechanism improves performance on the final task and provides a good way to visualize the most discriminative parts of the visual content. The second method goes beyond pose-based human action recognition. We develop a method able to automatically identify unstructured feature clouds of interest in the video using contextual information. Furthermore, we introduce a learned distributed system for aggregating the features in a recurrent manner and taking decisions in a distributed way. We demonstrate that we can achieve a better performance than obtained previously, without using articulated pose information at test time.

In the second part of this thesis, we investigate video representations from an object-level perspective. Given a set of detected persons and objects in the scene, we develop a method which learns to infer the important object interactions through space and time using the video-level annotation only. That allows to identify important objects and object interactions for a given action, as well as potential dataset bias.

Finally, in a third part, we go beyond the task of classification and supervised learning from visual content by tackling causality in interactions, in particular the problem of counterfactual learning. We introduce a new benchmark, namely CoPhy, where, after watching a video, the task is to predict the outcome after modifying the initial stage of the video. We develop a method based on object-level interactions able to infer object properties without supervision as well as future object locations after the intervention.

RÉSUMÉ

Avec l'augmentation massive du contenu vidéo sur Internet et au-delà, la compréhension automatique du contenu visuel pourrait avoir un impact sur de nombreux domaines d'application différents tels que la robotique, la santé, la recherche de contenu ou le filtrage. Le but de cette thèse est de fournir des contributions méthodologiques en vision par ordinateur et apprentissage statistique pour la compréhension automatique du contenu des vidéos. Nous mettons l'accent sur les problèmes de la reconnaissance de l'action humaine à grain fin et du raisonnement visuel à partir des intéractions entre objets.

Dans la première partie de ce manuscrit, nous abordons le problème de la reconnaissance fine de l'action humaine. Nous introduisons deux différents mécanismes d'attention, entrainés sur le contenu visuel à partir de la pose humaine articulée. Une première méthode est capable de porter automatiquement l'attention sur des points pré-sélectionnés importants de la vidéo, conditionnés sur des caractéristiques apprises extraites de la pose humaine articulée. Nous montrons qu'un tel mécanisme améliore les performances sur la tâche finale et fournit un bon moyen de visualiser les parties les plus discriminantes du contenu visuel. Une deuxième méthode va au-delà de la reconnaissance de l'action humaine basée sur la pose. Nous développons une méthode capable d'identifier automatiquement un nuage de points caractéristiques non structurés pour une video à l'aide d'informations contextuelles. De plus, nous introduisons un système distribué entrainé pour agréger les caractéristiques de manière récurrente et prendre des décisions de manière distribuée. Nous démontrons que nous pouvons obtenir de meilleures performances que celles illustrées précédemment, sans utiliser d'informations de pose articulée au moment de l'inférence.

Dans la deuxième partie de cette thèse, nous étudions les représentations vidéo d'un point de vue objet. Étant donné un ensemble de personnes et d'objets détectés dans la scène, nous développons une méthode qui a appris à déduire les interactions importantes des objets à travers l'espace et le temps en utilisant uniquement l'annotation au niveau vidéo. Cela permet d'identifier une interaction inter-objet importante pour une action donnée ainsi que le biais potentiel d'un ensemble de données.

Enfin, dans une troisième partie, nous allons au-delà de la tâche de classification et d'apprentissage supervisé à partir de contenus visuels, en abordant la causalité à travers les interactions, et en particulier le problème de l'apprentissage contrefactuel. Nous introduisons une nouvelle base de données, à savoir CoPhy, où, après avoir regardé une vidéo, la tâche consiste à prédire le résultat après avoir modifié la phase initiale de la vidéo. Nous développons une méthode basée sur des interactions au niveau des objets capables d'inférer les propriétés des objets sans supervision ainsi que les emplacements futurs des objets après l'intervention.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

NLP	Natural Language Processing
STN	Spatial Transformer Network
RL	Reinforcement Learning
AMT	Amazon Mechanical Turk
CNN	Convolutional Neural Network
AI	Artificial Intelligence
ANN	Artificial Neural Network
BoW	Bag-of-Words
CV	Computer Vision
DL	Deep Learning
DPM	Deformable Part-based Model
FV	Fisher Vector
VLAD	Vector of Locally Aggregated Descriptors
GAP	Global Average Pooling
GCN	Graph Convolutional Network
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
HOG	Histogram of Oriented Gradients
LSTM	Long-Short Term Memory
LSVM	Latent Support Vector Machine
MLP	Multi-Layer Perceptron
ML	Machine Learning
ReLU	Rectified Linear Unit
RN	Relation Network
RNN	Recurrent Neural Network
RPN	Region Proposal Network
RoI	Region of Interest
SIFT	Scale-Invariant Feature Transform
STIP	Space-Time Interest Points
SGD	Stochastic Gradient Descent

SOTA	State Of The Art
SPM	Spatial Pyramid Matching
SSD	Single Shot Detector
SVM	Support Vector Machine
VQA	Visual Question Answering
YOLO	You Only Look Once
CAD	Computer-Aided Design
HOF	Histograms of Optical Flow
MBH	Motion Boundary Histograms
IDT	Improved Dense Trajectories

Chapter 1

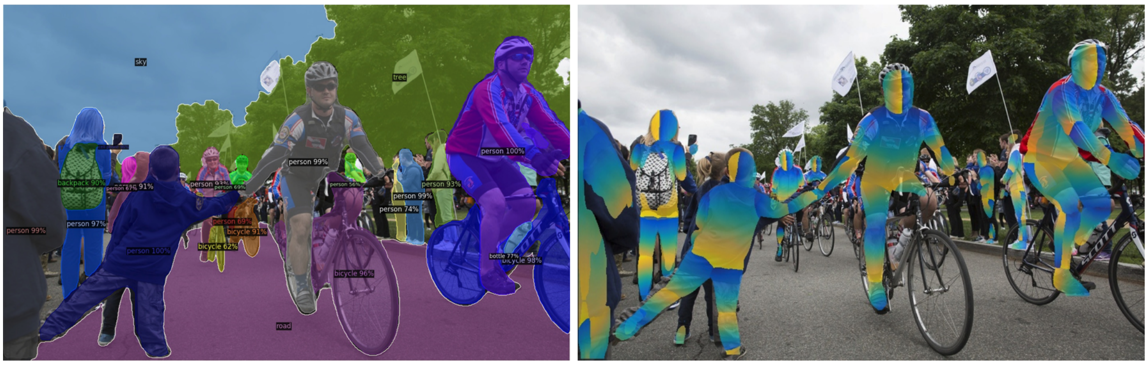
INTRODUCTION

Contents

1.1 Context

The field of Artificial Intelligence (AI) has received increasing attention over the last decades with the underlying objective being to make machines reach human capacities on specific tasks (*"narrow AI"*), or surpass them. Most of us use AI-powered applications in our everyday life without realizing it. For instance, to access a specific video content based on keywords, we employ a retrieval system. We also use a recommendation system to watch videos with similar or related semantic content. And if we want to communicate with someone from the other side of the world who does not speak our language, automatic translation tools can make our life easier.

Recently great progress has been demonstrated in Computer Vision (CV), a subfield of AI, which consists in automatically extracting high-level understanding from images or videos. This domain is of high interest given the increase of visual digital content since the advent of Internet and the popularization of digital photography. Everyday 300 millions photos are uploaded every day on Facebook (Noyes 2015) and 500 hours of videos per minute (Hale 2019) on YouTube. Nearly 80% of the internet traffic is due to the transfer of video data. Given this large amount of data shared every single second in the world, it becomes crucial to automatically process this visual content with machines. A classical task in CV called object recognition consists in classifying an image among a set of predefined objects. Object recognition methods extract semantic information at the image-level. More sophisticated tasks have been proposed such as human pose



Panoptic segmentation

Dense human pose estimation

Figure 1.1 – **Output of detection methods.** (Left) *Panoptic segmentation* combines instance segmentation and semantic segmentation by assigning a semantic label to each part of the scene. — (Right) *Dense human pose estimation* produces a human body mesh for each detected person. Figure reproduced from (Wu et al. 2019c).

estimation or image captioning, which can take decisions on a finer level detail, or decisions of different types. The common point is that they all propose to extract semantic content of different nature.

Machine Learning (ML) algorithms play an important role in modern CV methods. This is mainly due to the *semantic gap* (Smeulders et al. 2000) between raw signals representing low-level information (i.e. pixel colors) and the high-level semantic meaning of this content. Bridging this gap requires the extraction of high-level discriminative features, and for many years a solution for this has been to design *handcrafted* image descriptors such as Scale-Invariant Feature Transform (SIFT) and to aggregate them with a method like Bags-of-Words Bag-of-Words (BoW) to produce a single feature vector. ML and classifiers have been employed on top of these representations to solve CV tasks such as object recognition.

The downside of this strategy is that it requires extensive expert knowledge to design powerful features, which cannot be easily extendable to any type of data, and the discriminative power of this kind of descriptors is quite limited. The introduction of Deep Learning (DL), and in particular Convolutional Neural Network (CNN), aims at solving this issue. They consist in jointly learning feature extractors and the classifier to solve a given task. CNNs are the *de facto* standard since 2012 for solving CV tasks, although they were actually introduced a few decades ago (Fukushima 1980; LeCun et al. 1997). There are some (admitted, limited) analogies with the brain, as they are inspired by the receptive field representation (Hubel et al. 1959). The recent success of CNNs is mainly due to the introduction of large scale *annotated* image datasets to avoid overfitting and

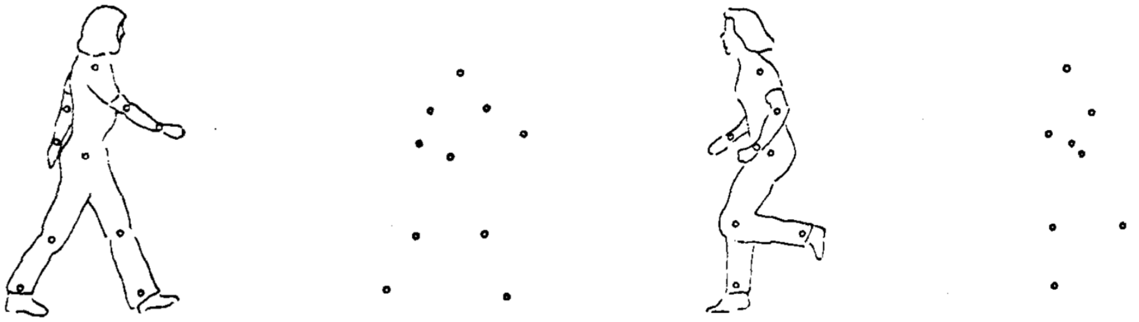


Figure 1.2 – **Moving Light Displays attached to the human body.** The action performed by a human can be recognized by a few moving light displays attached to the human body (Johansson 1973).
(Left): a person is walking. — (Right) a person is running.
Figure reproduced from (Johansson 1973).

the use of graphics processing units for speeding up the computational time by a factor of ~ 20 compared to central processing units.

CNNs learn hierarchical representations from low-level to high-level features, which also allows to transfer representations learned for one task to a different task. Indeed, features learned by training on a large scale object recognition dataset (i.e. Imagenet (Russakovsky et al. 2015)) have shown to be easily transferable for solving detection tasks such as shown in Figure 1.1.

However, the situation is a little bit different when video content is considered. Despite the efforts to annotate large scale video datasets (Kay et al. 2017), CNNs designed for extracting semantic content from videos are still highly biased towards context and background information. This is relatively annoying when we target systems required to extract *fine-grained* information, as for instance human action recognition or human-object interaction. Moreover, *handcrafted* methods are still achieving good performance on standard benchmarks (Wang et al. 2013a), which (arguably) indicates that we are not yet able to fully exploit the power of large capacity networks on these problems.

The goal of this manuscript is to provide DL based models and methods for solving these issues. To do so, we propose approaches which structure deep neural networks, and which are inspired by human abilities with a focus on visual attention and visual reasoning, which we will develop below.

1.2 Motivations

1.2.1 Visual attention

Laptev (2013) observed that about 35% of pixels in videos contained people, which makes this category extremely specific and important for identifying semantic concepts. While humans are highly complex and deformable objects, Johansson (1973) demonstrates that the visual interpretation of a few moving light displays attached to the human body can be sufficient for categorizing the action performed by a person, as shown in Figure 1.2. This indicates that a high-level understanding of motion can be achieved from a structured and low-dimensional representation. This has been applied to CV through the description of visual content of humans through articulated pose, mostly as a set of coordinates of selected joints or key points. Recent works (Liu et al. 2017b; Song et al. 2016) achieve high performance at predicting human actions using human articulated pose only. However, these pose-only methods remain limited in extracting information related to humans only, whereas understanding complex situations in videos often requires understanding context information and interactions between humans and the environment.

Complementary information can be extracted by gathering visual cues in the scene. Humans are particularly efficient in extracting information in the spatio-temporal domain. The human perception focuses selectively on parts of the scene to acquire information at specific places and times, a process known as *visual attention*. Yarbus (1967) demonstrate this principle by showing that eye gazes over an image depend on the task that the observer needs to perform.

In ML and CV this kind of process is referred to as an attention mechanism, and has drawn increasing interest in several fields and on tasks dealing with languages, images and data (Ba et al. 2015; Larochelle et al. 2010). Integrating attention can potentially lead to improvements of the overall system, as it may focus on parts of the data relevant to the task. Up to our knowledge, in CV, visual attention mechanisms are first introduced by Itti et al. (1998) inspired by works in neuroscience (Jonides 1983) and a learning component is integrated few years later by Larochelle et al. (2010).

Recent works in video understanding have shown that coupling a CNN with attention mechanisms can boost the performance of the overall system by attending to selected parts of the visual content given contextual information (Sharma et al. 2016; Wang et al. 2018a). Girdhar et al. (2017) propose to use the human pose estimation as an inductive bias for extracting visual cues around persons present in the scene.

Visual attention is a key point of the contributions we develop in this thesis. In particular, we study the role of articulated pose as source for attention on

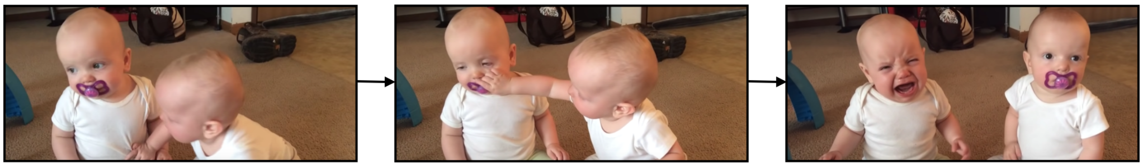


Figure 1.3 – **Example of daily life video.** The right-side baby is stealing the silencer from the left-side baby. Understanding the content of such a video goes beyond extracting high-level semantic information such as human actions. For instance, explaining why the left-side baby is crying at the end of the video needs to reason about what about happened during the video.
Images from youtu.be/UOlOrACAj6o.

visual data (Chapter 3), and we study attention mechanisms, which freely attend to points in spatio-temporal data, exploiting the fact that (unlike humains) an artificial agent is not bound to the physical notion of time and can attend to several glimpses (points) sequentially for a given time instant, only bound by restrictions on computational complexity and latency (Chapter 4).

1.2.2 Towards visual reasoning

Efficient information extraction plays an important role in human perception. Humans are able to infer what happened in a video given only a few sample frames. This faculty is called *reasoning* and is a key component of human intelligence. As an example we can consider the images in Figure 1.3, which show a complex situation involving articulated objects (two babies and a silencer), the change of location and composition of objects. For humans it is straightforward to draw a conclusion on what happened. The right-side baby steals the silencer from the left-side baby and the left-side baby starts crying.

Humans have this extraordinary ability to perform visual reasoning on very complicated tasks while this currently remains unattainable for contemporary CV algorithms (Fleuret et al. 2011). The goal of modern CV methods is to create (and therefore mostly train) systems with increased reasoning capacity, which thus rely less on memorizations and exploiting biases in training datasets. A particular task where this visual reasoning aspect has been introduced is Visual Question Answering (VQA) (Malinowski et al. 2016), which consists in answering any question formulated in a natural language question about any image. Recent approaches develop systems that take into account object interactions (Teney et al. 2017) and which require multi-hop reasoning (Ben-Younes* et al. 2019).

The notion of causality and causal chains of reasoning is at the heart of these considerations. For instance, recent works model video content by leveraging time as an explicit causal signal to identify causal object relations (Wang et al. 2018b;

Pickup et al. 2014). Such approaches rely on the concept of the *arrow of the time* involving the asymmetric nature of time. For the case of human action recognition, it is desirable to identify causal events or causal object relations happening in a video which affect its label, and mostly from an object level point of view (Sun et al. 2018). Object-level representations are at the heart of our contributions, in particular in Chapter 5 on object-level visual reasoning, and in Chapter 6, which reasons on physical interactions at an object level.

Humans show an incredible capacity at discovering *causal effects* from observations only. We make sense of fundamental concepts that ensures ability to leverage such experiences for robust *generalization* to new scenarios (Martin-Ordas et al. 2008). This can be explained by the ability that humans have to employ retrospection by accessing their past experiences and being able to judge them. Beyond modeling object interactions, humans are able to anticipate what would be a similar situation under a different scenario. For instance in Figure 1.3, would the left-side baby be crying at the end of the video if the silencer was not present in the first frame? It is likely that the answer is no, because the silencer seems to be the point of contention between the two babies. This way of expressing causality is based on the concept of *counterfactual* reasoning, which deals with a problem containing an 'if statement', which is untrue of unrealized. Using counterfactuals has been shown to be a way to perform reasoning over causal relationships between variables of low dimensional spaces (Balke et al. 1994; Tian et al. 2002; Tian et al. 2002). Recent works in ML (Bottou et al. 2013; Johansson et al. 2016) are trying to investigate this concept for bringing more cognitive ability to the learned system. In this thesis, we develop counterfactual reasoning on image content in chapter Chapter 6.

1.3 Contributions

Funded by ANR project "*Deepvision*" [1], in this thesis we propose DL based methods for tackling the problem of video analysis.

First, we focus on the problem of fine-grained human action recognition on trimmed videos and propose **visual attention** mechanisms. In Chapter 3, we present our first contribution which proposes human articulated pose to draw attention the most relevant part of images. Our model automatically selects discriminative patches of the video over a set of pre-defined potential attention points. Then, in Chapter 4 we introduce a method using RGB images without using articulated human pose at inference time. We build a model able to

[1] Deepvision (`https://projet.liris.cnrs.fr/deepvis/`) is a joint French-Canadien ANR/NSERC projet involving partners INSA-Lyon/LIRIS, Sorbonne University/LIP6, University of Guelph, Simon Fraser University which focuses on structured deep learning for human understanding.

automatically find discriminative points of interest in the video as well as a distributed recognition module.

Second, we propose video analysis methods from an object-level perspective pushing towards **visual reasoning**. In Chapter 5, we present a video classification method based on object interactions. We assume that a given scene is decomposed into a set of semantically identified visual concepts (persons and objects) using a trained object mask detector and we introduce a model able to learn important object interactions through time and space, an to aggregate them for video classification. In Chapter 6, we introduce a new problem of counterfactual learning from visual input. Given an initial sequence, we ask the system to produce a counterfactual prediction. We tackle this problem by introducing a new benchmark for intuitive physics and a new method for counterfactual learning. Given an observed sequence we encode the object properties into a latent space which is then used for predicting the outcome after the intervention in the initial stage.

List of publications — This manuscript is based on the material published in the following papers:

- Fabien Baradel, Christian Wolf, and Julien Mille (2017b). "Human Action Recognition: Pose-based Attention draws focus to Hands". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV) - Workshop "Hands in Action"* - Chapter 3;

- Fabien Baradel, Christian Wolf, and Julien Mille (2018b). "Human Activity Recognition with Pose-driven Attention to RGB". in: *Proceedings of the British Machine Vision Conference (BMVC)* - Chapter 3;

- Fabien Baradel, Christian Wolf, Julien Mille, and Graham Taylor (2018c). "Glimpse Clouds: Human Activity Recognition from Unstructured Feature Points". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* - Chapter 4;

- Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori (2018a). "Object Level Visual Reasoning in Videos". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* - Chapter 5;

- Fabien Baradel, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf (2020a). "CoPhy: Counterfactual Learning of Physical Dynamics". In: *Proceedings of the International Conference on Learning Representations (ICLR)* - (spotlight presentation) - Chapter 6.

We voluntary omit the following papers in this manuscript:

- Fabien Baradel, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf (2020b). "CoPhy++: Counterfactual Learning of Physical Dynamics from Visual Input". In: *to be submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*;

- Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid (2019). "Learning Video Representations using Contrastive Bidirectional Transformer". In: *arXiv preprint arXiv:1906.05743*.

The first one corresponds of an journal extension of the work presented in Chapter 5 and will be submitted in the next weeks. The second one corresponds to work that has been conducted during an internship at Google and is under submission.

Software and dataset contributions — The work conducted in this thesis has led to the following list of software and released datasets:

- **Glimpse Clouds**: The code for training the model and evaluating on the validation and test set is released as part of the project presented in Chapter 4: `https://github.com/fabienbaradel/glimpse_clouds`

- **Object level visual reasoning in videos:** The code for training and evaluating the model on the test set is released as well as the pre-trained model weights as part of the project presented in Chapter 5: `https://github.com/fabienbaradel/object_level_visual_reasoning`. We have extracted extra information for VLOG (Fouhey et al. 2018) and EPIC (Damen et al. 2018) datasets. For every image in each video we have run a pre-trained Mask-RCNN (He et al. 2017) for detecting the objects present in the scene as well as their spatial location (bounding box) and the estimated pixel mask. This data is available on the project page.

- **CoPhy**: The benchmark composed of 3 datasets, the code including the dataloaders for each of the dataset, the scripts for training and evaluating our model are released as part of the project presented in Chapter 6: `https://github.com/fabienbaradel/cophy`.

Chapter 2

RELATED WORKS

Contents

In this thesis, we are interested in learning meaningful representations for visual data with a focus on videos. First, in Section 2.1 we give an overview of the main concepts used for representation learning in Machine Learning (ML) with a focus on Deep Learning (DL). Second, Section 2.2 we present the origin of Computer Vision (CV) and the different strategies for extracting visual representations.

2.1 Background in Machine Learning

In this section, we review the different ways used in ML for representation learning from supervised learning to unsupervised learning. We also describe the link between ML and causal inference and its possible applications in CV.

2.1.1 Supervised learning

In short, supervised learning implied training a system to automatically predict an output value to an input, given a set of labeled examples. If the output value

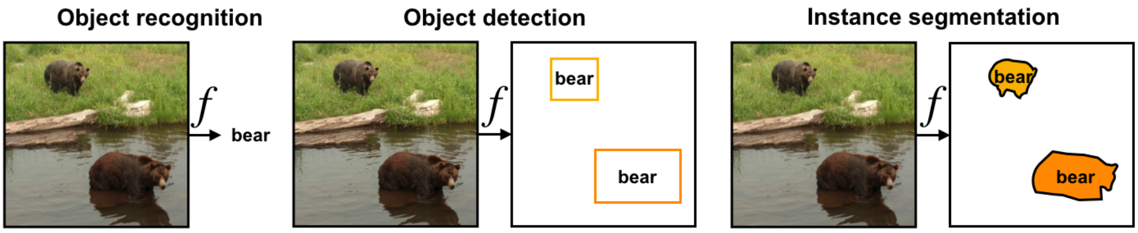


Figure 2.1 – **Examples of supervised learning problems.** (Left) *Object recognition*: The goal is to predict which object is present in the input image. — (Middle) *Object detection*: The goal is to detect objects by predicting the bounding box locations as well as the object categories. — (Right) *Instance segmentation*: This task goes beyond the object detection task, ones need to predict the pixel mask for each detected object.

is a *continuous* quantity then this is a *regression* problem. Otherwise if the output value is *discrete* class label, this is a *classification* problem. In this section we restrain our case to the *classification* task however the explanations are easily transferable to the *regression* task.

Different types of supervised learning problems in computer vision can be visualized in Figure 2.1.

Problem formulation — We assume $\boldsymbol{x} \in \mathcal{X}$ and $y \in \mathcal{Y}$ respectively as the input and the output to our system. $\mathcal{X}$ and $\mathcal{Y}$ represent respectively the *input* and *output* space. For example, in the case of object recognition $\boldsymbol{x}$ is an image and y is the object category index whose values are ranging from 1 to C where C is to the number of classes. We assume a training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N} \subset \mathcal{X} \times \mathcal{Y}$ composed of N training samples. The goal is to learn a mapping $f \in \mathcal{F}$ that can correctly predict the label y given the input $\boldsymbol{x}$. The predicted label denoted $\hat{y}$ produced by the system f is obtained using the following mapping:

$$f(\boldsymbol{x}) = \hat{\boldsymbol{y}} \tag{2.1}$$

$$\hat{y} = \arg\max_{c} \hat{\boldsymbol{y}}^c \tag{2.2}$$

where $\hat{\boldsymbol{y}}$ is a C-dimensional prediction vector which can be assimilated to a probability distribution over the set of possible classes such that $\sum_{c=1}^{C} \hat{\boldsymbol{y}}^c = 1$ and $\boldsymbol{y}^c > 0 \;\forall c = 1 \ldots C$. The value $\hat{\boldsymbol{y}}^c$ corresponds to the probability that the input $\boldsymbol{x}$ belongs to the c-th category.

Function f is learned using the training examples provided by $\mathcal{D}$, with the underlying goal of generalizing to unseen examples.

Neural networks — We restrict our mapping function f to *feedforward neural networks* since all the approaches proposed in this manuscript belong to this category. Neural networks are composed of many different functions or layers,

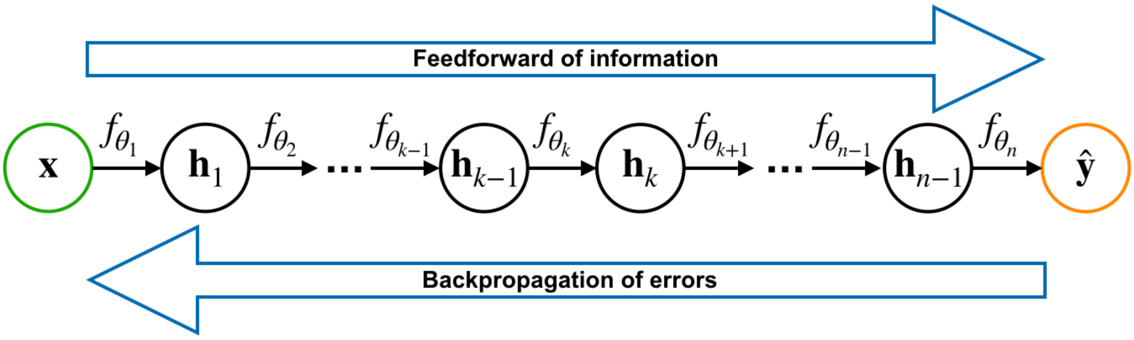


Figure 2.2 – **Feedforward neural networks as directed acyclic graphs.** A mapping function f which is a feedforward neural network can be composed of n layers or functions. The green node corresponds to the input while the orange node is the output. The information flows feedforward (from left to right) for producing an output given an input. And the system is trained by backpropagating the error in a backward manner (from right to left).

where each layer is itself a neural network. The term *feedfoward* means that the information flows strictly in a forward direction from the input to the output such as shown in Figure 2.2.

The mapping function f is composed of n layers and is reparameterized such that:

$$f(\boldsymbol{x}) = f_\theta(\boldsymbol{x}) = f_{\theta_n}(f_{\theta_{n-1}}(\dots f_{\theta_k}(\dots f_{\theta_1}(\boldsymbol{x})\dots)\dots)) \tag{2.3}$$

$$\boldsymbol{h}_k = f_{\theta_k}(\boldsymbol{h}_{k-1}) \tag{2.4}$$

where $\theta = \{\theta_1, \dots, \theta_n\} \in \Theta$ is the set of trainable parameters of f, Θ is the parameters space and f_k is the k-th layer whose input, output and trainable parameters are respectively $\boldsymbol{h}_k$, $\boldsymbol{h}_{k+1}$, θ_k. We present in pages 14 and 16 a set of commonly used neural network layers and (activation) functions.

Since the system f is parametrized by the trainable parameters θ (assuming the hyper-parameters of each layer fixed) the space of all possible mappings $\mathcal{F}$ is restricted to the space of all possible parameters denoted Θ. Hence in the rest of this section we use f_θ for denoting the mapping function.

A feedfoward neural network f_θ can be seen as acyclic directed graph from an information flow point of view such as shown in Figure 2.2. It means that for producing the output we need to iterate through each layer.

Optimization problem — For learning the function f_θ in Θ we use a loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ defined on a data point which corresponds to the cost of predicting $\hat{y}$ when the label is actually y.

A commonly used function in classification is the cross-entropy (LeCun et al. 1997):

$$\mathcal{L}_{\mathrm{CE}}(\hat{\boldsymbol{y}}, y) = -\sum_{c=1}^{C} \boldsymbol{y}^{c} \log \hat{\boldsymbol{y}} \tag{2.5}$$

, $\boldsymbol{y}$ is a one-hot vector representation of the groundtruth class such that $\boldsymbol{y} \in \{0,1\}^C$ and $\sum_{c=1}^{C} \boldsymbol{y}^c$.

We follow the principle of Empirical Risk Minimization (ERM) for learning the best system f where the *risk* is defined as the expectation of the loss function $\mathcal{L}$. Since we do not know the joint distribution of the data points we cannot compute the *true risk*, instead, we minimize the *empirical risk* by averaging the output of the loss function $\mathcal{L}$ on the training set $\mathcal{D}$. Hence, the optimization problem for obtaining the set of best parameters θ^* is of the general form

$$\mathcal{J}(\theta) = \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \mathcal{L}(f_\theta(\boldsymbol{x}), y) + \mathcal{R}(\theta) \tag{2.6}$$

$$\theta^* = \arg\min_{\theta\in\Theta} \mathcal{J}(\theta) \tag{2.7}$$

where $\mathcal{J}$ is the objective function composed of the loss function $\mathcal{J}$ and a $\mathcal{R}$ a regularization term.

Solving this optimization problem is not trivial since it is a non-convex problem due to the usage of non-linear functions in the mapping function (more details in 14). We employ the Stochastic Gradient Descent (SGD) (Bottou 1991) and the backpropagation rule (Rumelhart et al. 1986) for estimating the parameters such as described below.

Stochastic gradient descent — The standard way to minimize an objective function on the training set $\mathcal{D}$ for the case of neural network is to use SGD (Bottou 1991). This implies that f_θ should be differentiable with respect to every parameter in θ.

In short, the method corresponds to compute the gradient of the objective function $\frac{\partial \mathcal{J}(\theta)}{\partial \theta}$ on the training set $\mathcal{D}$ and update the parameters in the oppositive direction of the gradient given the current state of the parameters.

However computing the gradient of the objective function can be quite inefficient if the number of training examples is large. To overcome this problem, one solution is to use a variant called the *mini-batch* SGD which consists of replacing the true gradient of the objective by an estimation $\frac{\partial \hat{\mathcal{J}}(\theta)}{\partial \theta}$ computed from a set of training examples $\mathcal{S}$ randomly sampled from the training set $\mathcal{D}$ such that:

$$\frac{\partial J(\theta)}{\partial \theta} \approx \frac{\partial \hat{\mathcal{J}}(\theta)}{\partial \theta} = \frac{\partial}{\partial \theta}\left[\frac{1}{|\mathcal{S}|} \sum_{(\boldsymbol{x},y)\in\mathcal{S}} \mathcal{L}(f_\theta(\boldsymbol{x}), y) + \mathcal{R}(\theta)\right] \tag{2.8}$$

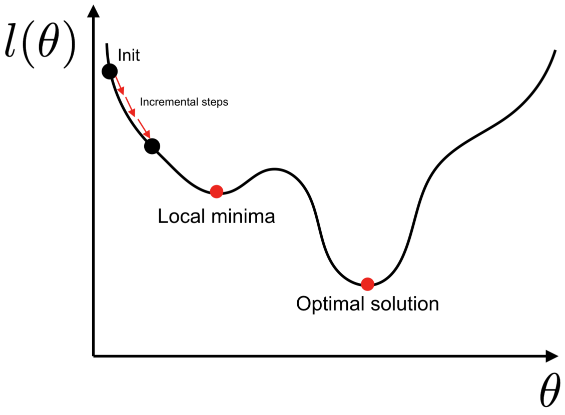


Figure 2.3 – **Gradient descent: toy example.** This graphs shows the iterative process of the gradient descent. There is a local minimum where the iterative process can get stuck, since it is initialization-dependent. The optimal solution is not guaranteed to be found.

For making sure that $\frac{\partial \hat{\mathcal{J}}(\theta)}{\partial \theta}$ is a good estimation of $\frac{\partial \mathcal{J}(\theta)}{\partial \theta}$, one need to enforce that the set $\mathcal{S}$ is representative of the training set $\mathcal{D}$. This can be done by enforcing $\mathcal{S}$ to contain enough number of training examples.

Then the update the parameters in the opposite direction of the gradient is done by following this updating rule:

$$\theta \leftarrow \theta - \eta \frac{\partial \hat{\mathcal{J}}(\theta)}{\partial \theta} \tag{2.9}$$

where is η is the learning rate.

Mini-batch SGD is an iterative process which starts with a random initilization of the trainable parameters in θ. An *epoch* consists in running over all mini-batches once. This training is an iterative process such that epochs are repeated until convergence of the descent on the training set. The learning rate η and the initilization of the trainable parameters are critical for the training procedure. If the learning rate is too small, the training procedure can be slow or/and gets stuck in a local minima due to a bad initialization of the trainable parameters. On the opposite, if the learning rate is too high the optimization may never converge. Figure 2.3 shows an example of iterative process of the gradient descent algorithm.

Since we want to generalize to unseen elements (i.e. outside of the training examples), most of the time we keep a validation set, which is used for stopping the iterative training procedure as soon as the metric computed on the validation set is not improving or reaching a plateau.

Backpropagation rule — Training the mapping function f_θ using mini-batch SGD involves computing the gradient $\frac{\partial \hat{\mathcal{J}}(\theta)}{\partial \theta}$ such as demonstrated in Equation 2.9.

However computing $\frac{\mathcal{L}(f_\theta(\boldsymbol{x}),y)}{\partial\theta}$, which is a subpart of the overall gradient could be cumbersome as soon as the mapping function f_θ becomes very deep.

One way to alleviate this issue is to use the chain rule and to compute the gradient one layer at a time such that:

$$\frac{\partial\mathcal{L}(f_\theta(\boldsymbol{x}),y)}{\partial\theta_l} = \frac{\partial\mathcal{L}(f_\theta(\boldsymbol{x}),y)}{\partial\theta_n}\left(\prod_{k=l+1}^{n}\frac{\partial\boldsymbol{h}_k}{\partial\boldsymbol{h}_{k-1}}\right)\frac{\partial\boldsymbol{h}_l}{\partial\theta_l} \tag{2.10}$$

This principle is known as the *backpropagation rule* (Rumelhart et al. 1986) and consists of an iterative backward from the last layer such as shown in Figure 2.2.

We now have introduced all the key components used for training neural network system in a supervised manner. When all parameters of a network architecture can be updated with the same instance of gradient descent, the network is said to be trained *end-to-end*.

A brief history of Artificial Neural Network — The first Artificial Neural Network (ANN), called Perceptron, is introduced by Rosenblatt (1957). Originally it consists of a single layer network that has n input values and a single output value. The input values are multiplied by their associated parameters and summed. Finally a activation function is applied for producing the final output value.

Fukushima (1980) introduce a hierarchical multilayered neural network *Neocognitron* which is the first application of neural network for the handwritten character recognition. Finally the first notable results appear in the fields of speech recognition (Lang et al. 1988) and are mainly due to the use of backpropagation (Rumelhart et al. 1986).

Common neural network layers — In CV, the first notable results are achieved on the task of handwritten character recognition by LeCun et al. (1997). They propose the first successful CNN called Graph Transformer Network (GTN) (LeCun et al. 1997) shown in Figure 2.4. A CNN is a feedforward neural network composed of at least one convolutional layer. We give a summary of common layers/functions that are employed in CNN architecture which takes visual data as input:

- *Convolutional* layers are a key component of CNN models. It consists of applying a convolutional operation with a learnable spatial kernels. The input $\boldsymbol{h}_{k-1}$ and output $\boldsymbol{h}_k$ are feature maps respectively of size $m_1^{k-1} \times m_2^{k-1} \times m_1^{k-1}$ and $m_1^k \times m_2^k \times m_1^k$. A convolutional layer consists of a bank of m_1 filters and each filter detects a particular spatial feature at every location. The i-th output feature map denoted $\boldsymbol{h}_i^k$ is given by

$$\boldsymbol{h}_k^i = B_i^k + \sum_{j=1}^{m_1^{k-1}} K_{ij}^k * \boldsymbol{h}_{k-1}^i \tag{2.11}$$

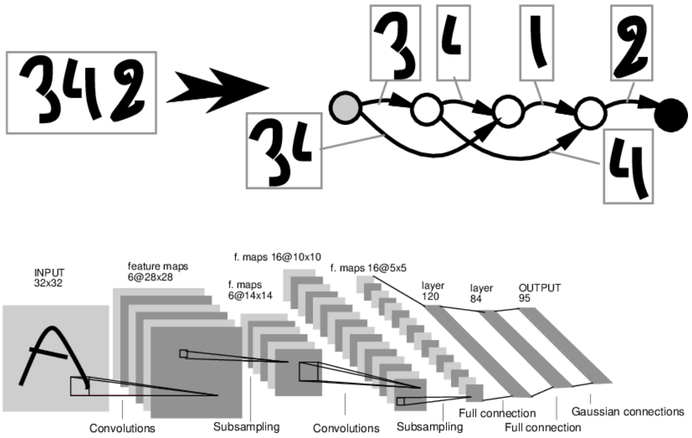


Figure 2.4 – **Graph Transformer Network architecture.** The Convolutional Neural Network (CNN) proposed by LeCun et al. (1997) for the task of handwritten character recognition is composed of convolutions, subsampling and full connections operations.
Figure reproduced from (LeCun et al. 1997).

where $*$ is the convolution operator, B_i^k is a learnable bias matrix and K_{ij}^k is the learnable spatial kernel filter of connecting the j-th feature map of $\boldsymbol{h}_{k-1}$ with the i-th feature map of $\boldsymbol{h}_k$. The success of this convolutional layer is mainly due to the weight sharing strategy employed for learning the convolution kernels (LeCun et al. 1989).

- *Pooling* layers consist of subsampling the information from feature maps. This can be done at any stage k of the neural network by employing a mean or max operator given a spatial extend F^k and a stride S^k. Hence the pooling layer that takes as input a feature map of size $m_1^{k-1} \times m_2^{k-1} \times m_1^{k-1}$ produces a feature map of size $m_1^k \times m_2^k \times m_1^k$ with

$$m_1^k = m_1^{k_1} \tag{2.12}$$

$$m_2^k = (m_2^{k_1} - F^k)/S^k + 1 \tag{2.13}$$

$$m_3^k = (m_3^{k_1} - F^k)/S^k + 1 \tag{2.14}$$

It is often employed after a convolutional layer to reduce the spatial dimension of a feature map.

- *Activation* functions denoted σ, allow to a introduce non-linearity (LeCun et al. 1998) into neural network. Given an activation function σ the output at stage k of a neural network is given by

$$\boldsymbol{h}_k = \sigma(\boldsymbol{h}_{k-1}) \tag{2.15}$$

It has been empirically demonstrated that this helps during the training procedure. The Rectified Linear Unit (ReLU) operation is a common non-linear activation function used in modern architecture.

- *Fully-connected* layers are an extension of the Perceptron (Rosenblatt 1957). They take as input and predict as output multiple values. This layer applies a linear transformation on the input a vector $\boldsymbol{h}_{k-1}$ of dimension m^{k-1} for producing an output vector $\boldsymbol{h}_k$ of dimension m^k such as

  $$\boldsymbol{h}_k = W_k \boldsymbol{h}_{k-1} + b_k \tag{2.16}$$

  where W_k is a matrix and b_k is an bias parameter. Stacking multiple fully-connected layers is known as *multi-layer perceptron*. In general, we employ activation functions after fully-connected layers.

Recurrent neural networks — For the moment we described systems that takes as input spatial signals such as images. However in this manuscript we are also interested in working with sequential data since we want to extract information from videos. In this case the input data $\boldsymbol{x} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_t, \dots, \boldsymbol{x}_T)$ is a sequence composed of T elements.

Other types of neural network layers are proposed for tackling sequential data such as Recurrent Neural Network (RNN) (Jordan 1986) and Long-Short Term Memory (LSTM) (Hochreiter et al. 1997). RNNs employ *hidden vector* denoted $\boldsymbol{h}$ which is recursively updated at each timestep using the element from the input, and from which the output $\boldsymbol{v}$ is predicted,

$$\boldsymbol{h}_t = \sigma_h(W_h \boldsymbol{x}_t + U_h \boldsymbol{h}_{t-1} + b_h) \tag{2.17}$$

$$\boldsymbol{v}_t = \sigma_y(W_y \boldsymbol{h}_t + b_y) \tag{2.18}$$

where σ_h and σ_y are activations function and W_h, W_y, b_h, b_y, U are parameters matrixes and biases.

RNNs suffer from the vanishing and exploding gradient problem (Bengio et al. 1994) especially when dealing with long sequences. This is due to the explosion (or vanishing) of the product of derivatives during the computation of the gradient using the backpropagation through time. One solution is to use LSTMs which employ a gating mechanism. This mechanism allows the gradient to backpropagate more easily essentially by smoothing out the update of the hidden vector h at each timestep by using activation functions. Gated Recurrent Unit (GRU) is also another option which employs a simplified version of the LSTM gating mechanism.

Limitations — While supervised learning in CV has shown great success in many tasks ranging from object recognition to object detection, this type of learning raise several issues and constraints.

In many cases, having enough labeled data is problematic. Annotating large scale datasets is expensive and cumbersome. While it can be considered as a reasonable task at an image-level, adding pixel annotations such as bounding boxes or instance segmentation is extremely time-consuming and prone to error depending on the annotator. (Dias et al. 2019) For example, labeling a single image with pixel-level annotations in the COCO-Stuff takes 19 minutes Caesar et al. 2018. Thus annotating the 164'000 images of this dataset would take around 53'000 hours.

Since it is easy to have access to *unannoted* visual content for free, we wish to construct a representation without having to deploy large-scale annotations. Motivations for learning systems without annotations comes from the fact that humans learn how the world works mainly by observations (Gopnik et al. 2000). In the next section we present alternatives to strong supervision.

2.1.2 Unsupervised learning

Unsupervised learning consists in learning meaningful representations of data without having access to manual annotations. We review different unsupervised methods and introduce recent self-supervised methods.

Problem formulation — Our training set is simply $\mathcal{D} = \{\boldsymbol{x}_i\}_{i=1}^N \subset \mathcal{X}$, composed of N training samples.Space $\mathcal{X} \subset \mathbb{R}^p$ being assumed to be a high-dimensional space, the underlying goal of an unsupervised method is to learn a function f that maps an input to a *representation* $\boldsymbol{z}$ lying in a lower-dimensional space $\mathcal{Z} \subset \mathbb{R}^k$ where inputs with similar semantical meanings are close to each other. In some sense we want to infer the underlying structure of $\mathcal{X}$ with $k \ll p$.

Pre-training — We are interested in unsupervised learning method as a *pre-training* strategy. The term *pre-training* means that we want to learn a representation $\boldsymbol{z}$ in an unsupervised manner with the ultimate goal that this representation could be helpful for solving supervised task (also called *downstream* tasks) with a limited number of annotated examples.

Clustering analysis — A common way for dealing with unannoted data is to employ clustering methods (Jolliffe 2005; Hartigan et al. 1979; McLachlan et al. 2008), which is the process of grouping similar entities together. Clustering is a traditional task in unsupervised learning, but classically it is used in contexts, where the discovery of clusters is the goal itself, as opposed to finding a suitable representation for further processing, which is our problem. Clustering can aso be used for representation learning. Although not widely used, we will give and example based on the popular K-means algorithm.

The key ingredient is a distance function and one common choice is to employ the Euclidean distance:

$$d(\boldsymbol{x}_i, \boldsymbol{x}_{i'}) = \sqrt{\sum_{j=1}^{p} (\boldsymbol{x}_{ij} - \boldsymbol{x}_{i'j})^2} \tag{2.19}$$

K-means algorithm (Likas et al. 2003), that aims at providing a *vector quantization* for each data point. This also corresponds to the final representation $\boldsymbol{z}$ provided by this method with the following constraint in $\boldsymbol{z}$, $\boldsymbol{z}_k \in \{0, 1\}$ and $\sum_{k=1}^{K} \boldsymbol{z}_k = 1$. $\boldsymbol{z}_{ik}$ is a binary indicator set to 1 if the i^{th} data point belongs to the k^{th} cluster.

The goal is to assign each data point to a cluster, among K clusters, which involves to learn the cluster centroids μ_k for k in $1 \ldots K$. The loss function $\mathcal{L}$ consists of computing the distance between a data point $\boldsymbol{x}$ and its assigned centroid such as:

$$\mathcal{L}(\boldsymbol{x}, \mu) \sum_{k=1}^{K} \boldsymbol{z}_k d(\boldsymbol{x}, \mu_k) \tag{2.20}$$

where $\mu = \{\mu_1, \ldots, \mu_K\} \in \Pi$ is the set of cluster centroids and Π represents their space.

And the optimization problem is defined by the objective function $\mathcal{J}$ defined as followed:

$$\mathcal{J}(\mu) = \sum_{\boldsymbol{x} \in \mathcal{D}} \mathcal{L}(\boldsymbol{x}, \mu) \tag{2.21}$$

$$\mu^* = \arg\min_{\mu \in \Pi} \mathcal{J}(\theta) \tag{2.22}$$

where the cluster centroids μ are computed by averaging the vectors belonging to them cluster.

At initialization, centroids are randomly set. Then, the minimization process consists of two steps that are repeated until convergence of the algorithm. First, each data point is assigned to its nearest cluster centroid. Second, each cluster centroid is updated by averaging vectors assigned to the cluster.

The final representation $\boldsymbol{z}$ obtained after using the K-means algorithm is quite limited and lack expressivity since it is a one-hot vector. This is the only part of the model which is learned during the optimization of the objective function since the computation of the cluster centroid is only parametrized by $\boldsymbol{z}$. Moreover fine-grained information which is important for visual data can easily be lost since no intermediate representation are learned. This lack of expressivity is not solved with more advanced clustering methods such as Gaussian Mixture Model (Reynolds 2009), Hierarchical k-means (Moore 2001) or spectral clustering (Von Luxburg 2007) that all require low-dimensional input for working properly.

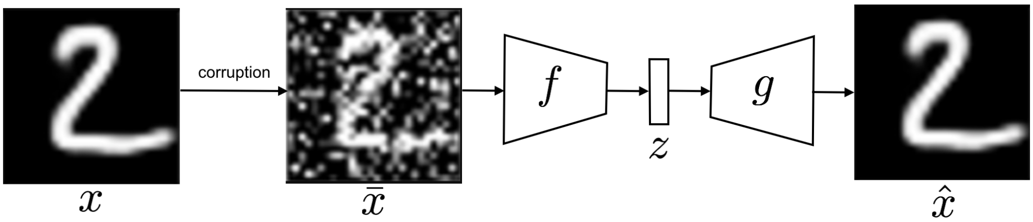


Figure 2.5 – **Denoising autoencoder.** A percentage of pixels' input are perturbated and feed to the autoencoder that needs to generate the non-perturbated input.
Figure reproduced from (Vincent et al. 2008).

Autoencoder — An autoencoder (Hinton et al. 1994) is a feedforward neural network for encoding data in unsupervised manner. The goal is to build a low-dimensional representation $\boldsymbol{z}$ of data called the *code* that contains all the important high-dimensional information of $\boldsymbol{x}$. To do so, the output is constrained to be equal to the input while trying to compress as much as possible the intermediate representation within the neural network. Making sure that the *code* is low-dimensional enough is an important assumption otherwise the autoencoder only consists of an identify mapping.

This method is composed of two networks: an *encoder* and a *decoder*. The *encoder*, denoted f, compresses the input $\boldsymbol{x}$ into a low-dimensional vector representation denoted $\boldsymbol{z}$ (the *compressed* or *hidden* representation, also known as *code*). The *decoder* g takes as input $\boldsymbol{z}$ and should re-generate the initial input $\boldsymbol{x}$ such as:

$$\boldsymbol{z} = f_{\theta_1}(\boldsymbol{x}) \tag{2.23}$$

$$\hat{\boldsymbol{x}} = g_{\theta_2}(\boldsymbol{z}) \tag{2.24}$$

where $\theta = \{\theta_1, \theta_2\}$ is the set of trainable parameters respectively the encoder and the decoder. We want the dimension of code z to be as small as possible, while z being as representative as possible of the input x.

The loss function $\mathcal{L}$ consists of penalizing reconstruction error,

$$\mathcal{L}(\boldsymbol{x}) = \sum_{i=1}^{N} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\| \tag{2.25}$$

We employ an objective function $\mathcal{J}$ similar to the one presented in Equation 2.6 and solve the optimization problem the same way as described in Section 2.1.1.

Some works extend the expressivity of autoencoder by proposing variants (Ng et al. 2011; Rifai et al. 2011; Kingma et al. 2014; Vincent et al. 2008). Vincent et al. (2008) propose the denoising autoencoder which consists in perturbing a percentage of the input x and still encode and decode the input such that we can generate the initial non-perturbated input x. An illustration of the procedure is shown in Figure 2.5. Similarly, the contractive autoencoder (Rifai et al. 2011) is proposed by forcing the model to be robust to slight variations of the input values. They propose a regularization term $\mathcal{R}$ that imposes strong constraints on the learnable parameters. Ng et al. (2011) focus on encouraging sparsity within the autoencoder by reducing the number of units and exploiting the Kullback-Leiber (KL) divergence. Finally, Kingma et al. (2014) propose to build a generative model using the autoencoder framework. They incorporate strong assumptions on the latent variables z by using a variational approach such that the latent variable should follow a prior distribution. This encourages independance of the values of the code, and often leads to the discovery of semantically meaningful representations.

Pre-training with autoencoders — Vincent et al. (2010) show that training an autoencoder as a *pre-training* strategy can be helpful. For solving the *downstream tasks* they initialize the parameters of the mapping function of the supervised system by using the parameters from the encoder. They show better results following this two-stage strategy compared to start solving the downstream tasks from random weights. While extremely encouraging these results have not been extended to real world images.

However this reconstructing technique demonstrates its effectiveness on discrete data. Devlin et al. (2018) propose a BERT (Bidirectional Encoder Representations from Transformers) in the field of Natural Language Processing (NLP), which can be seen as a denoising autoencoder (Vincent et al. 2008) (also called *masked* autoencoder) from a sequential discrete signal. The authors propose to mask a percentage (15%) of the input sentence and train the autoencoder at reconstructing the missing words given the context of the sentence. They show that training on a large-scale unannoted corpus is an effective pre-training strategy for solving downstream NLP tasks ranging from sentimental analysis to text summarization.

Data points in NLP are discrete since most of the works (Devlin et al. 2018; Vaswani et al. 2017) assume a pre-defined dictionary of all possible words. Hence the unsupervised task of *reconstruction* can be seen as a *prediction* task. Indeed given a masked word we need to find the more appropriate word over a set of possible words (usually around 30'000). This could be a potential explanation of why this unsupervised strategy is effective on discrete signal but not on continuous signal such as visual data, at least not when applied trivially.

Self-supervised learning — In CV, recent works in unsupervised learning (Gidaris et al. 2018; Hénaff et al. 2019; Caron et al. 2018; Novotny et al. 2018a; Novotny et al. 2018b) follow this strategy of *predicting* instead of *reconstructing*. This kind of representation learning is called *self-supervised learning* and consists of using the naturally available relevant context and embedded data as supervisory signal.

For example, Gidaris et al. (2018) propose to train the parameters of a CNN function trained on predicting a rotation randomly applied to an input image. While being simple, the authors show that this *pretext task* learns low-level and mid-level features as good as learning them in a fully-supervised way from a large-scale annotated dataset (e.g. Imagenet (Krizhevsky et al. 2012)). Caron et al. (2018) adapt clustering methods to end-to-end training by jointly learning the parameters of a CNN and the cluster assignments of the resulting features. They use a K-means procedure for grouping features and use output vector quantization as supervision signal. Oord et al. (2018) propose to predict the future in latent space in an autoregressive manner by using a probabilistic contrastive loss. This way they ensure to capture semantical information that is useful to predict the future. Following a similar strategy, Hjelm et al. (2018) introduce DeepInfoMax that consists in maximizing the mutual information between the input and the output of the neural network.

These methods show great results on downstream tasks such as image classification with limited number of training examples per class. However as soon as the number of training examples increases the usefulness of the pre-training strategy gets limited or even null.

Moreover using unsupervised algorithms does not solve the problem of learning dataset biases that can still exists even in unannoted data. One way to deal with such issue is to introduce causal inference that we are going to describe in the coming section.

2.1.3 Towards counterfactual reasoning

Reasoning is an essential ability of intelligent agents that enables them to understand causal relationships and to leverage this understanding to anticipate the future and act accordingly. In this section, we discuss the importance of causal inference in machine learning. We introduce a few important notions and focus on counterfactual prediction.

Human thinking and causal reasoning — Human cognitive processes can be assigned to two distinct systems as described by Kahneman (2011) (winner of the Nobel prize in 2002). *System 1* is fast, automatic and unconscious. It can detect and localize objects, understands simple sentences, solves very simple numerical problesm like $1 + 1$, but it is also prone to errors due to its continuously generated assessments. *System 2* is slow, effortful, logical, conscious. It can count the number of cars in a street, performs logical reasoning, solves more complex numerical problems like 121×28, generates questions and answers them. System 2 is more reliable but requires effort. These two systems are used for solving different types of cognitive tasks such as shown in Figure 2.6.

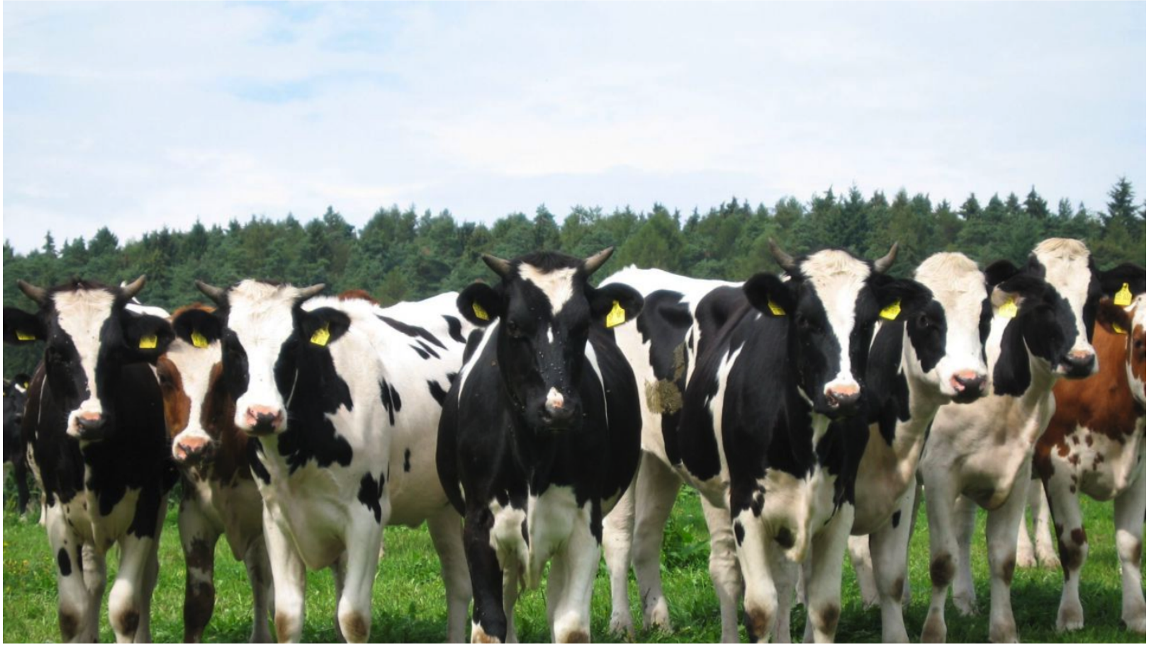


Figure 2.6 – **System 1 and System 2 in Computer Vision.** (Fast) The question *"Is there a cow in the picture?"* is straightforward to answer. **system 1** handles this question and gives a positive answer almost automatically. — (Slow) Answering the question *"How many cows are present in the image?"* requires identifying each cow despite occlusions and counting the total number of cows. This requires attention and can be seen as a multi-steps procedure. Such a task is solved by **System 2** and requires effort. However, verifying that the answer is **NOT** equal to 5 Billion is, again, done fast and by System 1.

Broadly stated, current DL methods are able to reach good performance on tasks tackled by System 1 by humans. Up to a certain performance, CNNs may provide a good decomposition of a scene into visual entities. However reasoning and planning is more difficult for current DL methods. At the moment it is difficult to manipulate semantic concepts that can be recombined combinatorially. Such behavior might be important for being able to solve issues related to out-of-distribution and transfer. In this manuscript we believe that developing models for high-level cognition requires tackling compositionality and causality.

Causal reasoning gained mainstream attention relatively recently in the ML community (Lopez-Paz et al. 2017a; Lopez-Paz et al. 2017b; Kocaoglu et al. 2018; Rojas-Carulla et al. 2018; Mooij et al. 2016; Schölkopf et al. 2012), due to limitations of statistical learning becoming increasingly apparent such as discussed earlier (Pearl 2018; Lake et al. 2017; Scholkopf 2019). The hope is to introduce causal inference to build more robust models able to *generalize* (Scholkopf 2019). Moreover, the ultimate goal is to move from machine learning to machine reasoning and let machines be able to discover causal concepts and *think* in an imagined space. In the rest of this section, we focus on causal inference and highlight important notions that can bring more reasoning abilities for machine learning systems.

Seeing versus doing — We assume a set of training examples $\mathcal{D} = \{(x_i, y_i, z_i)\}_{i=1}^{N}$ composed of N training examples and the respective associated random variables are denoted as upper case letters X, Y and Z (as opposed to lower case realizations). The training examples in $\mathcal{D}$ are drawn from the joint probability distribution $P(X, Y, Z)$.

In Section 2.1.1, we were interested in modeling Y given $X = x$ which corresponds to modeling the conditional distribution $P(Y|X = x)$. The mapping function f defined in Equation 2.1 that predicts an output y given an input x is an estimation of the conditional expectation $\mathbb{E}(Y|X = x)$ using the training examples $\mathcal{D}$. This procedure is *observational* and consists of computing statistics from data that we *see*. While being very useful for solving many tasks ranging from object detection to human pose estimation it has some limitations such as often learning the dataset bias rather than causal relationships and reasoning.

In causal inference we are interested in modeling the *interventional* distribution $P(Y|do(X = x))$ which is (in general) different from the *observational* distribution $P(Y|X = x)$. The main difference between these two conditional distributions is that for $P(Y|do(X = x))$ we **set** the value of X to x in the data generating process such as denoted by the *do*-operator while for $P(Y|X = x)$ we **observe** variable X at value Y. The interventional distribution differs from a observational one mainly because the samples does not comes from the same generating process.

We illustrate this with a simple example. If we interested in estimating the future GDP (gross domestic product) of different countries given the current GDP and some features in the form of key economic observations, we perform classical forecasting, which can be statically modeled as a conditional distribution estimated from observed data, hoping that our model is capable of generalizing to unseen realizations of the same type of data. On the other hand, if we were interested in verifying what the future GDP of the UK were given in the hypothetical case where *the UK hadn't left he European Union in 2020*, we can't learn a model from observed data, since this case has never been observed before. This case requires the design of a structural model of the effects of the do-operator on key random variables.

A concrete toy example has been proposed by Lopez-Paz (2016), given as follows. We consider a joint distribution $P(X, Y, Z)$ described by the following data generation process,

$$z_i \sim \mathcal{N}(0, 1) \tag{2.26}$$

$$x_i \leftarrow 5z_i \tag{2.27}$$

$$y_i \leftarrow x_i + 5z_i \tag{2.28}$$

We can simulate thousands data points and estimate $\mathbb{E}(Y|X = 1) \approx 2$ by observing data in a passive way. Now we *set* X to 1, such that we should consider the joint distribution $P_{do(X=1)}(X, Y, Z)$. This can be done by intervening on the data generation process as follows

$$z_i \sim \mathcal{N}(0, 1) \tag{2.29}$$

$$x_i \leftarrow 1 \tag{2.30}$$

$$y_i \leftarrow x_i + 5z_i \tag{2.31}$$

We can draw thousands of data points and compute the mean for estimating that $\mathbb{E}(Y|do(X = 1)) \approx 1$, which does not correspond to the observational quantity.

In practice modeling this quantity is of high interest for identifying the relations between variables. For example, in medicine, if x is a treatment and y is the outcome, it is of high interest to understand what is the impact of a specific treatment. Randomized controlled trials can be conducted for generating data from the joint distribution $p_{do(x)}(X, Y, Z)$. However very frequently such experiments cannot be performed in practice, for example due to ethical issues. Causal inference provides a way to answer interventional questions from observational data points.

Estimating the consequences of doing by seeing — One way to solve this problem is to make an assumption about the underlying *causal structure* of the data generation process. This type of information cannot be captured in the joint distribution; causal structure adds expressivity to the data generation mechanism.

A Structural Causal Model (SCM) $\mathcal{M}$ is represented by a triplet $< U, P(U), V, F >$ where:

- U is a set of *exogenous* variables of any types. An joint distribution $P(U)$ is defined over U.
- V is a set of *endogenous* variables. In our case we assume V to be composed of the three random variables X, Y and Z.
- F is a set of mapping functions. It gives the mapping of any endogenous variables $X \in V$ by
$$X = f_x(Pa(X), U_x)$$
where f_x is the mapping function of X, $Pa(X)$ is the set of endogenous variables for determining the value of X, and $U_x \in U$.

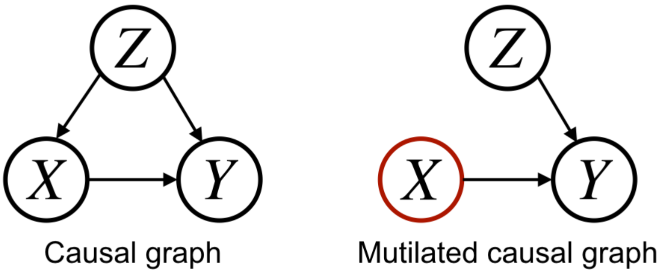


Figure 2.7 – **Directed Acyclic Graph:** (Left) A causal graph showing that both X and Z are causing Y; Z is a confounder of, both, X and Y. — (Right) A mutilated version of the causal graph shown in (Left). An intervention on X leads to removing the causal relationships from its ancestors, in this case Z, as X is now determined directly by the intervention does not depend on Z anymore. The data generation process is not the same as in the original graph in (left).

An *exogenous* variable is a variable whose value is determined outside the model while an *endogenous* variable is determined inside the model.

A SCM is associated to a causal graph $\mathcal{G}$ where each node represents a endogenous variables V, and each edge represents a causal relation that is present in the set of mapping functions F.

As an example, we show in Figure 2.7 (Left) a DAG representing the causal relationships $X \leftarrow Z \rightarrow Y \leftarrow X$. Here Z is a *confounder* variable which means that Z is a cause for, both, X and Y.

Intervening with the *do*-operator on a variable X corresponds to replacing its function f_X by the value x. The SCM model $\mathcal{M}$ after the intervention is the submodel denoted $\mathcal{M}_x$. In the causal graph representation that corresponds to cutting all the edges coming to the variable X such as shown in Figure 2.7 (Right). The effect of such an intervention on another endogenous variable, for instance Y, is denoted Y_x and corresponds to the interventional variant of Y in the submodel $\mathcal{M}_x$. The distribution of Y_x is our quantity of interest denoted $p(y|do(x))$. Pearl (2012) propose three rules of *do*-calculus to infer post-intervention distributions from observational data, by converting post-intervention distributions to observational distributions.

The *do*-calculus illustrates the power of causal inference, allowing us to reason, just by seeing, on the consequences of doing. Modeling intervention rather than association allows to answer *what if?* questions which is a necessary first step towards machine reasoning. However, humans go one step further by being able to imagine and retrospect potential outcome by expressing *counterfactual* conditions.

Expressing causality from counterfactuals — The philosopher Hume (1748) introduces the concept of counterfactuals as a way to express causal relationships. Answering counterfactual queries requires the existence of a twin/alternative world where everything is the same except the hypothetical intervention and its effects (Lopez-Paz 2016). Counterfactuals are the top level of the causal hierarchy proposed by (Pearl 2018) such as shown in Figure 2.8, and as such are seen as being a more difficult problem than the base problems in the hierarchy. In general, a query is a "counterfactual" if it contains an "if" portion that is untrue or unrealized.

Since the algorithmization of counterfactuals is an extension of the *do*-calculcus, the assumptions about the SCM still hold. Modeling counterfactual queries consists of modeling the probability distribution

$$p(y|x', y', z', do(x)).$$

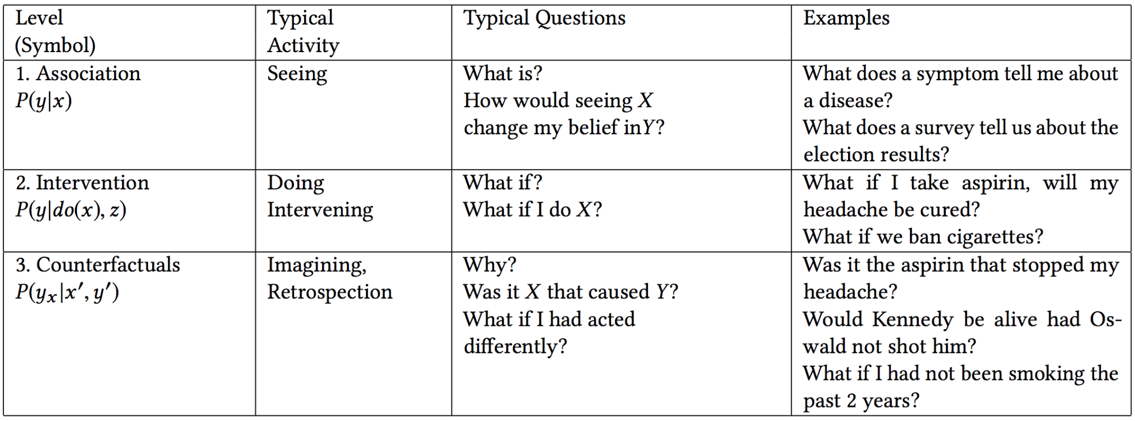
In other words, it consists of what would be Y if I had set $X = x$ given that I have observed (x', y', z'). It is important to notice that x', y' and z' are *observed* values while the quantities x and y are *unobserved* and belong to the alternative and hypothetical world. The distribution is thus conditioned on two different values for X, before and after intervention.

Answering a counterfactual query from observational data can be done using a three-step procedure (Balke et al. 1994; Tian et al. 2002):

- the **abduction** consists of updating the joint probability distribution over the endogenous variables $P(U)$ given the observations $o = (x', y', z')$ to obtain $P(u|o)$.
- the **action** corresponds to modifying the model $\mathcal{M}$ by the intervention $do(x)$ to obtain the model $M_x(x)$.
- the **prediction** step uses the modified model $\mathcal{M}_x$ as well as $P(u|o)$ to compute Y_x

Several extensions (Shpitser et al. 2009) are proposed to overcome the lack of complete knowledge of the causal model.

Limitations — The framework presented above offers an elegant solution for solving counterfactual queries, however it suffers from some criticisms as well. The definition itself of counterfactuals is not a concept approved by the entire causal inference community (Dawid 2000). This is mainly due to the fact that they are never observed so they are not empirically testable. Another source of criticism comes from the fact that we require a pre-defined structure of the causal world, in terms of machine learning, a handcrafted causal structure. While it is possible to express this statement for applications with low-dimensional data (Balke et al. 1994; Tian et al. 2002; Pearl et al. 2018) it seems difficult to scale up



Level (Symbol)	Typical Activity	Typical Questions	Examples
1. Association $P(y\|x)$	Seeing	What is? How would seeing X change my belief in$Y$?	What does a symptom tell me about a disease? What does a survey tell us about the election results?
2. Intervention $P(y\|do(x), z)$	Doing Intervening	What if? What if I do X?	What if I take aspirin, will my headache be cured? What if we ban cigarettes?
3. Counterfactuals $P(y_x\|x', y')$	Imagining, Retrospection	Why? Was it X that caused Y? What if I had acted differently?	Was it the aspirin that stopped my headache? Would Kennedy be alive had Oswald not shot him? What if I had not been smoking the past 2 years?

Figure 2.8 – **Causal hierarchy defnied by Judea Pearl.** The causal hierarchy is split into three categories: association, intervention and counterfactuals. The most high-level category is the counterfactuals one. Judea Pearl assesses that cause-effect relationships can only be expressed from this category.
Figure reproduced from (Pearl 2018).

this principle to high-dimensional data such as visual data (images and videos). Moreover, the concept of intervention is not clear for all fields of applications. In CV the list of possible interventions is long and complex. Interventions with the do-operator can be done directly on a pixel level, in feature space or at an object-level representation. Lopez-Paz et al. (2017a) suggest to perform interventions in the semantic space to align with works dealing with meaningful low-dimensional data (Rubin 1986). However, it is not clear how to define an *intervention* in a semantic space for visual data.

In our own work, presented in Chapter 6, we propose to supervise the *do*-operator as a first step to solve counterfactual from visual data by generating videos using a physical engine, where a scene is composed of objects. This allows to easily perform do-interventions on the positions of the objects and provide alternative world. On the upside, this allows to perform counterfactual queries on high-dimensional data, as images. However, on the downside, since we (for the moment) supervised the do-operator, this corresponds to performing statistics on observational data.

2.2 Visual Representations

In this section, we describe the strategies used in the last decade for extracting visual representations and their applications in associated active research topics.

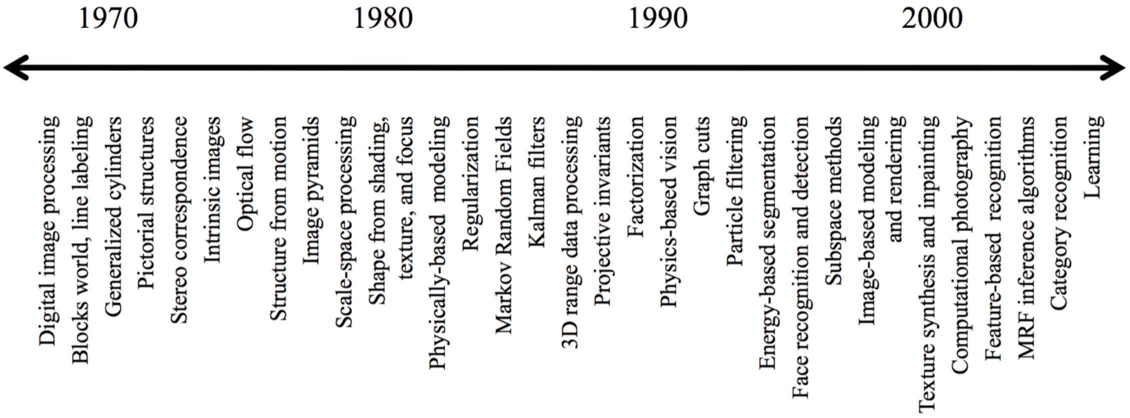


Figure 2.9 – **Timeline of active topis in CV.**
Figure reproduced from (Szeliski 2010).

2.2.1 Origin of Computer Vision

The first attempt to automatically extract information from an image dates back from 1963 by the Laurence Roberts who proposes in his PhD thesis (Roberts 1963) to infer the 3D structure of a cube from a picture. At that time pioneers (Seymour 1966; Marr 1982) in artificial intelligence thought that solving the vision problem would be an easy step toward more difficult tasks such as planning and high-level reasoning (Szeliski 2010). Seymour Papert conducts a summer vision project (Seymour 1966) with his undergraduate students with sub-goals ranging from foreground-background segmentation to scene analysis with simple non-overlapping objects. However as we will see in this manuscript, solving these tasks took decades. In the meantime, researchers focus on low-level processing tasks such as line labelling (Huffman 1971; Clowes 1971; Duda et al. 1972) and then move towards edge detection (Davis 1975; Canny 1986) with the underlying goal of simplifying the visual content of images.

Inspired by discoveries on the visual system from Hubel et al. (1959), David Marr proposes *bottom-up* approach for the vision pipeline. The first stage corresponds of using low-level image processing applied to 2D images leading to a *primal sketch* of the scene. Then a *2.3D sketch* is obtained by using binocular stereo. Finally the final representation is a *3D model* of the scene in a 3-dimensional map using structural analysis. Researchers (Huttenlocher 1987; Weiss 1988) get inspired from this paradigm for recognizing object in images either by fitting a corresponding 3D Computer-Aided Design (CAD) model based on image features (Huttenlocher 1987) or by using geometrical properties of the visible shape that are invariant to point of view (Weiss 1988).

However, geometry-based methods show difficulties due to illuminations Burns et al. (1993) and approaches move towards the use of appearance features. Sirovich et al. (1987) propose to decompose face images onto a low-dimensional space with PCA which is the first statistical approach for solving a vision task. This approach is improved by introducing the eigenfaces decomposition in 1991, a near-real-time system that can recognize faces using an Euclidean distance in the PCA space (Turk et al. 1991). The use of appearance features is then successfully applied to more tasks such as described in the next section. This quick summary of the origin of CV illustrates the general tendency leading towards the learning paradigm now used in the field. Figure 2.9 shows a historical timeline of active topics.

2.2.2 Object recognition

Humans recognize objects in images with little effort, no matter what is the camera view point, the size, scale and rotation of the object. They can even infer the category of an object which is partially occluded in the image. The goal of object recognition systems is to mimic this behavior which consists to categorize the object present in an image. To do so, object recognition systems need to construct *invariant*, *robust* and *discriminative* representations for each object.

Object recognition is a classification task that corresponds to assigning a predefined label to an input image. This is a standard task in CV which has been used for decades for juging the quality of image descriptors. Historically computer vision systems have employed two types of descriptors for extracting image representations. *Local* descriptors that aim to extract low-level features around points of interest in the images and *global* descriptors which encode a compressed representation of the entire image from the set of local features. These two descriptors are used by *handcrafted* methods which consists of a manually crafted vision pipeline from the pixels to a vector representation based on solid mathematical modeling. Recently, the field has moved towards *end-to-end learning* systems based on CNN where the goal is to learn at the same time low-level to high-level feature using a hierarchical structure.

Histograms, vocabularies and local descriptors — The notion of *histogram* is introduced by Swain et al. (1991) for image indexing. They propose to associate objects to a histogram of their colors and they develop a method called *histogram intersection* for indexing. The introduction of histogram allow to avoid explicit point correspondences and to model relations between different image points implicitly coded in the receptive field responses. The histogram approach is extended by going beyond the color and considering response of convolution filters such as Gabor filters and Gaussian derivatives applied to image patches (Schiele et al. 1996).

Following the same principle Leung et al. (2001) propose to consider histograms of local features called textons. The method consists of constructing a vocabulary

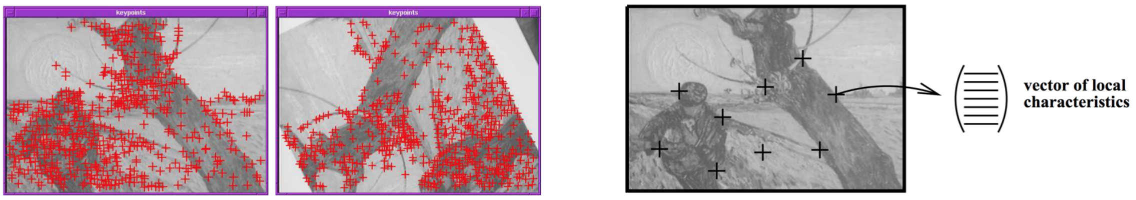


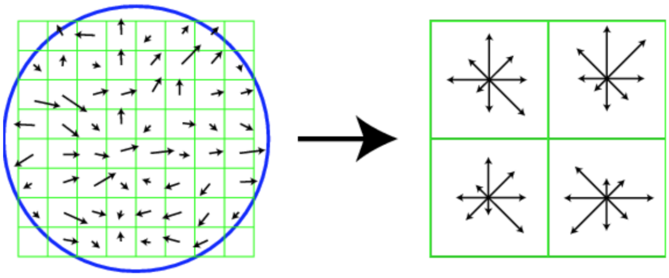
Figure 2.10 – **Local Greyvalue Invariants for Image Retrieval.** (Left) Red crosses represent the Interest Points detected on the same scene under rotation. – (Right) Vector representation of an image.
Figure reproduced from (Schmid et al. 1997).

of prototype tiny surface patches with associated local geometric and photometric properties. The idea of local descriptor is finally popularized (Schmid et al. 1997) by using the idea of representing images with local image descriptors based on image gradients (Koenderink et al. 1987). Schmid et al. (1997) applied this procedure to the task of image retrieval and image matching. A visualization is shown in Figure 2.10.

Scale Invariant Feature Transform — The success of local descriptors for object recognition and image matching largely inspired the work of David Lowe (Lowe 1999; Lowe 2004), leading to the highly influential Scale-Invariant Feature Transform (SIFT) descriptor. SIFT features are extracted using a multi-step procedure. First the algorithm detects point of interest also called keypoints (*Scale-space Extrema Detection*). This done by convolving the image with difference Gaussian filters and then taking the difference of successive Gaussian-blurred (also called "Difference of Gaussians" - DoG) images. Keypoints are taken from the extrema values from the difference of gaussians. Second the number of detected keypoint is reduced (*Keypoint Localization*) by discarding low-contrast keypoints and eliminating edge responses. Third each keypoint is assigned an orientation (*Orientation Assignment*) based on local image gradients directions for achieving invariance to rotation. Finally we assign a descriptor to each keypoint (*Keypoint Descriptor*). To do so we compute we create a set of orientation histograms of image gradients magnitude and orientation values in the 4×4 pixel neighborhoods region with 8 bins each around the keypoint. The magnitudes are weighted by a Gaussian function and the final keypoint detector is a vector of 128 elements ($4 \times 4 \times 8$).

A visual explanation of the final step of the SIFT algorithm is given in Figure 2.11 and an application of SIFT descriptor is given in Figure 2.12. An extension of SIFT called SURF (Bay et al. 2006) is developed with the objective to be a fast approximation of SIFT.

Bag-of-words approaches — The objective of Bag-of-Words (BoW) is to produce a vector representation given an input image. The BoW approach is inspired from



(a) image gradients

(b) keypoint descriptor

Figure 2.11 – **Explanation of the SIFT descriptor.** (Left) Image gradients. – (Right) Keypoint descriptor.
Figure reproduced and caption from (Lowe 2004).

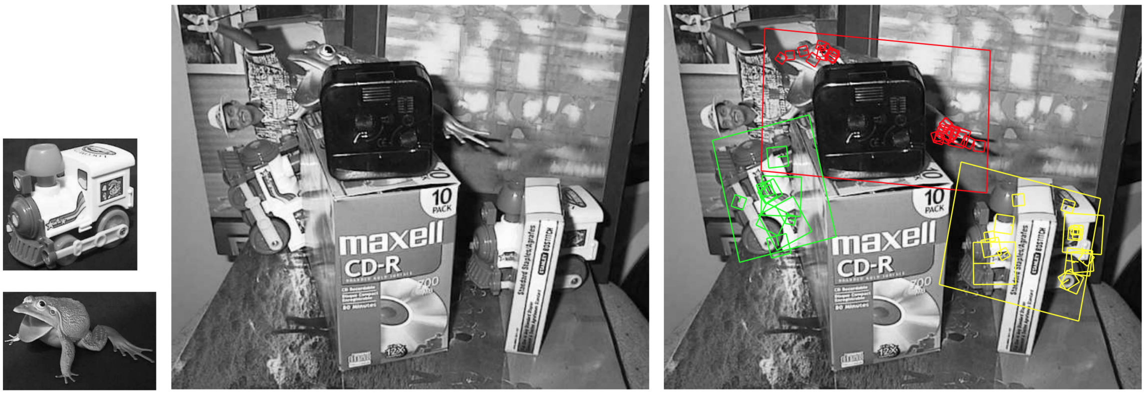


Figure 2.12 – **Application of SIFT.** (Left) The training images for two objects. – (Middle) These can be recognized in a cluttered image with extensive occlusion. – (Right) The results of recognition. A parallelogram is drawn around each recognized object, showing the boundaries of the original training image under the affine transformation solved for during recognition.
Figure reproduced and caption from (Lowe 2004).

text information retrieval (Salton et al. 1986) where the idea is to represent a document as a histogram of occurrence rates of words from a dictionary. While initially proposed for visual recognition by Ma et al. (1999), (Csurka et al. 2004) popularized this approach for the task of object recognition. The BoW approach for representing an image consists of a three-step procedure.

First, *local descriptors* are extracted from patches of the images at different scales. SIFT is the most used local descriptor and converts patches to 128 dimensional vectors as described previously. Hence an image is a collection of vectors of same dimension.

Second, the *codebook generation* (also called *coding*) step encodes the local descriptors as a function of the dictionary visual words, and outputs visual codes. One simple method is performing k-means clustering over all the vectors. Codewords are then defined as the centers of the learned clusters, and local descriptors are hard-assigned to clusters. However more sophisticated functions (Perronnin et al. 2010; Jegou et al. 2010) are proposed with the underlying goal to better represent the visual content. Perronnin et al. (2010) find that encoding first and second order statistics leads to improved performances, by decreasing the codebooke size. The proposed method called Fisher Vector (FV) copes with large sets of images, using Fisher Kernels in a Gaussian Mixture Model framework with computationally efficient linear classifiers. Jegou et al. (2010) introduces Vector of Locally Aggregated Descriptors (VLAD) following a simpler aggregation technique, which does not require storing second-order statistics. It can be considered as a more efficient variant of FV and was originally introduced for image retrieval. An extension of this approach is introduced by (Zhou et al. 2010) namely Super-Vector Coding (SVC).

Third, the *pooling* (also called *aggregation*) stage constructs a single vector representation from the set of local visual codes collected across the image. The standard approaches consist of average pooling (Sivic et al. 2003) or max pooling. However such pooling function looses spatial information about the disposition of the local visual codes. Lazebnik et al. (2006) develop Spatial Pyramid Matching (SPM) for taking into account the spatial information using the pyramid matching method introduced by Grauman et al. (2005). SPM consists in applying the histogram computations to a pyramid of image regions such as shown in Figure 2.13.

Finally the classifier can be learned using any ML classifier methods. The most popular in CV are k-NN (T.Cover et al. 1967), decision trees (Breiman et al. 1984) and Support Vector Machine (SVM) (Boser et al. 1992).

The introduction of large-scale datasets pushes the limits of the visual recognition system. The Pascal VOC challenge (Everingham et al. 2010) is the first competition followed by the introduction of Imagenet in 2009 (Russakovsky et al. 2015), which corresponds of more than 1.3 million annotated images spread among 1k classes.

End-to-end learning with Convolutional Neural Networks — The big conceptual shift from hand-crafted features to end-to-end learned features appears in 2012, when AlexNet (Krizhevsky et al. 2012) won the Imagenet object recognition challenge. Instead of proposing a multi-stage procedure, in the pipeline of traditional hand-crafted methods, Krizhevsky et al. (2012) propose to use an end-to-end differentiable neural network composed of convolution, pooling and fully-connected layers for mapping the input image to the predicted vector of probability. In some sense the different stages present in handcrafted methods still exist but they are all connected and learned during the training procedure. The only hand-crafted stage is to correctly design the neural network. It consists

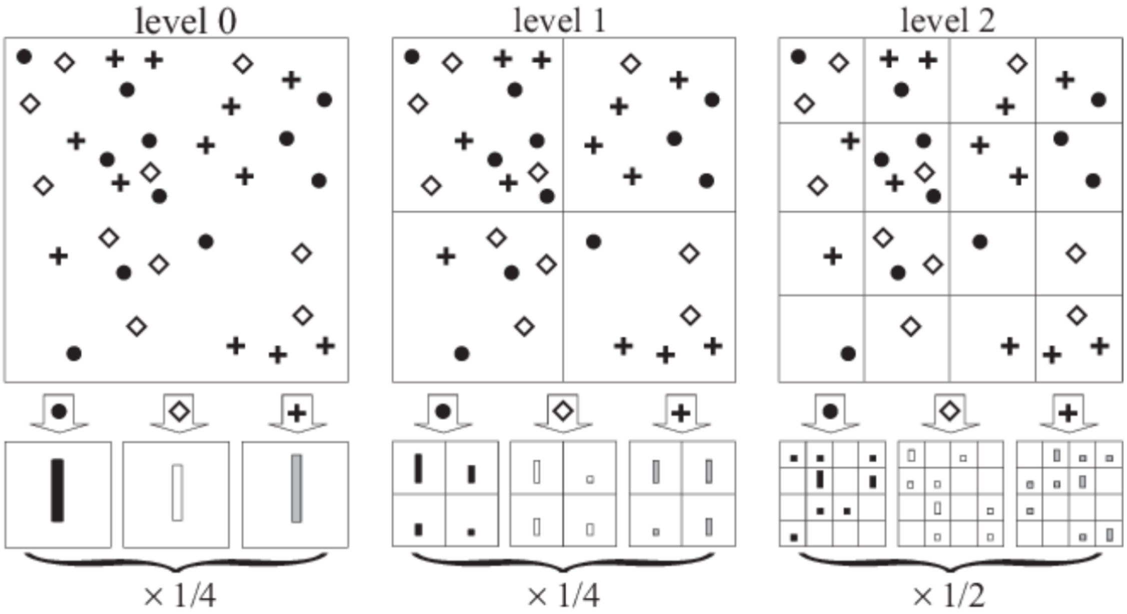


Figure 2.13 – **Spatial Pyramid Matching.** Toy example explaining the SPM strategy. Point of interests are extracted using different image regions and they are then combined.
Figure reproduced from (Lazebnik et al. 2006).

of choosing the neural networks architectures and the hyperparameters of each layers such as presented in Section 2.1.1.

This breakthrough allows the application oto many different tasks after realizing that the features learned on Imagenet were transferable to other tasks (Donahue et al. 2014; Zeiler et al. 2014).

Modern Convolutional Neural Networks — The performance on the Imagenet challenge keeps improving year after year with the release of more powerful CNN models (Simonyan et al. 2015; Szegedy et al. 2015; He et al. 2016). The increase in performance is highly correlated with the network depths such as shown in Figure 2.15.

Simonyan et al. (2015) propose the VGG architecture composed of small convolution filters. Szegedy et al. (2015) develop GoogleNet (also called InceptionV1) composed Inception module that reduce the number of trainable weights. They also employ Global Average Pooling (GAP) from the last feature map for extracting a final fixed size vector representation of the input image which drastically reduce the number of paramaters. He et al. (2016) introduce Resnet which is very deep neural network composed of residual module and batch normalization (Ioffe et al. 2015) for facilitating backpropagation and hence the training of the parameters.

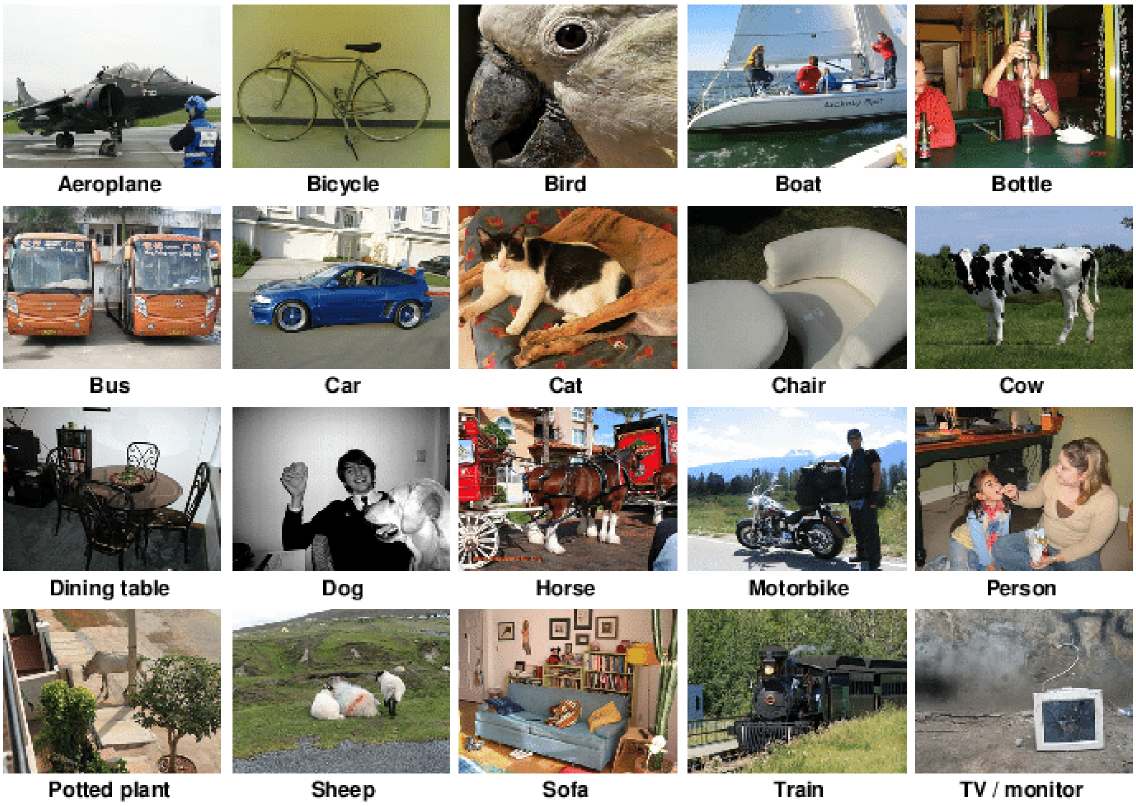


Figure 2.14 – **Samples from the Pascal VOC dataset.**
Figure reproduced from (Everingham et al. 2010).

2.2.3 Object detection

Images may contain multiple objects and this is often the case in real-word scenarios. Hence for having a more fine-grained understanding of complex images, CV systems need to go one step further object recognition by detecting instances of semantic objects. This is the purpose of object detection systems that aims at finding potential locations in the input image that contain objects and classify them. Hence solving an object detection problem consists of a supervised learning task where the system should output localisations of objects present in the images (a bounding box) as well as their categories. The varying number of objects present in the images make this problem very challenging.

Historically, this task has been solved by methods (Dalal et al. 2005; Felzenszwalb et al. 2008) operating with *sliding windows* at different scales on the images using. Then this problem has been tackled using a *two-stage* procedure (Ren et al. 2015). First, detecting potential locations that may contains an object and then, recognizing the category of the object of each of these locations. Finally a third category of methods (He et al. 2017) proposes to detect objects using a *one-stage*

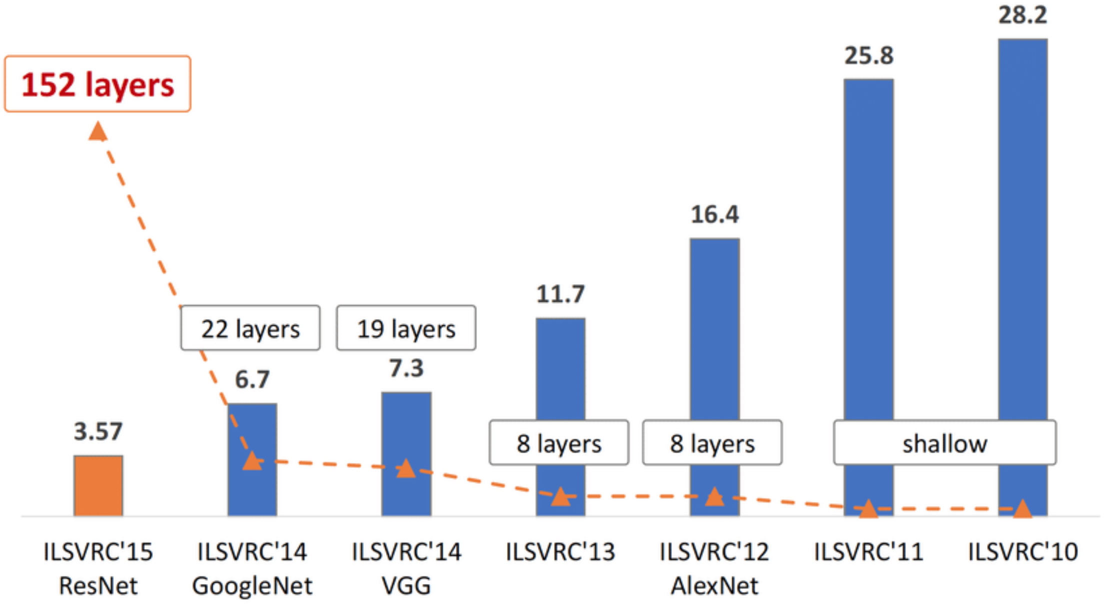


Figure 2.15 – **Evolution of the Imagenet results through time.** Figure reproduced from (Bottou et al. 2016).

procedure by predicting the locations and the category of an object at the time. We refer the reader to Liu et al. (1809) for a survey on this topic.

Handcrafted methods — A new feature descriptor called Histogram of Oriented Gradients (HOG) is proposed by Dalal et al. (2005) for the task of person detection. This method consists of counting occurrences of gradient orientation in localized portions of an image. While HOG share some similarities with SIFT, the main difference is that the features are computed on a dense grid. They employ a sliding window detection and employ non-maxima suppression during inference time. For the task of object detection, Felzenszwalb et al. (2008) introduce Deformable Part-based Model (DPM) by combining HOG and SPM, and finally using a Latent Support Vector Machine (LSVM).

Object proposals — The localization of objects by operating on sliding window technique at different scales is unefficient and very slow. To encounter this issue, some works (Alexe et al. 2010; Sande et al. 2011) propose to define a first step that onsists of generating parts of the images that may contained objects. This procedure is called *object proposal* and is used as a first stage of the object detection pipeline. Alexe et al. (2010) propose a generic objectness measure describing how likely it is for a window to contain an object rather than background or a small parts of objects. They employ a Bayesian framework where they measure characteristics of objects, such as appearing different from their surroundings and having a closed boundary. The computations of the objectness takes around 4 seconds per image. Finally, a more robust method called Selective Search window proposals (Sande et al. 2011) has becomed the key component for building state-of-the-art object detector. It consists of using a bottom-up grouping of image regions for generating a hierarchy of regions (from small to large) by using the image structure to guide the sampling process.

Two-stages CNN methods — The field of object detection gains in popularity with the introduction of CNN approaches as well as new large-scale datasets such as MSCOCO (Lin et al. 2014). RCNN (Girshick et al. 2014) is the first method employing CNN features. They use the Selective Search algorithm for localizing potential object locations, then wrap the Region of Interest (RoI), compute CNN features using a pre-trained Alexnet and finally classify the region. The extension called Fast-RCNN (Girshick 2015) simplifies the process by feeding the entire image to a CNN and then extracting features using a RoI pooling layer. They also propose to jointly learn the object proposals. Faster-RCNN (Ren et al. 2015) speeds up the object proposal stage by introducing the Region Proposal Network sharing full-image convolutional features with the detection network enabling cost-free region proposals.

Representing an object shape by its bounding box localisation is sometimes not enough when objects have very specific shape. This can be solved by solving the task of instance segmentation goes one step being object detection and which consists of predicting the pixel masks of an object. He et al. (2017) extend Faster-RCNN by adding a mask prediction head on top of the framework. They predict a fixed size mask that can be wrap in the original image using the coordinates of the bounding boxes. They also propose to extract better features from the regions of interests by replacing the RoI Pool by a RoI Align operation. An overview of the Mask-RCNN model is given in Figure 2.16. Recent extensions are proposed for a better fine-grained description of the visual content of the scene with TensorMask (Chen et al. 2019) and the Panapoptic task (Kirillov et al. 2018).

Single-stage object detection — Two-stage object detection methods provide high quality predictions, however they cannot run on real time on a Graphics Processing Unit (GPU). This is mainly due to the two-stages strategy of first predicting the proposals and then recognizing the object categories. To encounter this issue, recent works (Redmon et al. 2016; Liu et al. 2016b; Lin et al. 2017) propose to tackle the problem of object detection using a single stage procedure. Single Shot Detector (SSD) (Liu et al. 2016b) propose to discretize the output space of bounding boxes into a set of default boxes over different aspect ratios and scales per feature map location. They also combine prediction from multiple feature maps with different resolutions. At the same time, You Only Look Once (YOLO) (Redmon et al. 2016) apply a single neural network to the full image which divides the image into regions. And for each region they predict bounding boxes and

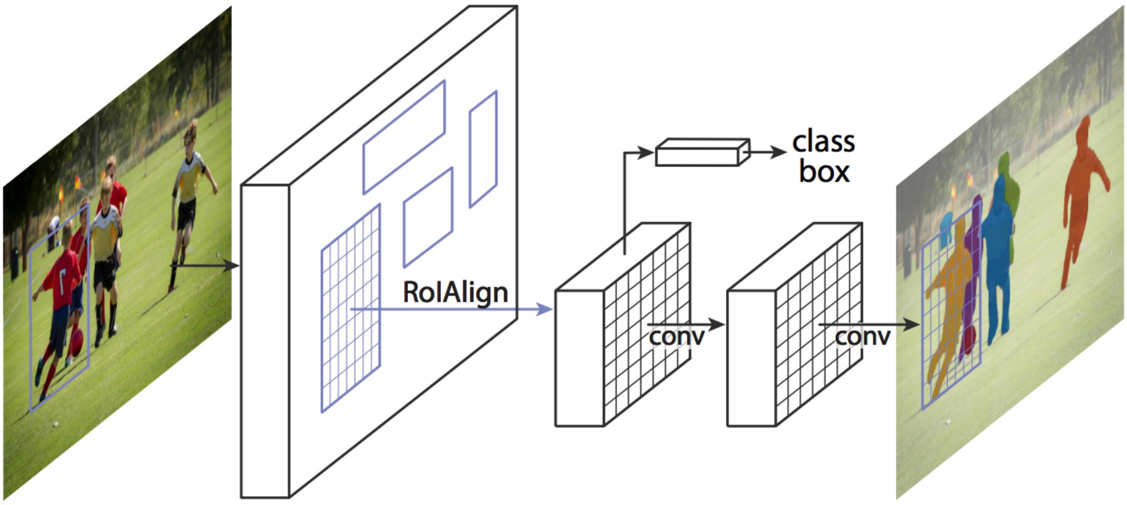


Figure 2.16 – **Overview of Mask-RCNN.** First, a CNN backbone extracts features from the input image. Second, fixed-size feature maps are extracted using the RoI Align operation given the object proposals. Finally, the object class, box offset and object masked are predict from the object features.
Figure reproduced from (He et al. 2017).

probabilities. Since the predictions are local the network needs to take attention at the global context of the image. They use non-maximal suppression at test time for getting ride of redundant object predictions. In follow-up papers (Redmon et al. 2017; Redmon et al. 2018), the authors boost the performance of the method by training at the same time the classification task as well as the object detection task. Since the object classes from object detection and recognition datasets (COCO and Imagenet) does not overlap they use hierarchical three structure based on WordTree.

2.2.4 Human pose estimation

The task of human pose estimation consists of identifying persons in an image as well as the location of their joints (arms, head, hands, . . .). Approaches can be categorized into two categories: top-down or bottom-up approaches. Top-down approaches first localize humans present in the scene and then estimate the pose of each person, while bottom-up approaches start by identifying human joints and then group them into person instances. In this section we review these two kinds of approaches using handcrafted and CNN features.

Human body definition and geometry — Pose estimation is a difficult and active research topic in CV which is mainly due to the fact that the human body is a deformable object compared to solid common objects encountered in object detection, such as a chair. The human body is composed of 230 joints and has 244 degrees of freedom. However, most of the works are simplying definition by considering that it is composed of 10 parts and 20 degrees of freedom which already makes the problem challenging. A first definition of the human body in term of cylinders in given by O'Rourke et al. (1979) and the first works (Hogg 1983; Lee et al. 1985) in this topic focus on fitting a 3D human body model given a single image. However, recovering 3D model is very challenging and may even be a ill-posed problem due for example to occlusion.

Handcrafted methods — Build on the seminal work on pictorial structures from Fischler et al. (1973), Forsyth et al. (1997) define body plans. This method corresponds to a sequential grouping of parts for finding people present in an image. Following a similar trend, Mori et al. (2004) develop a method for detecting human body parts one by one and assemble them by enforcing global constraints. Ramanan (2006) propose an iterative parsing process for learning better and better features without relying on face or skin detection. Agarwal et al. (2005) propose to first extract a shape descriptor vector extracted from the image silhouettes and then to regress each joint of the predifined 3D human pose. Following the success of DPM (Felzenszwalb et al. 2008) in object detection, Yang et al. (2011) capture the orientation with a mixture of templates for each body part which is useful for capturing the notion of local rigidity and hence dealing with human body constraints. This method is then extended by Gkioxari et al. (2014) which define *k*-poselets where a each poselet corresponds to a body part.

End-to-end 2D Human Pose — The first work to proposed DL architecture for human pose estimation is DeepPose (Toshev et al. 2014) which is casting the task of human pose estimation as a regression problem allowing end-to-end learning. The proposed model is much simpler than previous approaches using graphical models. They improve the predictions done by the model by employing a cascade of pose regressors at different resolutions. However they need to use a person detector for the first stage.

On the same spirit Chen et al. (2014) also use the fact that the local appearance of a joint can help in predicting the appearance of neighboring joints by employing a graphical model. Carreira et al. (2016) expend the expressive power of hierarchical feature extractors such a CNN to encompass both input and output space by using a top-down feedback. They propose to self-correct the predicted human joints in a iterative manner with a mechanism called Iterative Error Feedback.

Following this self-correction strategy, human pose estimation methods start employing more context features (Wei et al. 2016; Newell et al. 2016; Luvizon et al. 2019). Newell et al. (2016) propose Stacked Hourglass Networks which the main motivation being that repeating bottom-up and top-down processing with

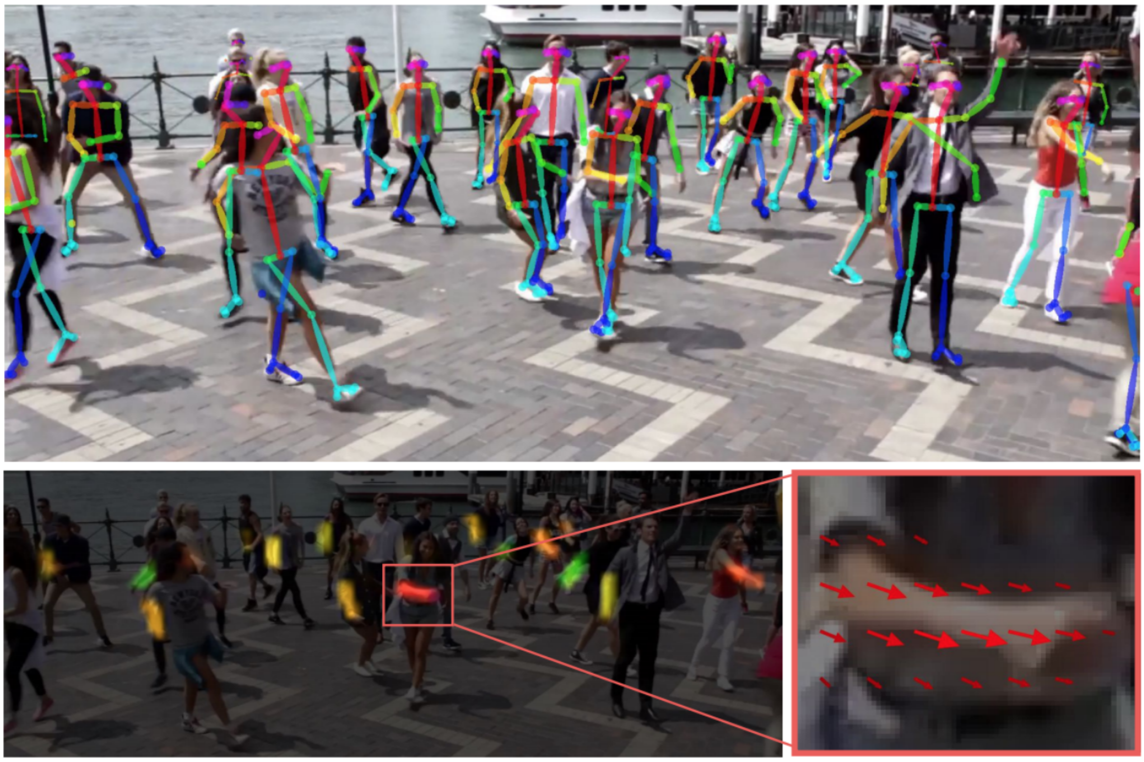


Figure 2.17 – **Visualization of OpenPose results.**
Figure reproduced from (Cao et al. 2017).

intermediate supervision leads to better performance. They employ successively pooling and upsampling operations before outputing the final predictions. Wei et al. (2016) introduce Convolution Pose Machines which is a sequence of convolution networks producing a 2D belief map for each human body part. This strategy enforces the network to implicitly learn spatial arrangements between human body parts.

The next generations of human pose estimation models (Pishchulin et al. 2016; Cao et al. 2017) tackle more real world images with occlusions and high number of people in the scene. Pishchulin et al. (2016) employ a method that jointly detect the number of people in the image and predict the human joint locations. The method is able to identify occluded body parts and disambiguates body parts between people close to each other. A major advance in mult-person 2D pose estimation is done by the introduction of OpenPose (Cao et al. 2017). The method proposed by the authors consists of predicting confidence maps for detecting body parts and part affinity fields for body parts association which has the advantage of being a nonparametric representations. An overview of the OpenPose method is shown in Figure 2.17. OpenPose is a open-source and real-time system on CPU for multi-person 2D pose detection. The method has also been extended to hand and facial keypoints. Neverova et al. (2017) propose to add topological constraints to hand pose estimation, formulated as a weakly and semi-supervised problem.

Beyond 2D human pose — 2D human pose estimation methods show excellent performance on current benchmarks however occlusion still remain a big challenge. One way to solve this issue is to recover the full body part locations by reasoning about the full human body. Recent works propose to predict the the full-body 3D human pose (Rogez et al. 2019; Mehta et al. 2017). Rogez et al. (2019) introduce a method that generates and scores a number of pose proposals per image. It does not require an approximate localization of the humans for initialization but the pose is refined in a second stage. Mehta et al. (2017) propose a similar three-stages procedure. First they extract the actor bounding box from 2D detections, second they regress 3D pose and third they compute the global root position by aliging 3D to 2D human pose.

Finally, recent works (Rza Alp Guler 2018; Kanazawa et al. 2018) go one step further by predicting a human body mesh which allows to model at the same time the human pose and shape. DensePose (Rza Alp Guler 2018) is a variant of Mask-RCNN which instead of predicting a pixel mask, they densely regress part-specific UV coordinates. Kanazawa et al. (2018) introduce Human Mesh Recovery which aims at minimizing the reprojection loss of keypoints. This allows to train the model on ground truth 2D annotations only.

2.2.5 Video representations

Moving from image to video understanding requires modeling the temporal information. This dimension cannot be treated as the spatial ones and plays a crucial role for extracting information such as motion that cannot be estimated from static image only. A common task of interest when dealing with video content is to predict the action happening in the video. Since actions are (most of the time) performed by humans it is of high interest to model human behavior as first step before extracting semantic content. Initial works on action recognition (Ramanan et al. 2007; Efros et al. 2003) relies on body parts or person detectors for predicting the action that happens in the video. However such detectors cannot always be deployed in practice and/or can be noisy. Unconstraint methods have been proposed to encounter this issue (Laptev et al. 2003; Wang et al. 2013a). Similar to image understanding frameworks, the introduction of *end-to-end* methods (Carreira et al. 2017) with deep learning methods has changed the field of action recognition both for unconstrained and human related methods.

Articulated human model based methods — An important question to answer when modeling actions in videos from a human body perspective is to know what is the minimal information for recognizing motion. Johansson (1973) demonstrates that the visual interpretation of few moving light displays attached to the human body can be enough for categorizing the action performed by a person. This seminal work motivated approaches (Ali et al. 2007; Parameswaran et al. 2006;

Yilmaz et al. 2005) using trajectories of joint positions, landmark points or body parts based on 3D or 2D human models for recognizing the human action. The localization of body parts (Ramanan et al. 2007; Ferrari et al. 2008) shows excellent results: however, it is still a difficult problem especially for unconstrained videos which limits its applicability.

Human global dynamic methods — For solving this issue of noisy baody parts localisation researchers developed a less constrained approach which consists of modeling the global human dynamic given a region of interest centered on the human body (Polana et al. 1994). The modeling of the human dynamic can be split into two categories.

The first one makes use of the *shape masks* and *silhouette information*. Yamato et al. (1992) propose the first approach based on silhouette images using the ratio of foreground-background among a grid over the silhouette. Bobick et al. (2001) develop a method based on shape masks. They propose to recognize human actions using motion energy images and motion history images, hence they are the first to propose temporal template for action recognition. Blank et al. (2005) propose to use space-time shapes from silhouette information computed using background substraction. They extract features such as local saliency and shape structure.

The other one uses *optical flow* (Gibson 1950) and *shape information*. The first work by Polana et al. (1994) propose to use spatio-temporal grids of optical flow magnitudes. Efros et al. (2003) propose a two stage strategy by first tracking soccer players in videos and then computing a descriptor on the stabilized tracks using blurred optical flow. Fathi et al. (2008) employ a similar technic by building mid-level features from low-level optical flow information. Laptev et al. (2007) propose to detect drinking actions in movie using HOG, motion features and a pre-filtering operation using a human detector.

Handcrafted unconstrained local descriptors — Relying on a person detector or a tracking system propagates the error made on the first stage of the system. For solving this issue, several works propose handcrafted unconstrained methods similar to the method employed in object recognition. However since the input signal is spatio-temporal they adapt the local descriptors to this data type.

A first category of work focuses on modeling low-level representations. Laptev introduce Space-Time Interest Points (STIP) (Laptev et al. 2003; Laptev 2005) by extending the notion of spatial interest point to the spatio-temporal domain. They use the Harris and Förstner interest point operator (Harris et al. 1988) for detecting local spatio-temporal structures that have significant structure. A visualization of STIP is shown in Figure 2.18. Kläser et al. (2008) build on HOG from Dalal et al. (2005) by proposing HOG3D which consists of computing gradients in space and time. Scovanner et al. (2007) extend SIFT to the video domain. SPM

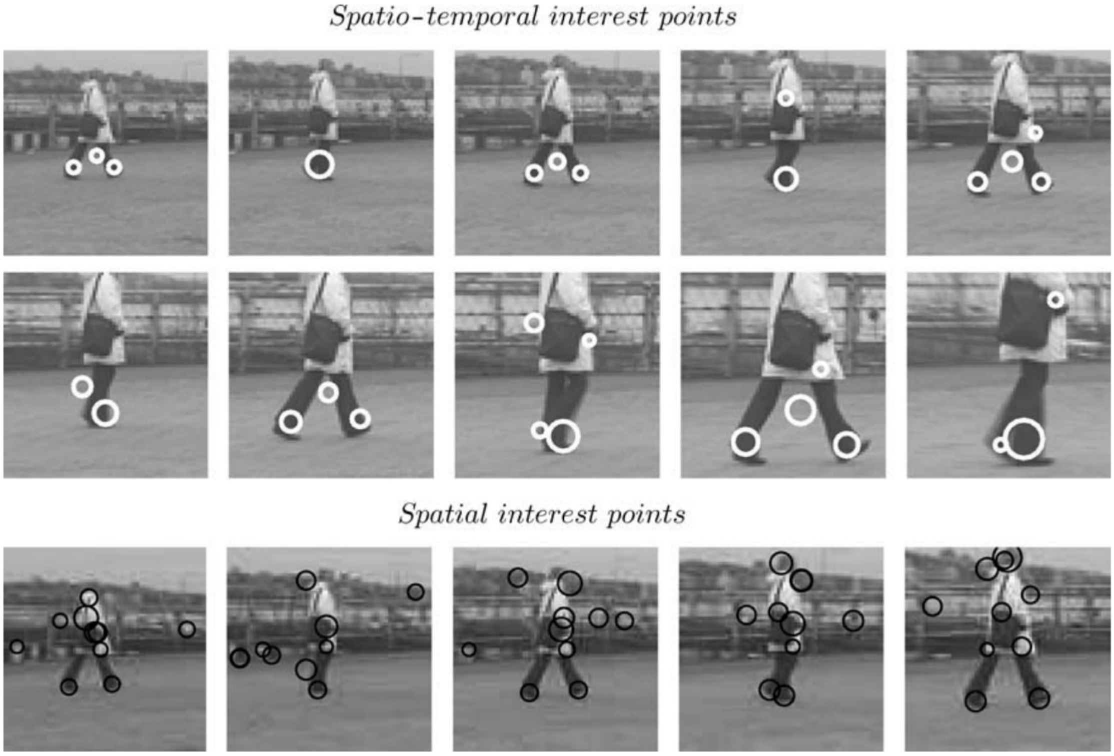


Figure 2.18 – **Space-Time Interest Points.** Difference between Space-Time Interest Points (Laptev 2005) and pure spatial point of interests (Mikolajczyk et al. 2002).
Figure reproduced from (Laptev 2005).

(Lazebnik et al. 2006) is also extended to video data by Laptev et al. (2005) with Histograms of Optical Flow (HOF). Local features are extracted from several video volumes and then aggregated by concatenation. Dalal et al. (2006) introduce Motion Boundary Histograms (MBH) by matching HOG local features extracted from two images. In a similar spirit, the Hessian detector proposed by Willems et al. (2008) is an extension of the blob detector (Beaudet 1978) in images. Dollar et al. (2005) develop spatio-temporal interest point detectors to find local regions of interest in space and time. It allows to extract cuboids that are finally used for behavior recognition.

Trajectory features — After adapting low-level descriptors from space to space-time signal, works focus on developing mid-level representations (Wang et al. 2013b; Wang et al. 2013a; Gaidon et al. 2013). Matikainen et al. (2009) propose to recognize human actions by analyzing the motion based on quantized trajectory snippets of tracked features. They show that feature trajectories are efficient for representing videos. Following this strategy, Wang et al. (2013b) propose the Dense Trajectories (DT) descriptor by showing that descriptor sampling along

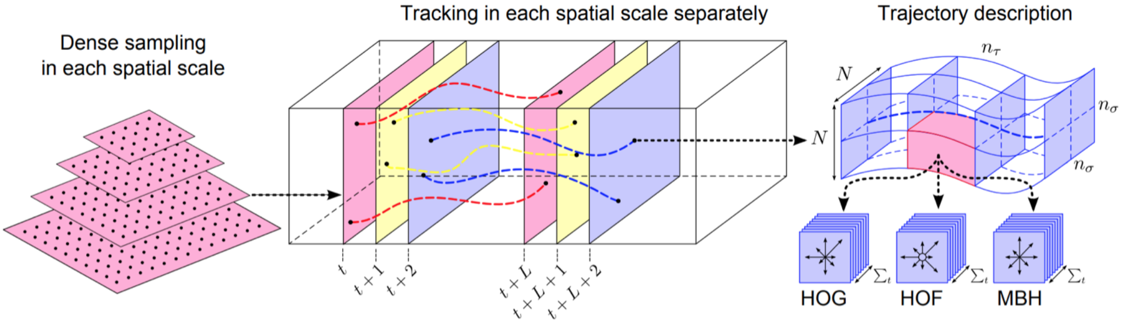


Figure 2.19 – **Improved Dense Trajectories.** Overview of the method for extracting and aggregating dense trajectories.
Figure reproduced from (Wang et al. 2013a).

dense trajectories outperforms sparse sampling. They introduce a novel descriptor based on MBH robust to camera motion. The extension, namely Improved Dense Trajectories (IDT) (Wang et al. 2013a), improves the representation by using camera motion calibration. This can be done through the estimation of a homography with RANSAC after macthing the feature points between frames using SURF descriptors and optical flow. An overview of the proposed approach is shown in Figure 2.19. IDT is still a very competitive baseline compared to CNN approaches. In a final extension, Wang et al. (2015) find a better way to aggregate descriptors using FV.

Handcrafted temporal modeling — Some works (Fernando et al. 2015; Sand et al. 2008) focus on the modeling of the temporal dynamic with a focus to long videos with the oal to build high-level representaions. Fernando et al. (2015) explicitly represent the temporal representation by learning a ranking function per video via a ranking machine. Another approach is to model the temporal evolution of some point coordinates based on long-term trajectories (Sand et al. 2008; Brox et al. 2010; Lezama et al. 2011). Lezama et al. (2011) aim to capture long-range temporal interactions among objects.

Early days of CNN on space-time data — The first method that introduces neural networks in the context of action recognition is proposed by Baccouche et al. (2011). The authors employ 3D convolutions for extracting spatio-temporal features in a CNN architecture as well as LSTM for modeling long-range temporal dependencies in the sequence. It is interesting to notice that same authors extend this architecture with an unsupervised training (Baccouche et al. 2012). This seminal work motivated most of the DL approaches for video analysis that employ similar architectures. We split them in three distinct categories:

- First, some works extend the spatial convolution kernels to spatio-temporal one by incorporating 3D convolutions instead fo 2D convolutions (Taylor et al. 2010; Ji et al. 2013; Karpathy et al. 2014; Tran et al. 2015). It is the simpliest extension which consists of learning space-time kernel filters instead of space kernel filters only. However moving from 2D to 3D convolutions greatly increase the number of learnable parameters which requires more computational power and bigger dataset Karpathy et al. 2014.

- Second, researchers combine 2D CNN with RNN (Baccouche et al. 2011; Donahue et al. 2015; Ng et al. 2015). For each image of the video sequence a feature vector is extracted using a 2D CNN shared over all frames. And then a RNN is employed for modeling the temporal representation of the sequence of spatial features. This approach shows some difficulties to model fine-grained action since local motion cannot be modeled.

- Finally, the third categories is composed of two-stream approaches (Simonyan et al. 2014; Feichtenhofer et al. 2016) which combine appearance and motion features. Appearance features are extracted from the RGB stream using a 2D CNN while the motion features are computed using a 2D CNN from the optical flow. This needs to pre-extract the optical flow since it is computationally extensive to compute.

However handcrafted video representations such as IDT (Wang et al. 2013a) are still competitive approaches on standard bechnmarks such as UCF-101 (Soomro et al. 2012) or HMDB-51 (Kuehne et al. 2011) compared to the works based on CNN models explained above. They obtain similar performances on standard action recognition benchmarks UCF-101 (Kuehne et al. 2011) and HMDB-51 (Soomro et al. 2012). Some work proposed to boost the performance of DL technics by modeling of the temporal aggregation (Wang et al. 2016a), employing long-term convolution (Varol et al. 2018) or using spatio-temporal multiplier (C. Feichtenhofer 2017).

Inflated 3D ConvNet — Finally a breakthrough in action recognition and more generally in video analysis is due to the introduction of a large-scale annotated dataset called Kinetics (Kay et al. 2017) composed of more than four hundreds thousands of clips. Carreira et al. (2017) propose to train a CNN composed of 3D convolutions (called I3D) by inflating 2D Imagenet pre-trained weights into 3D at the start of the training procedure. The I3D network (Carreira et al. 2017) shows great performance on smaller datasets by transfer learning and even state-of-the-art performances on tasks such as action localization and action detection.

Since 3D convolutions are taking a long time to train, some works (Xie et al. 2017; Tran et al. 2018) propose to decompose 3D kernels into spatial 2D kernels followed by a 1D temporal kernels (called (2+1)D kernel) which allow to reduce the number of trainable parameters while still being able to model space-time

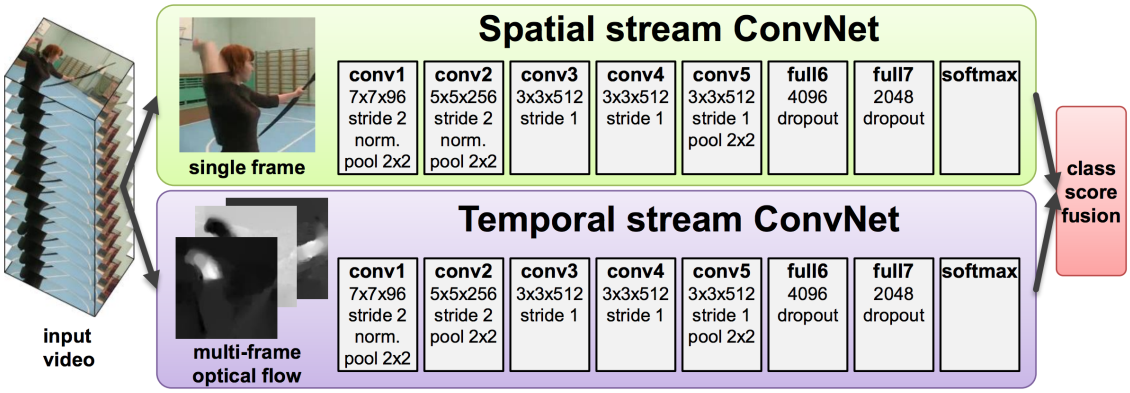


Figure 2.20 – **Overview of the two-stream approach for action recognition.** The appearance features are extracted from the RGB stream using a 2D CNN. The motion features are extracted by first computing the optical flow and then feeding it to a 2D CNN. The aggregation is done at the logits level.
Figure reproduced from (Simonyan et al. 2014).

feature. Xie et al. (2017) demonstrate that good performance can be reach by introduce (2+1)D convolutions only on the last layers while keeping 2D convolutions at the bottom of the network which greatly reduce the training and inference time. Tran et al. (2018) show that increasing the number of temporal channels can lead to better performance on action recognition tasks.

In a different topic, Wang et al. (2018a) show that the local space-time features can be improved by taking into account context information. They propose to use non-local neural network after each convolutional block for updating local feature map values based on context information.

While previous presented approaches handle short clips of a few seconds only, Wu et al. (2019a) introduce a Long-Term-Feature-Bank method for representing long videos.

They propose to use a feature bank for storing important information while iterating over the video. Feichtenhofer et al. (2019) extend the idea of two-stream approaches but without the need of optical flow as input by introducing SlowFast network. It consists of having two networks with skip connections taking as inputs the same video but with different frame rates. An overview of SlowFast network is shown in Figure 2.21.

Current action recognition methods are not efficient at training and inference time. They require a massive amount of GPUs for training which is not available for most academic research laboratories. To encounter this issue, Lin et al. (2019) propose a CNN based on 2D convolutions only using a temporal shifting strategy while maintaining good performance on popular benchmarks. The proposed

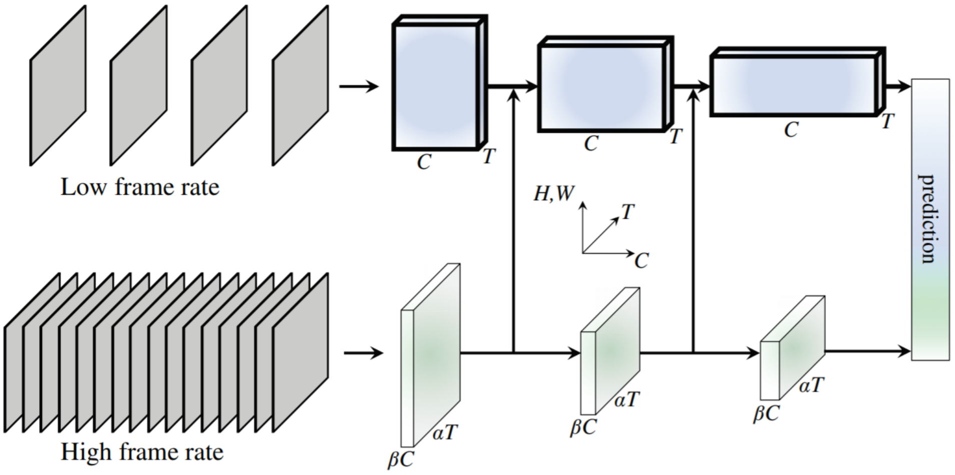


Figure 2.21 – **SlowFast network.** The *Slow* network (in blue) takes as input low temporal resolution while the *Fast* network (in green) takes fast temporal resolution. There are lateral connections from the fast pathway to the slow pathway.
Figure reproduced from (Feichtenhofer et al. 2019).

strategy consists of taking into account some spatial features from neighbouring frames.

Skeleton-based human action recognition — While offering excellent performances on standard action recognition benchmarks unconstrained video representations suffer of learning dataset bias by cheating on the context and background information. To encounter this issue researchers propose large-scale fine-grained human activity datasets such as NTU RGB+D (Shahroudy et al. 2016a) or UESTC (Ji et al. 2018). The task is to predict the action from the trajectories of skeleton joints (Du et al. 2015a; Yan et al. 2018; Zhu et al. 2016; Du et al. 2015b; Liu et al. 2017b; Ke et al. 2017). Du et al. (2015a) use a hierachical RNN for modeling long-term contexual information by dividing the human skeleton into subparts. Following a similar structured strategy, Yan et al. (2018) propose to see the skeleton joints as a space-time graph and train a Spatio-Temporal Graph Convolutional Networks (described on the next section). It leads to strong generalization capability. From a different perspective, Ke et al. (2017) propose to rearrange the sequence of skeleton joints positions as an image and to train a CNN. However, the main inconvenient of these approaches is that some human actions cannot be estimated from the skeleton trajectories only. It is sometimes important to look at the appearance for distinguishing two human actions sharing the same skeleton trajectory pattern. To encounter this issue, Luvizon et al. (2020) introduce a multi-task framework which jointly estimates human poses for each image and classifies the whole video sequence into human actions. The proposed method outperforms skeleton-based methods while working in real-time from RGB data only (Luvizon et al. 2018).

Multi-modal approaches — Some works (Neverova et al. 2016; Wang et al. 2018b; Sun et al. 2018) combine unconstrained approaches with human based methods with the leading gaol to build a more robust methods. A pose-based CNN features is proposed by (Chéron et al. 2015). The main motivation is that the representation for action recognition should derived from human pose. Hence they use a human pose detector for identifying joints locations and extract features using pre-trained CNN. Girdhar et al. (2017) propose to use the human pose as complementary information for learning an attentional pooling over the last feature map. This method allows to guide the extraction of features around the person detected in the video. Neverova et al. (2016) propose a multi-modal training strategy for fusing information from human pose, audio and appearance streams. Recently, researchers model the person-object interactions (Wang et al. 2018b; Sun et al. 2018) for improving the final video representation. Wang et al. (2018b) propose to handle a video as a space-time graph where the nodes are represented by the object detected using a pre-trained object detectors. Following a similar strategy, Sun et al. (2018) introduce an actor-centric method which is specifically modeling the interactions between the persons detected in the scene and their surrounding objects.

2.2.6 Attention mechanisms

Human perception focuses selectively on parts of the scene to acquire information at specific places and times as shown in Figure 2.22. In ML, this kind of processes is referred to as attention mechanism (Itti et al. 1998), and has drawn increasing interest when dealing with language (Bahdanau et al. 2014; Kim et al. 2017b), images (Larochelle et al. 2010) and other data. Integrating attention can potentially lead to improved overall performance, as the system can focus on parts of the data, which are most relevant to the task. Attention mechanisms are at the head of ou contributions, we therefore present them in more details in this chapter.

Attention mechanisms for Machine Translation — The first attention mechanism in DL was proposed in the field of NLP and more specially for the task of Machine Translation (Bahdanau et al. 2014). This task can be cast as a sequence-to-sequence problem where the input is a variable length sequence from one language and the output is the translated sentence in another language. The architectures devoted to this task are based on an encoder-decoder pipeline where the encoder and decoder are both RNN. This is mainly due to the fact that the final hidden representation output by the encoder should encode the information

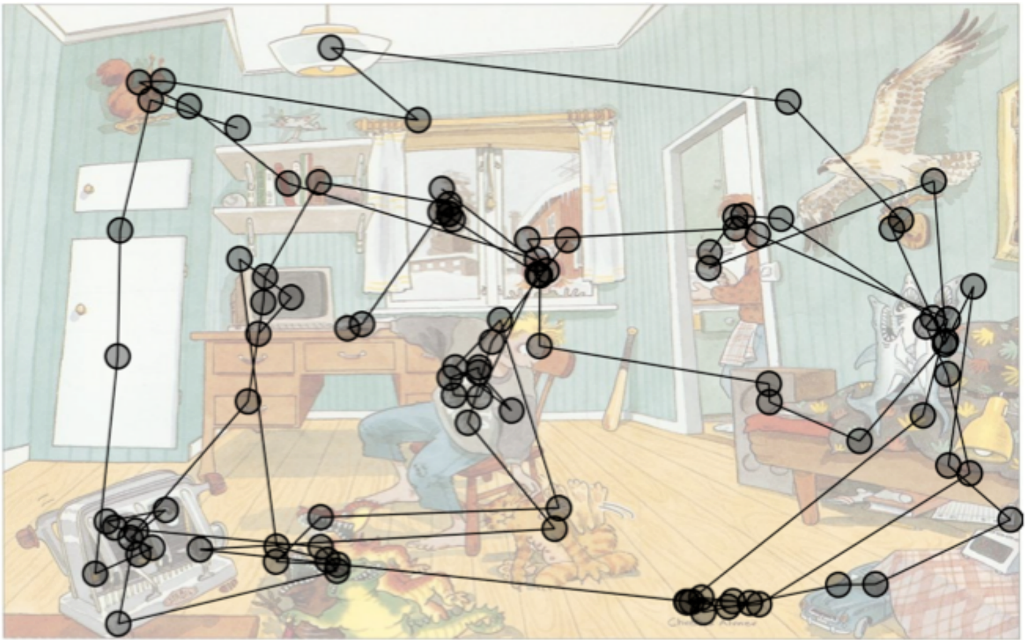


Figure 2.22 – **Human attention: gaze patterns.** Humans use spatial attention to prioritize the processing of visual information, by selecting parts of the visual field. This mechanism is an iterative process for gathering information about a visual content.
Figure reproduced from (Roger et al. 2012).

about the full sentence and this is difficult for generating a good output for the decoder.

For solving this issue, Bahdanau et al. (2014) propose a global attention-based mechanism where the main idea is to automatically attend to the most important parts of the sequence of hidden representations output by the encoder while generating the new sequence with the decoder. We give an overview of the proposed mechanism.

The Machine Translation problem can be seen as a sequence-to-sequence task where we have a sequence $x = (x_1, \dots, x_s, \dots, x_n)$ as input and we wish to predict another sequence denoted $y = (y_1, \dots, y_t, \dots, y_m)$ as output. Both sequences may differ in term of length. The input sequence x is encoded sequentially using a (bidirectional) LSTM denoted g_E such that $\bar{h}_s = g_E(x_s, \bar{h}_{s-1})$ resulting in a sequence of hidden states denoted $\bar{h} = (\bar{h}_1, \dots, \bar{h}_s, \dots, \bar{h}_n)$. The prediction of the output sequence is also done in a recurrent manner by incorporating the mechanism system as a way to pay attention to the most discriminative parts of the encoded input sequence. To do so we employ a *context vector* c_t which is automatically encoding contextual information within the whole sequence about the most related information regarding to the current timestep.

$$h_t = g_D(y_{t-1}, h_{t-1}) \tag{2.32}$$

$$\tilde{h}_t = f(h_t, c_t) \tag{2.33}$$

$$y_t = \text{softmax}(W_y \tilde{h}_t) \tag{2.34}$$

where h is the hidden state of the output sequence, $\tilde{h}$ is the hidden state of the output sequence updated by the *context vector* c, g_D is the recurrent decoder, f is a Multi-Layer Perceptron (MLP) and W_y is a learnable matrix (including bias). To do so we compute *attention weights* (can also be described as a *alignment score*) such that:

$$a_{ts} = \text{align}(h_t, \bar{h}_s) = \frac{\exp\left(\text{score}(h_t, \bar{h}_s)\right)}{\sum_{s'} \exp\left(\text{score}(h_t, \bar{h}_{s'})\right)}. \tag{2.35}$$

The score function may have different forms such that:

$$\text{score}(h_t, \bar{h}_s) = \begin{cases} h_t^\top \bar{h}_s & \text{dot product} \\ h_t^\top W_a \bar{h}_s & \text{bilinear operator} \\ f_a(h_t, \bar{h}_s) & \text{concatenation} \end{cases} \tag{2.36}$$

And finally we compute the *context vector* using a weighted sum over hidden states of the input sequence,

$$c_t = \sum_s a_{ts} \bar{h}_s. \tag{2.37}$$

The computation *context vector* is a key component of an attention mechanism since it should encoded the contextual information about the sequence. The proposed mechanism is shown in Figure 2.23.

We may notice that this mechanism is computationally intensive since we need to compute $n \times m$ scores for computing the context vectors. For solving this issue Luong et al. (2015) introduce a *local attention mechanism*. It consists of a more elegant way to compute the attention weights and the context vector. Many extensions (Rocktäschel et al. 2015; Shen et al. 2018; Hu 2019; Zhou et al. 2016; Shen et al. 2018; Yin et al. 2016) of global and local attentions have been explored in the fields of NLP with a focus on simplifying the attention mechanism.

The CV community has also incorporated attention mechanisms, which can be split into two categories: *hard* and *soft* attention mechanisms. The example presented below for NLP corresponds to a soft-attention mechanism. We present both categories below.

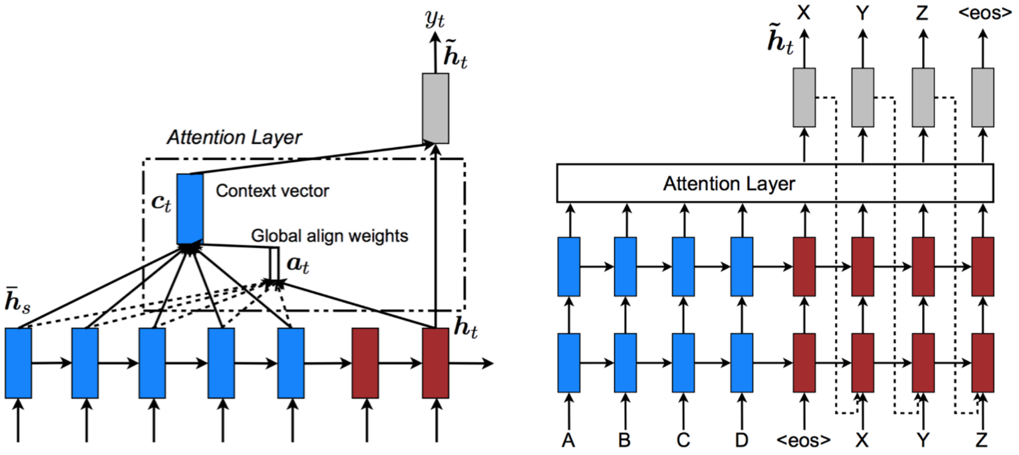


Figure 2.23 – **Attention mechanism for Machine Translation.** (Left) Computational graph for the *global* attention mechanism. – (Right) Computational graph for the *local* attention mechanism..
Figure reproduced from (Luong et al. 2015).

Hard-attention mechanism — *Hard attention* consists in choosing some parts of the input and completely omitting the remaining parts. It greatly reduces the computation time since a large part of the input data is not taken into account for solving the final task. However taking such hard decisions makes the overall pipeline not differentiable end-to-end. For training such models we need to lie on stochastic algorithms, which cannot be easily learned through gradient descent and back-propagation. In a seminal paper, Mnih et al. (2014) proposed visual hard-attention for image classification built around on RNN, which implements the policy of a virtual agent. Visualization of the proposed method is shown in Figure 2.24. A reinforcement learning problem is thus solved during learning using the REINFORCE algorithm (Williams 2012). The model selects the next location to focus on, based on past information. Similar approaches have been applied for tackling multiple object recognition (Ba et al. 2015), saliency map generation (Kuen et al. 2015), action detection (Yeung et al. 2016) and image generation (Eslami et al. 2016). Malinowski et al. (2018) demonstrate, with the Hard Attention Network, that a soft-attention mechanism can be efficiently replaced by a hard selection of multiple regions in the image that are relevant for a given question.

Soft-attention mechanism — Soft attention was proposed for image (Cho et al. 2015) and video understanding (Sharma et al. 2016; Song et al. 2016; Yeung et al. 2015), with spatial, temporal and spatio-temporal variants. The common key components of soft-attention mechanisms on visual data is to assign higher importance to the most discriminative parts of the signal (e.g. timesteps, areas in the image). Xu et al. (2015) propose to benchmark hard and soft attention mechanisms

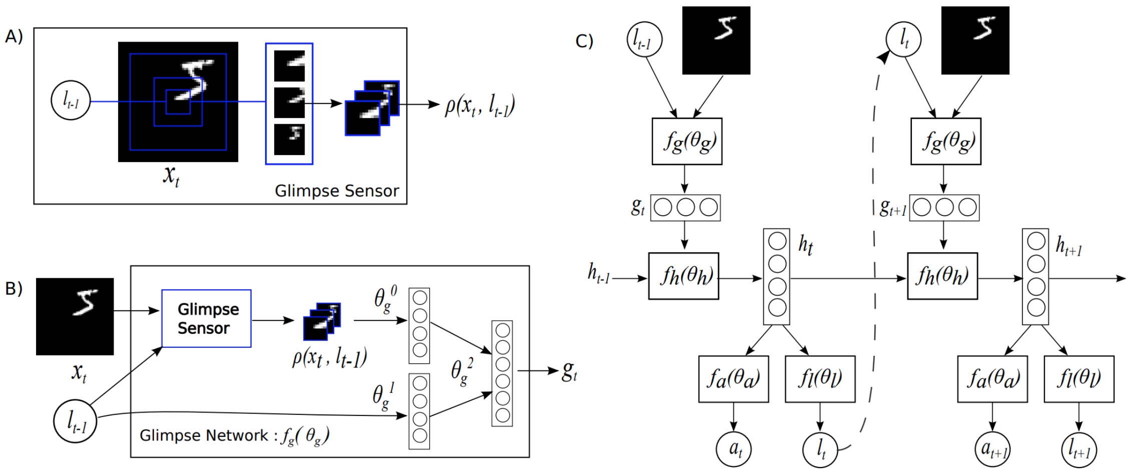


Figure 2.24 – **Hard attention mechanism for image recognition.** (**A)**) The glimpse sensor for extracting visual features at a specific location. – (**B)**) The glimpse network is associating visual as well as spatial features. – (**C)**) The overall architecture employs a recurrent system on top of the glimpse network for solving the task of image recognition in a iterative manner.
Figure reproduced from (Mnih et al. 2014).

for the task of image captioning. They show that similar performances can be achieved using both soft and hard mechanism. However, the training procedure of the sof-attention mechanism is easier and more robust. Sharma et al. (2016) develop a recurrent mechanism for action recognition from RGB data, which integrates convolutional features from different parts of a space-time volume such as shown in Figure 2.25. They propose to automatically assign more weights to the most discriminative parts of the images given the information leveraged from the past. Song et al. (2016) propose to separate spatial and temporal attention networks for action recognition from pose. At each frame, the spatial attention model gives more importance to the joints most relevant to the current action, whereas the temporal model selects most important timesteps. For the task of Visual Question Answering (VQA), several works (Fukui et al. 2016; Kim et al. 2017a; Yu et al. 2017a) propose to use multi-glimpse soft-attention mechanisms. Different glimpses can also be computed in an iterative fashion, as in the Stacked Attention Network (Yang et al. 2016). Authors show that it helps for answering multi-hop questions.

Attention and Memory — Early attention mechanisms were based on the RNN model. However, this type of network has shown problem for learning long-term dependencies. To alleviate this problem, memory networks (Sukhbaatar et al. 2015; Graves et al. 2014; Weston et al. 2015) were developed. They include an

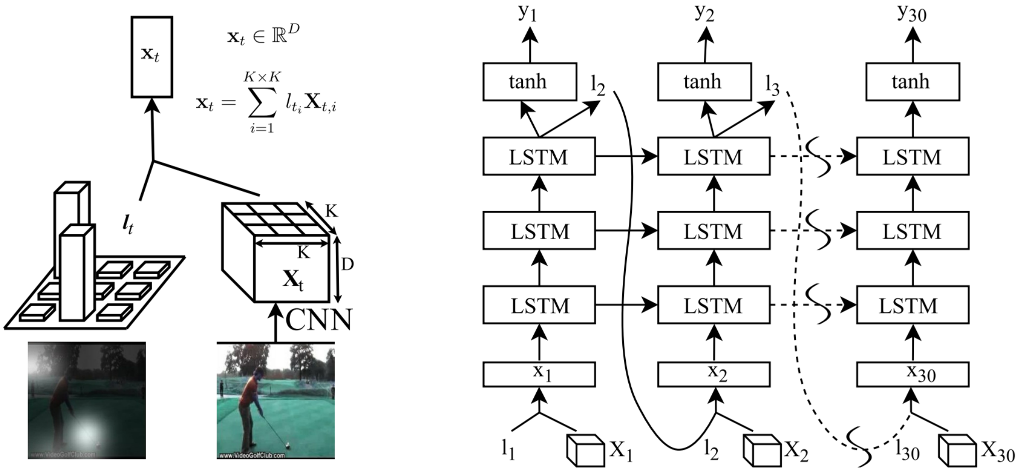


Figure 2.25 – **Soft attention mechanism for action recognition.** (Left) The proposed attention mechanism is updating the values within the feature map based on a context vector. – (Right) The overall architecture is composed of a recurrent network which is predicting a spatial attention map at each timestep.
Figure reproduced from (Sharma et al. 2016).

external memory where the system can read and write for recording an retrieving important content.

Initially, memory networks (Weston et al. 2015; Sukhbaatar et al. 2015) were proposed for NLP, especially for the task of Question Answering, which consists in reading a long document and answering questions relative to the text.

In a different area, Graves et al. (2014) propose a Neural Turing Machine which consists of a neural network controller (a LSTM) combined with an external memory. The controller interacts with the external memory through an attention mechanism system. It can performed read and write operation depending on the input and its internal state. The overview of the proposed approach is shown in Figure 2.26.

Self-attention — For solving the issue of learning long-term dependencies in sequential data, Vaswani et al. (2017) develop *self-attention* with the *transformer* architecture still in the field of NLP for the machine translation task. The transformer architecture is based solely on *attention mechanisms* and does not employ any recurrence or convolution. The overall architecture is still based on an encoder-decoder architecture where the input sequence $x = (x_1, \dots, x_n)$ is mapped to a sequence of hidden representations $z = (z_1, \dots, z_n)$, and z is input to the decoder that generates the output predictions $y = (y_1, \dots, y_m)$. Each encoder and decoder is composed of N identical layers composed respectively of a *multi-head self attention mechanism* and a positional mechanism. The *self-attention mechanism* (also called

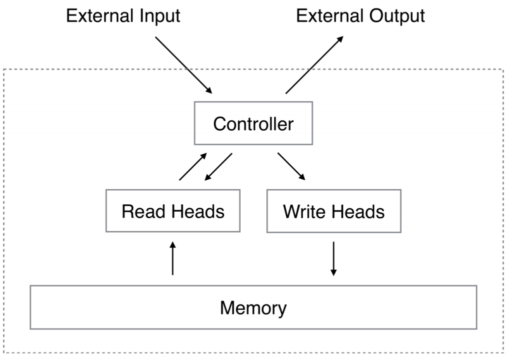


Figure 2.26 – **Overview of the Neural Turing Machine.**
At each timestep the controller may read or write into the external memory for retrieving or storing important content.
Figure reproduced from (Graves et al. 2014).

"Scaled Dot-Product" attention) is a type of attention that uses scaled dot product for computing similarity.

The attention function corresponds to mapping a query and a set of key-value pairs to an output. q, k and v are respectively called *queries*, *keys* and *values* obtained using linear mappings from input sequence x;

$$k_i = W_k x_i \tag{2.38}$$

$$v_i = W_v x_i \tag{2.39}$$

$$q_i = W_q x_i \tag{2.40}$$

where W_k, W_v and W_q are learnable matrices.

Attention weights α are obtained by computing the dot-product similarity,

$$\alpha_{ij} = \text{softmax}(\frac{q_i^\top k_j}{\sqrt{d_k}}) \tag{2.41}$$

where d_k is the vector dimension of v_i.

And finally the output vector is computed such as,

$$z_i = \sum_j \alpha_{ij} v_j. \tag{2.42}$$

The attention function can be seen as mapping a query and a set of key-value pairs to an output. In practice, there are multiple head attentions at the same stage of the network and we concatenate the output. Visualization of the self-attention mechanism and the Transformer architecture are shown in Figure 2.27.

Transformer is now the standard architecture in NLP and its applications have shown recent successes with self-supervised training using the Masked Language

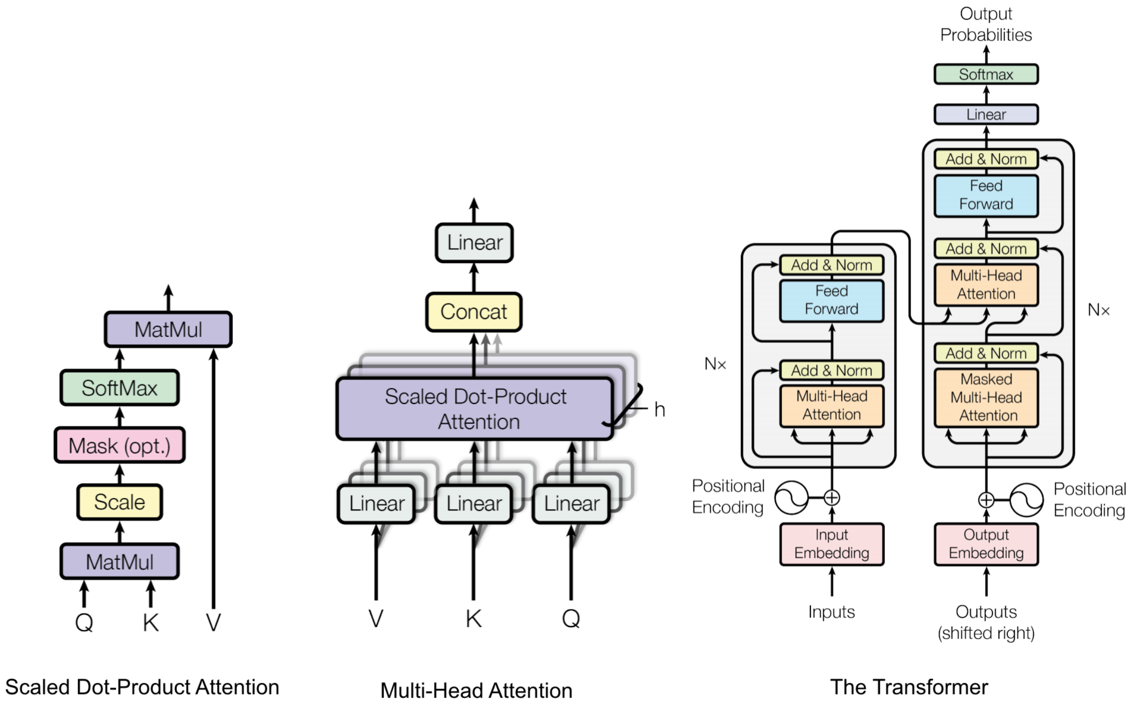


Figure 2.27 – **Self-attention and Transformer architecture.** (Left) Self-attention using the dot-product attention. – (Middle) Multi-head attention by performing several self-attention modules in parallel and concatenating their outputs. – (Right) The overall Transformer architecture that consists of stacking multiple multi-head attention layers. Figure reproduced from (Vaswani et al. 2017)

Modeling tasks (Devlin et al. 2018; Yang et al. 2019; Lan et al. 2020). Dai* et al. (2019) extend the Transformer architecture for dealing with long sequences. They incoporate a transferable memory which make the full pipeline possible to operate on sequence of any length. The success of self-attention encountered in NLP has not yet reached similar performance in CV however some works are going into this direction. Ramachandran et al. (2019) propose to replace convolution by self-attention layers. They show that self-attention can learn feature as good as the one obtained with convolution operations. Wang et al. (2018a) incorporate self-attention for updating feature map values with context information for the task of action recognition. They show that updating local feature with context information can be benificial for improving the local descriptors. Hu et al. (2018) develop the Squeeze-and-Excitation block for including global information in the decision process of the network.

2.2.7 Relational reasoning

Reasoning is a key component of human intelligence. While CV systems show excellent performance on task such as object recognition, they show limited abilities to reason from a visual content.

Humans are able to reason from the visual space by solving simple task such as understanding simple physical laws by solving intuitive physic task. But more interesting they are able to identify more abstract concepts such as human-object relations. A common practice in CV is to express visual content as a graph where each nodes corresponds to (semantic) visual entities. In this section we discuss what are common methods that are employed for achieving reasoning from such representation and their applications in tasks relying on reasoning such as VQA.

Intuitive Physics — Fundamental studies on cognitive psychology have shown that humans perform poorly when asked to reason about expected outcomes of a dynamic-based event, demonstrating striking deviations from Newtonian physics in their intuitions (McCloskey et al. 1983; McClooskey et al. 1980; McClooskey et al. 1983; Kubricht et al. 2017). The questions of approximating these mechanisms, learning from noisy observed and non-observed physical quantities (such as sizes or velocities vs masses or gravity), as well as justifying importance of explicit physical concepts vs cognitive constructs in intelligent agents, have been raised and explored in recent works on deep learning (Wu et al. 2015).

Lerer et al. (2016) train a network for predicting the stability of a block tower composed of cubes as well as the future mask segmentations. While they show great results compared to copying baselines, the model performs poorly under high-uncertainty and outputs blurry predictions. Groth et al. (2018) extend this approach by predicting the stability of towers composed of different type of objects (cubes, cylinders, balls). They present a analysis for identifying the unstable objects in the visual domain but they do not try to predict the future neither in the visual or the physical space.

Ye et al. (2018) build an interpretable intuitive physical model from visual signals using full supervision on the physical properties of each object. A two steps procedure is proposed by (Wu et al. 2017) to solve this task. First, they infer object physical properties using an object detector. Second, they predict the future object properties in the physical world. Finally, they render the objects in the visual domain using a graphical engine. Similarly but without making use of rendering engine, Zheng et al. (2018) propose to solve this task by first extracting a visual perception of the world state and then predict the future.

Beyond the identification of physical laws, more abstract concepts can be extracted from images. An example of such concept is the identification of human-object relations.

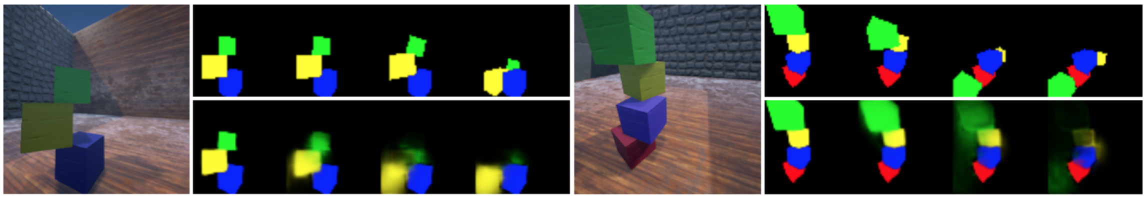


Figure 2.28 – **Learning Physical Intuition of Block Towers by Example.** In (Lerer et al. 2016), a CNN is trained at predicting the future object masks. Figure reproduced from (Lerer et al. 2016).

Visual Object Relations — Recent object detectors (Girshick 2015; Ren et al. 2015) show great performance at localizing and detecting pre-defined objects in an image. For a better fine-grained scene understanding, recent works focus on detecting object relations within an image. Understanding an image from an object relations point of view allows to solve many different tasks such as image captioning or visual question answering. Visual relationship detection is highly correlated to the action recognition since (most of the time) it corresponds to identifying human actions in their context. The relationships (also called interactions) can be seen as a triplet: t =(*subject-predicate-object*). That allows also to connect language and vision. For example in Figure 2.29 (second image from left), a *person* is *on* a *motorcycle*. *Person*, *on* and *motorcycle* corresponds respectively to the *subject*, the *predicate* and the *object*. But one image may be described by multiple visual relationships such as shown in Figure 2.29.

Recent datasets such as MS-COCO (Lin et al. 2014), Visual Genome (Krishna et al. 2017) or Visual Relationship Dataset (Lu et al. 2016) drive the field into more robust methods (Dai et al. 2017; Sadeghi et al. 2011). Approaches can be cast into two categories. The first one is composed of methods (Dai et al. 2017) that propose a compositional approach which consists of learning a separate detector for each subject, predicate and object. Whereas the other category of methods (Sadeghi et al. 2011) treat the entire triplet t as a visual phrase and train a detector for each visual phrase. Both type of approaches perform poorly when evaluated on unseen visual relations at test time. One way to solve this issue is to combine compositional and visual phrase approaches (Peyre et al. 2019). They also propose to use language priors in the word embeddings for transferring to similar objects.

In real world applications, complex relationships cannot be expressed by triplet containing only three elements. The number of elements could vary and could depend of the contextual information. A way to deal with such abstract relationships is to express visual content from a graph structure perspective where each node represent a detected visual entities.

Graph Convolution Network — Graph Convolutional Network (GCN) are a type of neural network that can be used for extracting information from graph

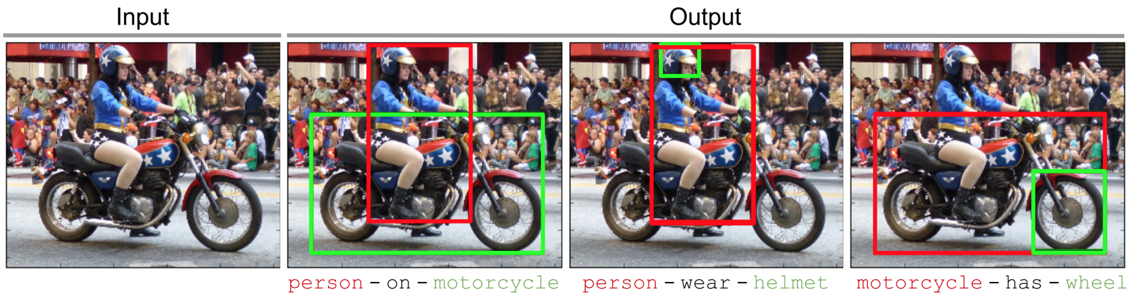


Figure 2.29 – **Example of visual relationships.**
Figure reproduced from (Lu et al. 2016).

structures (Kipf et al. 2017). This method explores relational inductive biases within deep learning architectures can facilitate learning about entities, relations, and rules for composing them. We give a quick review of GCN in the following and refer the reader to Battaglia et al. (2018) for a complete survey on this topic.

We assume a graph G defined by the triplet (V, E, u). $V = \{v_i\}_{i=1...N}$ is the set of N nodes and a node correspond to a pre-defined entity defined by its attributes v_i (e.g. position, appearance, semantic). $E = \{(e_k, r_k, s_k)\}_{k=1...M}$ is the set of edges where e_k is the edge's attribute, r_k and s_k correspond respectively to the index of the receiver and sender nodes. Most of the time there is no prior knowledge about the graph structure such that we have to assume a fully-connected graph. Finally u represents the global attribute of the system. For example, u can represent the gravitational coefficients of a physical system. An example of graph G can be seen in Figure 2.30 a) and the associated attributes in Figure 2.30 b).

The goal of a GCN (Kipf et al. 2017) is to update the node attributes by taking into account its relations with its neighbors within the global system. This updating scheme can be decomposed into three distinct steps.

First, edge attributes are updated given the nodes attributes of the receiver and sender,

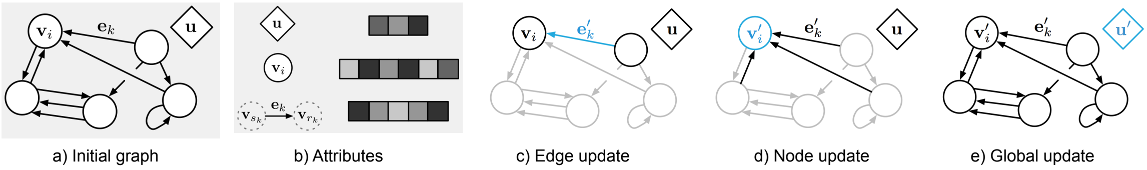
$$e'_k = f_e(e_k, v_{r_k}, v_{s_k}, u) \tag{2.43}$$

$$\bar{e}'_i = g_{e\to v}(E'_i) \tag{2.44}$$

where $E'_i = \{(e'_k, r_k, s_k)\}_{r_k=i,k=1...M}$ is the set of edges connected to the receiver node i, f_e is a MLP and $g_{e\to v}$ is an aggregation function (e.g. mean, maximum, ...). In the rest of the GCN equations, f_* and g_* are the same types of function. It is important that g_* should be invariant to inputs permutations and invariants to the number of inputs.

The second step is a *node update* which corresponds to updating the node attribute given the new edges attributes such that:

$$v'_i = f_v(\bar{e}'_i, v_i, u) \tag{2.45}$$



a) Initial graph b) Attributes c) Edge update d) Node update e) Global update

Figure 2.30 – **Graph Convolutional Network.** Figures a) and b) show the initial graph and the definition of the node attributes, edges and global attributes. Figures c), d) and e) show the three-step procedure updates respectively *edge*, *node*, *global* updates.
Figure reproduced from (Battaglia et al. 2018).

Finally we perform a *global update* which corresponds to updating the global attribute by taking into account the global nodes and edges attributes such that:

$$\bar{e}' = g_{e \to u}(E') \tag{2.46}$$

$$\bar{v}' = g_{v \to u}(V') \tag{2.47}$$

$$u' = f_u(\bar{e}', \bar{v}', u) \tag{2.48}$$

where $E' = \{(e'_k, r_k, s_k)\}_k$ and $V' = \{v'_i\}_{1\ldots N}$.

This three-step procedure is shown in Figure 2.30 c), d), and e).

Applications of Graph Convolution Network — GCN show great success ina large number of tasks. Battaglia et al. (2016) introduce a fully-differentiable network physics engine called Interaction Network for reasoning about physical systems (gravitational systems, rigid body dynamics, and mass-spring systems). Their method is able to predict the future object positions accurately by taking the object interactions into account. Moreover the method is able to generalize to unseen numbers of objects. However this system assumes ground truth object properties such as mass, gravity, friction coefficient. Recent approaches (Chang et al. 2017; Janner et al. 2019; Battaglia et al. 2018; Veličković et al. 2018) based on GCN (Kipf et al. 2017) have shown promising results on learning physics but are restricted to setups where physical properties need to be fully observable.For solving this issue, Steenkiste et al. (2018) focus on discovering objects and their interactions in a unsupervised manner from a virtual environment from raw visual images. The proposed approach is able to handle occlusion and to extrapolate to different numbers of objects.

Visual Question Answering on Synthetic Data — VQA is an active topic which deals with reasoning about the visual input. The task consists of answering a question given the visual content provided by an input image. Many recent works (Hu et al. 2017; Hudson et al. 2018a; Johnson et al. 2017; Mao et al. 2019; Perez et al. 2018; Santoro et al. 2017) focus on synthetic data since the dataset bias can be more controlled compared to real data. Relation Network (RN) (Santoro et al. 2017)

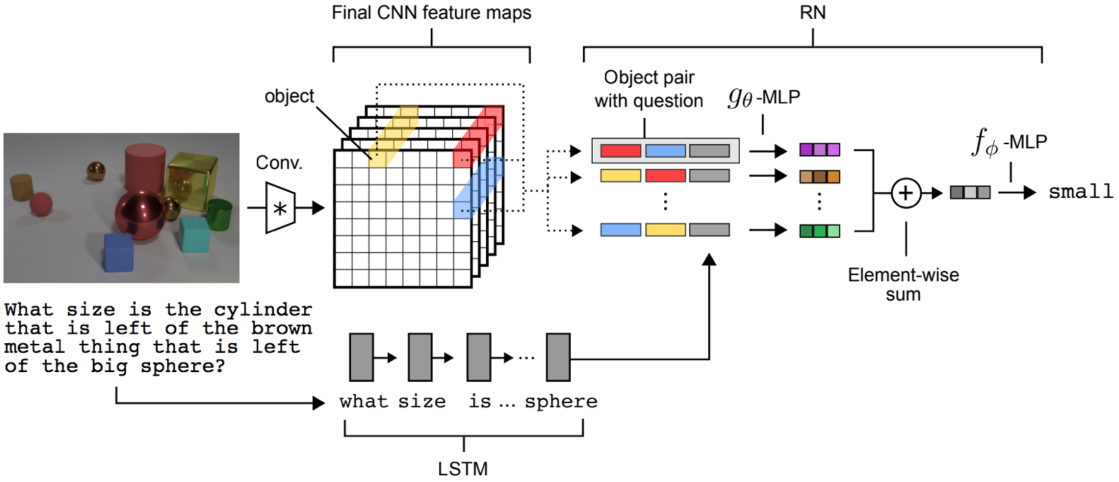


Figure 2.31 – **Relation Network for VQA.** Architecture proposed by Santoro et al. (2017). A CNN is extracting local visual features while a LSTM is infering a text representation. Solving the VQA task is done by learning the relations between visual and textual features using a GCN.
Figure reproduced from (Santoro et al. 2017).

can be seen as a fully-differentiable trainable layer for reasoning in deep networks. It is a specific and simplified version of GCN where the nodes correspond to each discrete cell in the feature map produced by the CNN. Figure 2.31 shows an overview of the proposed model. From a different perspective, Perez et al. (2017) show that relational reasoning can be learned for visual reasoning in a data driven way without any prior. They use conditional batch normalization with a feature-wise affine transformation based on conditioning information. In an opposite approach, Hudson et al. (2018b) focus on learning a strong structural prior in the form of a complex attention mechanism. An external memory module combined with attention processes over input images and text questions, performing iterative reasoning. Finally for going beyond the task of VQA on synthetic data, Santoro et al. (2018) build a challenging dataset for solving the problem of abstract reasoning on the visual domain. They focus on tasks that require interpolation or extrapolation mechanisms.

Visual Question Answering on Real Data — Answering relational questions on real data needs understanding the relationships between objects present in the image. Initially methods based on attention showed great performance by integrating multiple attention maps (Yu et al. 2017b; Ben-Younes* et al. 2017). A more structured attention mechanism is employed by Chen et al. (2017b). They generate a locally-connected graphical structure for inferring region saliency scores. Teney et al. (2017) incorporate the structure in the scene and in the

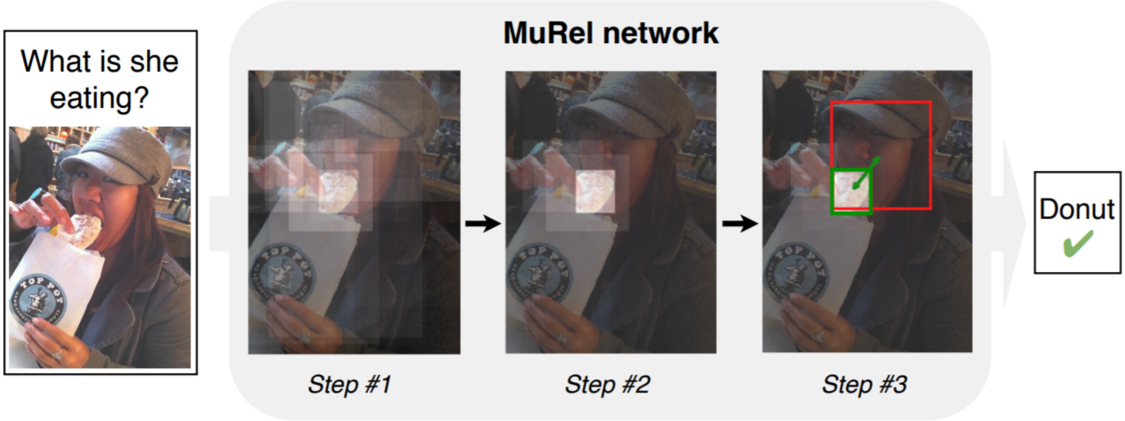


Figure 2.32 – **Visualization of the MuRel approach.** MuRel is an iterative process which is at each step refining the information extracted from the image. Three steps are shown in this figure.
Figure reproduced from (Ben-Younes* et al. 2019).

question. They first build a graph over the scene objects and the question words. And then they employ this structure for solving the task. Following the same line, Li et al. (2019) first detect objects and then propose to learn the most important object relationships using a GCN with attention mechanism. Ben-Younes* et al. (2019) propose to use bilinear fusion methods for better modelling pairwise object relationships. They also employ an iterative process which allows to refine the visual information extracted from the image. Figure 2.32 shows an example of this procedure. Following another line of work, Hudson et al. (2019) propose to predict a probabilistic graph given an image which serves as a structured world model and then to perform sequential reasoning over the graph. This modeling allow the model to operates on abstract latent space since transform the visual and linguistic modalities into semantic concept-based representations.

Recent success of such well structured methods on VQA has been extended to other domain where the need of structuring the incoming information is of high interest as well.

Reasoning in videos — While most of the works on visual reasoning tackle problems which consists of analysing singe image, moving to video allow to incorporate the temporal information for a better understanding of the scene. Reasoning in videos on a mask or segmentation level is attempted for video prediction (Luc et al. 2017), where the goal is to leverage semantic information to be able predict further into the future. For the task of action recognition, Bolei et al. (2017) propose Temporal Relation Network which is an extension of RN for video stream. They propose to model the frames relations from the visual embedding. They deploy relationships at different time scales such as two-frames or three-

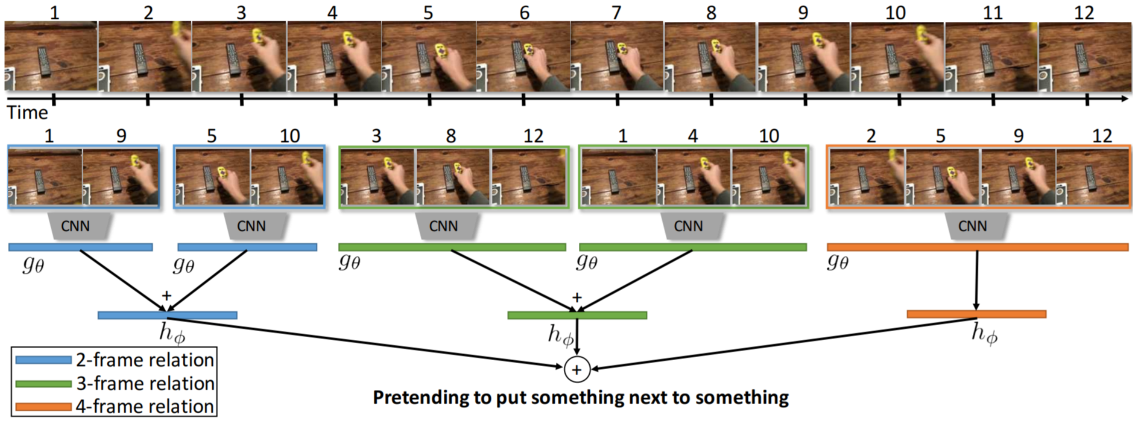


Figure 2.33 – **Illustration of the Temporal Relation Network.** Bolei et al. (2017) propose to model frame relations. They go from two frames to four frames relations. Only a subset of the frame relations are shown in this figure.
Figure reproduced from (Bolei et al. 2017).

frames relations. Figure 2.33 shows an overview of the system. Detected persons and objects are used for building a spatio-temporal graph structure in videos (Sun et al. 2018; Wang et al. 2018b) with the final goal to classify the video. Wang et al. (2018b) show that the representation learned from the graph structure could be befinicial in addition to the global information. In a similar line, a gated energy function parametrization that learns adaptive relations conditioned on visual observations is proposed by Tsai et al. (2019). They propose an computationally efficient optimization strategy.

GCN are also employed for skeleton-based action recognition (Yan et al. 2018; Shi et al. 2019). Yan et al. (2018) build a spatio-temporal graph from the detected human joints. They show that they can learn good representation from ground truth human pose as well as noisy an incomplete detected human joints. Shi et al. (2019) extend this work by proposing a two-stream adapative GCN by modeling both the first and second orders information simultaneously. They propose a method where the topology of the graph can be either uniformly or individually learned in a data-driven way which brings more generality for adapting to different data samples.

In the next chapter, we will present the four contributions of this thesis. In Chapter 3 and Chapter 4, we propose new soft-attention mechanisms for identifying human actions in videos. First, in Chapter 3 we introduce a method that use human pose information for automatically drawing attention to pre-defined points of interest in the video. Second, we go one step further in Chapter 3 by using the contextual information for selecting locations of interest in the video in a free manner. In Chapter 5, we propose to model videos as a space-time graphes where each node represents an object. We introduce a method based on space-time object interactions for classifying video content. Finally in Chapter 6 we go beyond supervised learning task from video content and propose to tackle counterfactual learning from high-dimensional data.

Chapter 3

HUMAN ACTIVITY RECOGNITION WITH POSE-DRIVEN ATTENTION TO RGB

Chapter abstract

Articulated human pose carries useful information for human action recognition. However, this type of information is limited for modeling fine-grained actions, where gathering visual cues is crucial.

In this chapter, we propose a method based on articulated pose and RGB data for addressing the human action recognition task. We process the pose stream using a Convolutional Neural Network (CNN) for extracting spatio-temporal features with a specific joint ordering. Moreover, we introduce a spatio-temporal soft mechanism conditioned on pose features for extracting discriminative visual cues over a set of pre-defined image locations: the four hands. Appearance features give important cues on hand motion and on objects held in each hand. We show that it is of high interest to shift the attention to different hands at different time steps depending on the activity itself.

We experiment our method on the largest dataset for human activity recognition, namely NTU RGB+D, where we show state-of-the-art results. In addition we also show transfer learning results on two small datasets, MSR Daily Activity and SBU Interaction Dataset.

The work in this chapter has led to the publication of conference papers:

- Fabien Baradel, Christian Wolf, and Julien Mille (2017b). "Human Action Recognition: Pose-based Attention draws focus to Hands". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV) - Workshop "Hands in Action";*

- Fabien Baradel, Christian Wolf, and Julien Mille (2018b). "Human Activity Recognition with Pose-driven Attention to RGB". in: *Proceedings of the British Machine Vision Conference (BMVC).*

Contents

3.1 Introduction

Videos contain a high amount of information which makes them challenging to process for extracting the underlying semantical meaning. As described in Section 2.2.5, recognizing human actions in videos remains a challenging task and current methods focus on modeling global context, which is clearly not sufficient for discriminating fine-grained human actions. One way to extract meaningful information about the video content is to use human pose data that can be captured from consumer depth cameras and that can help at structuring the visual content. In this chapter we assume having access to this type of information using RGB-D cameras. Recent works (Liu et al. 2017b; Shahroudy et al. 2016a) using skeleton data only have shown great success in standard benchmarks such as shown in Figure 2.2.5. However, they are by definition restricted to human behavior understanding only, and cannot extract information about objects that humans are interacting with. To solve this issue, a common way is to downsample the video stream and the full resolution at certain positions only, that may help extracting important cues on small or distant objects (or people). In this regard, models of visual attention (Mnih et al. 2014; Cho et al. 2015; Sharma et al. 2016) have drawn considerable interest recently as described in Section 2.2.6. Since these models are able to focus their attention to specific important points, their parameters are not wasted on input parts which are considered of low relevance to the task at hand.

In this chapter, we present a method for human activity recognition, which addresses this problem by using articulated pose and raw RGB input in a novel way: our method attends to some parts of the RGB stream given information from the pose stream. In our approach, pose has three complementary roles:

- It is used as an input stream in its own right, providing important cues for the discrimination of activity classes.
- Raw pose, made up of body joints, serves as an input for the model handling the RGB stream, selecting positions where glimpses are taken in the image.
- Features learned on pose serve as an input to the soft-attention mechanism, which weights each glimpse output according to an estimated importance w.r.t. the task at hand, in contrast to unconstrained soft-attention on the RGB video (Sharma et al. 2016). We give an overview of our model in Figure 3.1.

The RGB stream model is recurrent (a Long-Short Term Memory (LSTM)), whereas our pose representation is learned using a Convolutional Neural Network (CNN) taking as input a sub-sequence of the video. The benefits are twofold: a pose representation over a large temporal range allows the attention model to assign an estimated importance for each glimpse point and each time instant taking into account knowledge of this temporal range. As an example, the pose stream might indicate that one of the person's hand moves towards another person, which still leaves several possible choices for the activity class. These choices might require attention to be moved to this hand at a specific instant to verify what kind of object is held, which itself may help to discriminate activities.

The contributions of this chapter are as follows:

- We propose a spatial attention mechanism in Section 3.3.1 on RGB videos which is conditioned on deep pose features from the full sub-sequence.
- We propose a temporal attention mechanism in Section 3.3.2 which learns how to pool features output from the recurrent (LSTM) network over time in an adaptive way.
- As an additional contribution, we experimentally show in Section 3.4 that knowledge transfer from a large activity dataset like NTU (57'000 activities) to smaller datasets like SBU Interaction Dataset 3D (300 videos) or MSR Daily Activity (300 videos) is possible.

3.2 Related work

Activities, gestures and multimodal data — Recent gesture/action recognition methods dealing with several modalities typically process 2D+T RGB and/or depth data as 3D. Sequences of frames are stacked into volumes and fed into convolutional layers at first stages (Baccouche et al. 2011; Ji et al. 2013; Molchanov et al. 2016; Neverova et al. 2016; Wu et al. 2016) such as discussed in Section 2.2.5. When additional pose data is available, the 3D joint positions are typically fed

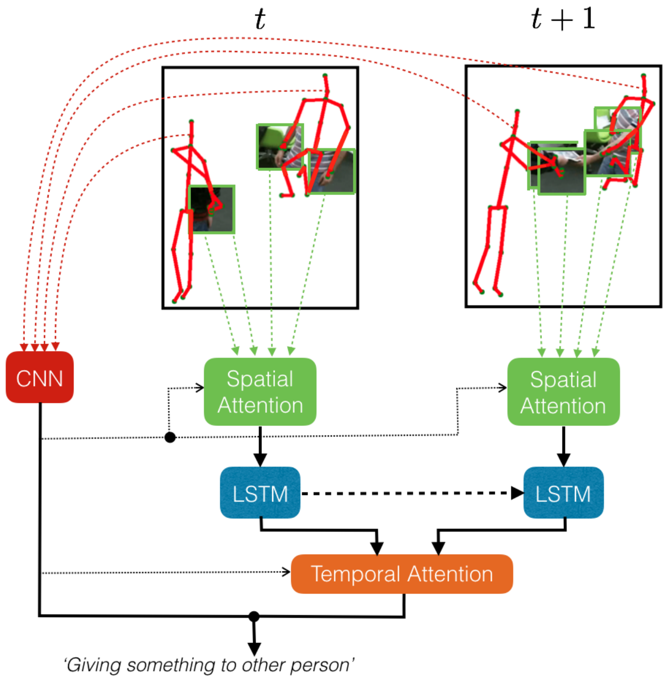


Figure 3.1 – **Overview of the proposed pose-driven model.** We recognize human activities fusing a model trained on pose sub-sequences and a spatio-temporal attention model on RGB video conditioned on pose features

into a separate network. Preprocessing pose is reported to improve performance in some situations. Examples of preprocessing are the augmention of pose coordinates with velocities and accelerations (Zanfir et al. 2013), or the normalization of bone lengths and view point (Neverova et al. 2016). Fusing pose and raw video modalities can be done in a late stage, e.g averaging softmax scores output by several subnetworks, as in (Molchanov et al. 2016), or in an earlier through multiple fusion layers (Wu et al. 2016). We believe that information extracted from different modalities are complementary but at the same time redundant. Our approach addresses this issue by using learned features from one modality (pose) to attend to some part of another modality (RGB). Hence it can attend to some parts of the RGB stream which give discriminative features that can be detected from the pose data.

Architectures for pose data — Recent fine-grained activity recognition methods using pose data are based on recurrent neural networks and/or convolutional neural networks.

Regarding recurrent networks, part-aware LSTMs (Shahroudy et al. 2016a) separate the memory cells of LSTM networks (Hochreiter et al. 1997) into part-based sub-cells and let the network learn long-term representations individually for each part, fusing the parts for output. Similarly, (Du et al. 2015b) use bi-directional LSTM layers which fit anatomical hierarchy. Skeletons are split into anatomically-relevant parts (legs, arms, torso, *etc*), so that each subnetwork in the first layers gets specialized on one part. Features are progressively merged as they pass through layers. Multi-dimensional LSTMs (Graves et al. 2009) are models with multiple recurrences from different dimensions. Originally introduced for images, they also have been applied to activity recognition from pose sequences (Liu et al. 2016a). One dimension is time, the second is a topological traversal of the joints in a bidirectional depth-first search, which preserves the neighborhood relationships in the graph.

On the other side, convolutional architectures are used from pose data. CNN (Ke et al. 2017; Hou et al. 2016; Wang et al. 2016b) are also used to handle pose sequences. Such approaches require a 3D tensor as input. To satisfy this condition, they encode the sequence of pose as a trajectory (Ke et al. 2017) or into a RGB-like image for benefiting from an Imagenet initialization of the weights (Hou et al. 2016). Our solution is close to (Wang et al. 2016b), which stacks the 3D coordinates into a Tensor. However, we follow a topological ordering to extract a better representation of the pose sequence.

3.3 Model

A single or multi-person activity is described by a sequence of two modalities: the set of RGB input images $\boldsymbol{I} = \{I_t\}$, the set of articulated human poses $\boldsymbol{x}=\{\boldsymbol{x}_t\}$ and we wish to predict the activity class y. We do not use raw depth data in our method, although the extension would be straightforward. Both signals are indexed by time t. Poses $\boldsymbol{x}_t$ are defined by 3D coordinates of K joints per subject, for instance delivered by the middleware of a depth camera. In our case we restrict our application to activities involving one or two people and their interactions. We propose a two-stream model, which classifies activity sequences by extracting features from articulated human poses and RGB frames jointly. Our main contribution comes from the fact that we use features learned from the pose stream to attend to some parts of the RGB stream where all the features are end-to-end learnable.

3.3.1 Spatial Attention on RGB videos

The sequence of full-HD RGB input images $\{\boldsymbol{I}_t\}$ is arguably not compact enough to easily extract an efficient global representation with a feed-forward neural network. We opt for a recurrent solution, where, at each time instant, glimpses on the seen input is selected using an attention mechanism.

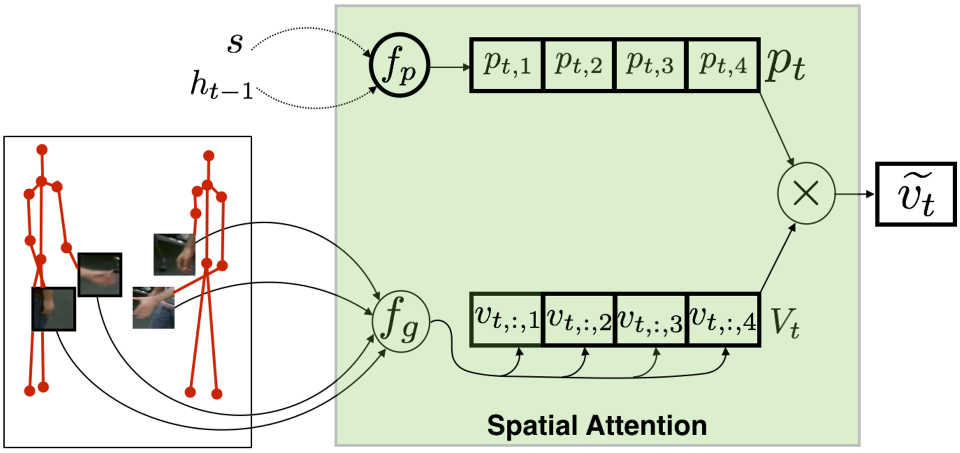


Figure 3.2 – **The spatial attention mechanism.** We propose to use the pose features for drawing the attentions on the set of available hands.

In some aspects similar to (Mnih et al. 2014), we define a trainable bandwidth limited sensor. However, in contrast to (Mnih et al. 2014), our attention process is conditional to the pose input $\boldsymbol{x}_t$, thus limited to a set of N discrete attention points. In our experiments, we selected N=4 attention points, which are the 4 hand joints of the two people involved in the interaction. We choose hands as attention points because humans use their hands for performing most of their daily actions. Our method can be extended to more attention points. The goal is to extract additional information about hand shape and about manipulated objects. Many activities such as *Reading*, *Writing*, *Eating*, *Drinking* are similar in motion but can be highly correlated to manipulated objects. As the glimpse location is not output by the network, this results in a differentiable soft-attention mechanism, which can be trained by gradient descent.

The glimpse representation for a given attention point i is a CNN f_g with parameters θ_g, taking as inputs a crop taken from image I_t at the position of joint i from the set $\boldsymbol{x}_t$:

$$\boldsymbol{v}_{t,:,i} = f_g(\text{crop}(I_t, \boldsymbol{x}_t, i); \theta_g) \qquad i \in \{1, \ldots, N\} \tag{3.1}$$

Here and in the rest of the chapter, subscripts of mappings f_g and their parameters θ_g choose a specific mapping, they are not indices. Subscripts of variables and tensors are indices. $\boldsymbol{v}_{t,:,i}$ is a (column) feature vector for time t and hand i. For a given time t, we stack the vectors into a matrix $\boldsymbol{V}_t$=$[\boldsymbol{v}_{t,j,i}]_{i,j}$, where i is the index over hand joints and j is the index over features. $\boldsymbol{V}_t$ is a 2D tensor, since t is fixed for a given instant.

A recurrent model receives inputs from the glimpse sensor sequentially and models the information from the seen sequence with a componential hidden state $\boldsymbol{h}_t$:

$$\boldsymbol{h}_t = f_h(\boldsymbol{h}_{t-1}, \tilde{\boldsymbol{v}}_t; \theta_h) \tag{3.2}$$

We chose a fully gated LSTM model including input, forget and output gates and a cell state. To keep the notation simple, we omitted the gates and the cell state from the equations. The input to the LSTM network is the context vector $\tilde{\boldsymbol{v}}_t$, defined further below, which corresponds to an integration of the different attention points (hands) in $\boldsymbol{V}_t$.

An obvious choice of integration are simple functions like sum and concatenation. While the former tends to squash feature dynamics by pooling strong feature activations in one hand with average or low activations in other hands, the latter leads to high capacity models with low generalization. The soft-attention mechanism dynamically weights the integration process through a distribution $\boldsymbol{p}_t$, determining how much attention hand i needs with a calculated weight $\boldsymbol{p}_{t,i}$. In contrast to unconstrained soft-attention mechanisms on RGB video (Sharma et al. 2016), our attention distributions not only depend on the LSTM state $\boldsymbol{h}$, but also on the pose features $\boldsymbol{s}$ (explained in Section 3.3.3) extracted from the sub-sequence, through a learned mapping with parameters θ_p:

$$\boldsymbol{p}_t = f_p(\boldsymbol{h}_{t-1}, \boldsymbol{s}; \theta_p) \tag{3.3}$$

Attention distribution $\boldsymbol{p}_t$ and features $\boldsymbol{V}_t$ are integrated through a linear combination as

$$\tilde{\boldsymbol{v}}_t = \boldsymbol{V}_t \boldsymbol{p}_t \ , \tag{3.4}$$

which is input to the LSTM network at time t (see Equation 3.2). The conditioning on the pose features in Equation 3.3 is important, as it provides valuable context derived from motion. Note that the recurrent model itself (Equation 3.2) is not conditional (Mikolov et al. 2016), this would significantly increase the amount of parameters.

3.3.2 Temporal Attention

Recurrent models can provide predictions for each time step t. Most current work in sequence classification proceeds by temporal pooling of these predictions, e.g. through a sum or average (Sharma et al. 2016). We show that it can be important to perform this pooling in an adaptive way. In recent work on dense activity labelling, temporal attention for dynamical pooling of LSTM logits has been proposed (Yeung et al. 2015). In contrast, we perform temporal pooling directly at feature level. In particular, at each instant t, features are calculated by a learned mapping given the current hidden state:

$$\boldsymbol{u}_{:,t} = f_u(\boldsymbol{h}_t; \theta_u) \tag{3.5}$$

The features for all instants t of the sub-sequence are stacked into a matrix $\boldsymbol{U}{=}\{\boldsymbol{u}_{j,t}\}$, where j is the index over the feature dimension. A temporal attention distribution $\boldsymbol{p}'$ is predicted through a learned mapping. To be efficient, this

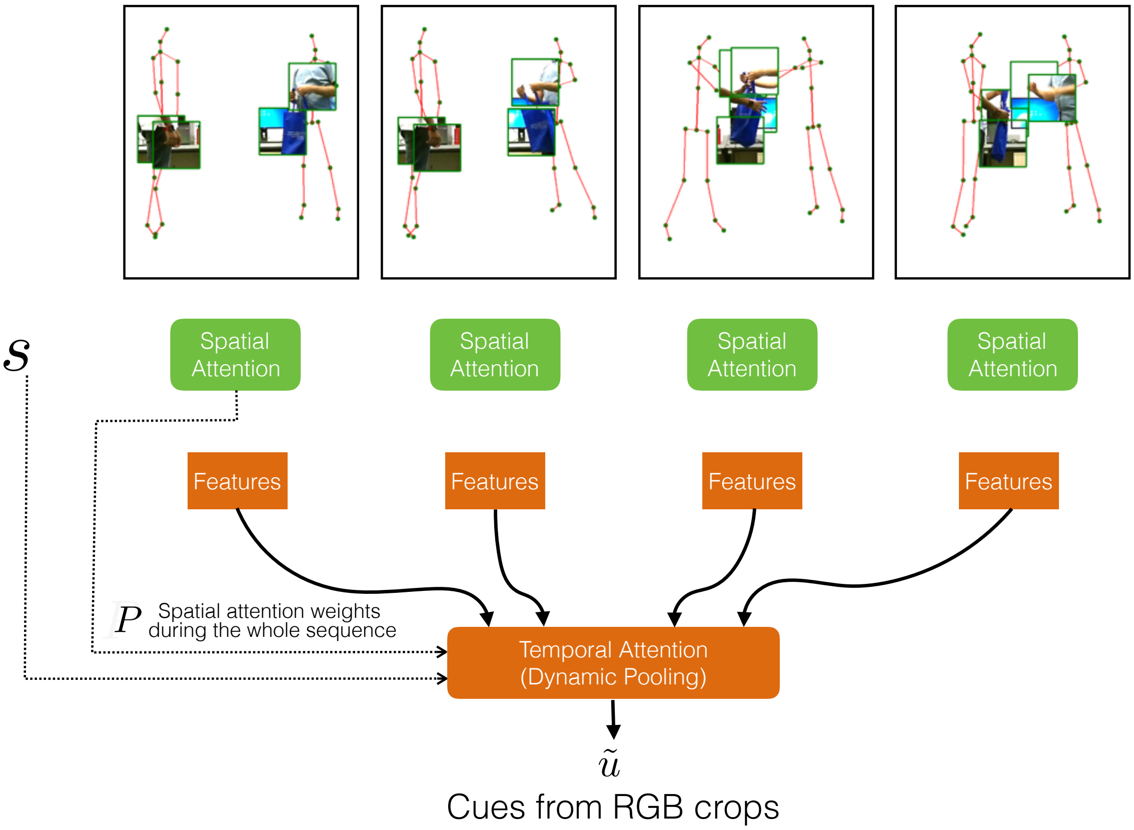


Figure 3.3 – **Temporal attention mechanism.** We use the spatial attention weights as well as the pose features for attenting the most relevant part of the sequence features.

mapping should have seen the full sub-sequence before giving a prediction for an instant t, as giving a low weight to features at the beginning of a sequence might be caused by the need to give higher weights to features at the end. In the context of sequence-to-sequence alignment, this has been addressed with bi-directional recurrent networks (Bahdanau et al. 2014). To keep the model simple, we benefit from the fact that (sub) sequences are of fixed length and that spatial attention information is already available. We conjecture that (combined with pose) the spatial attention distributions $\boldsymbol{p}_t$ over time t are a good indicator for temporal attention, and stack them into a single vector $\boldsymbol{P}$, input into the network predicting temporal attention:

$$\boldsymbol{p}' = f_{p'}(\boldsymbol{P}, \boldsymbol{s}; \theta'_p) \tag{3.6}$$

This attention is used as weight for adaptive temporal pooling of the features $\boldsymbol{U}$, i.e. $\tilde{\boldsymbol{u}} = \boldsymbol{U} \times \boldsymbol{p}'$. A visual explanation of the temporal attention is given in Figure 3.3.

3.3.3 Convolutional pose features

Given the K body joints, we wish to extract features which model i) the temporal behaviour of the pose(s) and ii) correlations between different joints. An attention mechanism on poses could have been an option, similar to (Song et al. 2016). We argue that the available pose information is sufficiently compact to learn a global representation and show that this is efficient. In our case, attention is performed on RGB conditioned on pose instead, as described earlier. We also argue for the need to find a hierarchical representation which respects the spatio-temporal relationships of the data. In the particular case of pose data, joints also have strong neighbourhood relationships in the human body.

In the lines of (Liu et al. 2016a), we define a topological ordering of the joints in a human body as a connected cyclic path over joints. The path itself is not Hamiltonian as each node can be visited multiple times: once during a forward pass over a limb, and once during a backward pass over the limb back to the joint it is attached to. The double entries in the path are important, since they ensure that the path preserves neighbourhood relationships.

In (Liu et al. 2016a), a similar path is used to define an order in a multi-dimensional LSTM network. In contrast, we propose a CNN which takes three-dimensional inputs (tensors) calculated by concatenating pose vectors over time. In particular, input tensors $\boldsymbol{X}$ are defined as $\boldsymbol{X}=\{\boldsymbol{X}_{t,j,k}\}$, where t is the time index, j is the joint & coordinate index, and k is a feature index: each line corresponds to a time instant; the first three columns correspond to the x, y and z coordinates of the first joint, followed by the x, y and z coordinate of the second joint, which is a neighbour of the first etc. The first channel corresponds to raw coordinates, the second channel corresponds to first derivates of coordinates (velocities), the third channel to second derivates (accelerations). Poses of two people are stacked into a single tensor along the second dimension.

We learn a pose network f_{sk} with parameters θ_{sk} on this input, resulting in the pose feature representation $\boldsymbol{s}$:

$$\boldsymbol{s} = f_{sk}(\boldsymbol{X}; \theta_{sk}) \tag{3.7}$$

f_{sk} is implemented as a CNN alternating convolutions and max-pooling.

3.3.4 Stream fusion

Each stream, pose and RGB, leads to its own set of features, with the particularity that pose features $\boldsymbol{s}$ are input to the attention mechanism for the RGB stream. We first train the pose stream and then the RGB stream. The final model fuse both streams on logit level. More sophisticated techniques, which learn fusion (Neverova et al. 2016), do not seem to be necessary.

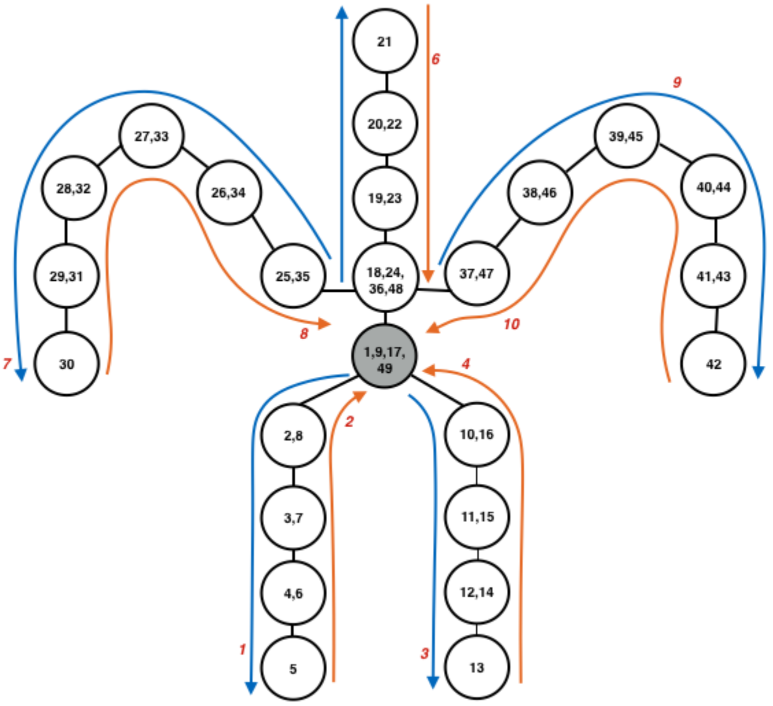


Figure 3.4 – **Topological ordering of joints.** blue arrows visit joints for the first time and orange arrows go back to the "middle spine".

3.3.5 Training

Architectures — The pose network f_{sk} consists of 3 convolutional layers of respective sizes 8×3, 8×3, 5×75. Inputs are of size $20\times300\times3$ and feature maps are, respectively, 10×150, 5×75 and $1\times1\times1024$. Max pooling is employed after each convolutional layer, activations are Rectified Linear Unit (ReLU). The glimpse sensor f_g is implemented as an Inception V3 network (Szegedy et al. 2016). Each vector $\boldsymbol{v}_{t,:,i}$ corresponds to the last layer before output and is of size 2048. The LSTM network f_h has a single recurrent layer with 1024 units. The spatial attention network f_p is an Multi-Layer Perceptron (MLP) with a single hidden layer of 256 units and sigmoid activation. The temporal attention network f'_p is an MLP with a single hidden layer of 512 units and sigmoid activation. The feature extractor f_u is a single linear layer with ReLU activation. The output layers of both stream representations are linear layers followed by softmax activation. The full model (without glimpse sensor f_g) has 38 millions trainable parameters.

Training — All classification outputs are softmax activated and trained with cross-entropy loss. The glimpse sensor f_g is trained on the ILSVRC 2012 data (Russakovsky et al. 2015). The pose learner is trained discriminatively with an additional linear+softmax layer to predict action classes. The RGB stream model is trained with pose parameters θ_{sk} and glimpse parameters θ_g frozen.

Implementation details — Following (Shahroudy et al. 2016a), we cut videos into sub sequences of 20 frames and sample sub-sequences. During training a single sub-sequence is sampled, during testing we sample 10 sub-sequences and average the logits. We apply a normalization step on the joint coordinates by translating them to a body centered coordinate system with the "middle of the spine" joint as the origin. If only one subject is present in a frame, we set the coordinates of the second subject to zero. We crop sub images of static size on the positions of the hand joints (50×50 for NTU, 100×100 for SBU and MSR). Cropped images are then resized to 299×299 and fed into the Inception model.

Training is done using the Adam Optimizer (Kingma et al. 2015) with an initial learning rate of 0.0001. We use minibatches of size 64 and dropout with a probability of 0.5. Following (Shahroudy et al. 2016a), we sample 5% of the initial training set as a validation set, which is used for hyper-parameter optimization and for early stopping. All hyper-parameters have been optimized on the validation sets of the respective datasets. When transferring knowledge from NTU to SBU, the target networks were initialized with models pre-trained on NTU. Skeleton definitions are different and were adapted. All layers were finetuned on the smaller datasets with an initial learning rate 10 times smaller then the learning rate for pre-training.

Runtime — For a sub-squence of 20 frames, we get the following runtimes for a single Titan-X (Maxwell) Graphics Processing Unit (GPU) and an i7-5930 CPU: A full prediction from features takes 1.4ms including pose feature extraction. This does not include RGB pre-processing, which takes additional 1sec (loading Full-HD video, cropping sub-windows and extracting Inception features). Classification can thus be done close to real-time. Fully training one model (w/o Inception) takes ~4h on a Titan-X GPU. Hyper-parameters have been optimized on a computing cluster with 12 Titan-X GPUs. The proposed model has been implemented in Tensorflow.

3.4 Experiments

3.4.1 Comparison with leading methods

Datasets — The proposed method has been evaluated on three datasets: NTU RGB+D (NTU), SBU Kinect Interaction (SBU) and MSR Daily Activity (MSR). NTU (Shahroudy et al. 2016a) is the largest dataset for human activity recognition with 56K videos and 60 different activities. We follow the cross-subject and cross-view

Methods	Pose	RGB	CS	CV	Avg
Part-aware LSTM (Shahroudy et al. 2016a)	X	-	62.9	70.3	66.6
ST-LSTM + TrustG. (Liu et al. 2016a)	X	-	69.2	77.7	73.5
STA-LSTM (Song et al. 2016)	X	-	73.4	81.2	77.2
GCA-LSTM (Liu et al. 2017a)	X	-	74.4	82.8	78.6
JTM (Wang et al. 2016b)	X	-	76.3	81.1	78.7
MTLN (Ke et al. 2017)	X	-	79.6	84.8	82.2
VA-LSTM (Zhang et al. 2017)	X	-	79.4	87.6	83.5
View-invariant (Liu et al. 2017c)	X	-	80.0	87.2	83.6
DSSCA - SSLM (Shahroudy et al. 2016b)	X	X	74.9	-	-
C3D† (Tran et al. 2015)	-	X	63.5	70.3	66.9
Resnet50+LSTM†	-	X	71.3	80.2	75.8
Ours (pose only)	X	-	**77.1**	**84.5**	**80.8**
Ours (RGB only)	○	X	**75.6**	**80.5**	**78.1**
Ours (pose +RGB)	X	X	**84.8**	**90.6**	**87.7**

Table 3.1 – **Results on the NTU RGB+D dataset.** With Cross-Subject (CS) and Cross-View (CV) settings (accuracies in %); († indicates method has been re-implemented).

split protocol from (Shahroudy et al. 2016a). We extensively tested on NTU and we shows two transfer experiments on smaller datasets SBU and MSR. SBU is an interaction dataset features with two people with a total of 282 sequences and 8 activities while MSR is an daily action dataset features with one people with a total of 320 videos and 16 actions. We follow the standard experimental protocols of (Yun et al. 2012) and (Wang et al. 2012) respectively for SBU and MSR.

Comparisons to the State Of The Art (SOTA) — We show comparisons of our model against the SOTA methods in Table 3.1, Table 3.3 and Table 3.2 respectively. At the time of the publication of this work, we achieved state of the art performance on the NTU dataset with the full model fusing both streams.

We also show a good generalization of our model by showing competitive results on SBU and MSR.

We conducted extensive ablation studies to understand the impact of our design choices.

3.4.2 Ablation studies and further analysis

Joint ordering — The joint ordering in the input tensor $\boldsymbol{X}$ has an effect on performance, as shown in Table 3.4. Following the topological order described in Section 3.3.3 gains >1.6 percentage point on the NTU dataset w.r.t. random joint

Methods	Pose	RGB	Depth	Acc.
Raw skeleton (Yun et al. 2012)	X	-	-	49.7
Joint feature (Yun et al. 2012)	X	-	-	80.3
Raw skeleton (Yun et al. 2014)	X	-	-	79.4
Joint feature (Yun et al. 2014)	X	-	-	86.9
Co-occurence Recurrent Neural Network (RNN) (Zhu et al. 2016)	X	-	-	90.4
STA-LSTM (Song et al. 2016)	X	-	-	91.5
ST-LSTM + Trust Gate (Liu et al. 2016a)	X	-	-	93.3
DSPM (Lin et al. 2015)	-	X	X	93.4
VA-LSTM (Zhang et al. 2017)	-	X	X	97.5
Ours (Pose only)	X	-	-	90.5
Ours (RGB only)	○	X	-	72.0
Ours (Pose + RGB)	X	X	-	94.1

Table 3.2 – **Results on SBU Kinect Interaction dataset.** Accuracies in %.

Methods	Pose	RGB	Depth	Acc.
Action Ensemble (Wang et al. 2012)	X	-	-	68.0
Efficient Pose-Based (Eweiwi et al. 2014)	X	-	-	73.1
Moving Pose (Zanfir et al. 2013)	X	-	-	73.8
Moving Poselets (Tao et al. 2015)	X	-	-	74.5
MP (Shahroudy et al. 2016b)	X	-	-	79.4
Depth Fusion (Zhu et al. 2015)	-	-	X	88.8
MMMP (Shahroudy et al. 2016b)	X	-	X	91.3
DL-GSGC (Luo et al. 2013)	X	-	X	95.0
DSSCA - SSLM (Shahroudy et al. 2016b)	-	X	X	97.5
Ours (Pose only, no finetuning)	X	-	-	72.2
Ours (Pose only)	X	-	-	74.6
Ours (RGB only)	○	X	-	75.3
Ours (Pose + RGB)	X	X	-	90.0

Table 3.3 – **Results on MSR Daily Action dataset.** Accuracies in %.

order, which confirms the interest of a meaningful hierarchical representation. As anticipated, keeping the redundant double joint entries in the tensors gives an advantage, although it increases the amount of trainable parameters.

The effect of the attention mechanism — The attention mechanism on RGB data has a significant impact in term of performance as shown in Table 3.5. We compare it to baseline summing (B) or concatenating (C) features. In these cases,

Methods	CS	CV	Avg
Random joint order	75.5	83.2	79.4
Topological order w/o double entries	76.2	83.9	80.0
Topological order	77.1	84.5	80.8

Table 3.4 – **Effect of joint ordering.** Results on NTU using pose only, accuracies in %.

Methods	*P*	RGB	Attention			CS	CV	Avg
			S	*T*	*P*			
A *P* only	X	-	-	-	-	77.1	84.5	80.8
B RGB only, no attention (sum of features)	-	X	-	-	-	61.5	65.9	63.7
C RGB only, no attention (concat of features)	-	X	-	-	-	63.2	67.2	65,2
E RGB only + spatial attention	○	X	X	-	X	67.4	71.2	69.3
G RGB only + spatio-temporal attention	○	X	X	X	X	75.6	80.5	78.1
H Multi-modal, no attention (A+B)	X	X	-	-	-	83.0	88.5	85.3
I Multi-modal, spatial attention (A+E)	X	X	X	-	X	84.1	90.0	87.1
K Multi-modal, spatio-temporal attention (A+G)	X	X	X	X	X	84.8	90.6	87.7

Table 3.5 – **Affect of attention.** Results on NTU, ○ means that pose is only used for the attention mechanism, S, T, and P means respectively *Spatial*, *Temporal* and *Pose*.

hyper-parametres where optimized for these meta-architectures. The performance margin is particularly high in the case of the single stream RGB model (methods E and G). In the case of the multi-modal (two-stream) models, the advantage of attention is still high but not as high as for RGB alone. A part of the gain of the attention process seems to be complementary to the information in the pose stream, and it cannot be excluded that in the one stream setting a (small) part of the pose information is translated into direct cues for discrimination through an innovative (but admittedly not originally planned) use of the attention mechanism. However, the gain is still significant, with $\sim$2.5 percentage points compared to the baseline.

Figure 3.5 shows an example of the effect of the spatial attention process: during the activity of *Putting an object into the pocket of somebody*, the attention shifts to the "putting" hand at the point where the object is actually put.

Pose-conditioned attention mechanism — Making the spatial attention model conditional to the pose features $\boldsymbol{s}$ is confirmed to be a key design choice, as can be seen in Table 3.6. In the multi-modal setting, a full point is gained, >12 points in the RGB only case.

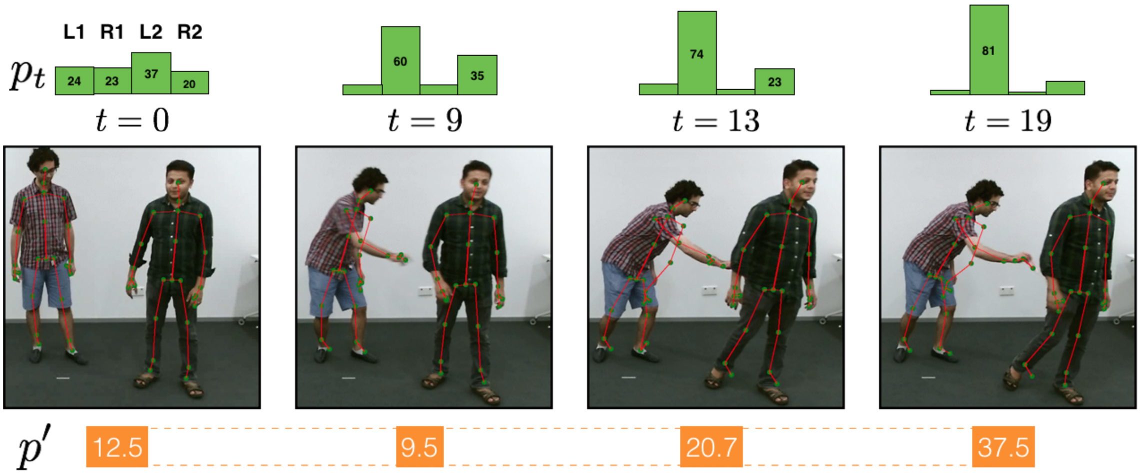


Figure 3.5 – **Spatial attention over time.** Putting an object into the pocket of someone will make the attention shift to this hand.

Methods	Attention Conditional to pose	CS	CV	Avg
RGB only	-	66.5	72.0	69.3
RGB only	X	75.6	80.5	78.1
Multi-modal	-	83.9	90.0	87.0
Multi-modal	X	84.8	90.6	87.7

Table 3.6 – **Conditioning the attention mechanism on pose.** Results on NTU with RGB only, accuracies in %.

Comparison with RGB only methods — There is a clear gap between our approach and standard methods for action recognition on RGB data such as C3D and CNN+LSTM (+21.8 for C3D and +12.1 for CNN+LSTM) as shown in Table 3.1. These methods need to downsample the RGB stream to a lower resolution, which leads to poor performances for fine-grained action recognition. Using some parts of the high resolution RGB stream such as done by our method is important for extracting discriminative features.

3.5 Conclusion

In this chapter, we propose a general method for dealing with pose and RGB video data for human action recognition. A CNN processes input tensors encoding pose data, where features are organized in an anatomically-relevant order. A soft-attention mechanism crops on hand joints and allows the model to collect relevant features on hand shapes and on manipulated objects. Adaptive temporal pooling further increases performance. Our method shows SOTA results on the NTU RGB+D dataset and competitive performance by performance transfer learning on SBU Interaction dataset and MSR Daily Activity.

In the next chapter, we focus on finding an alternative method able to work without having access to articulated pose data as input to the system. We also develop an attention mechanism that can attend to all parts of the video data in an unconstrained manner, in particular, compared to this chapter, without limitation to attend to joint positions only.

Chapter 4

GLIMPSE CLOUDS: HUMAN ACTIVITY RECOGNITION FROM UNSTRUCTURED FEATURE POINTS

Chapter abstract

Articulated human pose is not always available and moreover it has not been demonstrated to be the best representation for estimating human actions.

In this chapter, we propose a method for human activity recognition from RGB data only. Moreover our method does not explicitly compute the pose information internally. Instead, a visual attention module learns to predict glimpse sequences in each frame. These glimpses correspond to interest points in the scene which are relevant to the classified activities. No spatial coherence is forced on the glimpse locations, which allows the module to explore different points at each frame and better optimize the process of scrutinizing visual information. Tracking and sequentially integrating this kind of unstructured data is a challenge, which we address by separating the set of glimpses from a set of recurrent tracking/recognition workers. These workers receive the glimpses, jointly performing subsequent motion tracking and prediction of the activity itself. The glimpses are soft-assigned to the workers, optimizing coherence of the assignments in space, time and feature space using an external memory module. No hard decisions are taken, i.e. each glimpse point is assigned to all existing workers, albeit with different importance.

We improve the results shown in Chapter 3 without the use of articulated human pose on the the NTU RGB+D dataset and also show state-of-the-art results on the Northwestern UCLA dataset.

The work in this chapter has led to the publication of a conference paper:

- Fabien Baradel, Christian Wolf, Julien Mille, and Graham Taylor (2018c). "Glimpse Clouds: Human Activity Recognition from Unstructured Feature Points". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Contents

4.1 Introduction

In Chapter 3, we saw that articulated human pose data obtained by RGB-D cameras is a useful information for recognizing fine-grained human actions. However depth data is not always available for example on resource-constrained robots or low-resource embedded devices such as smartphone. Moreover the question whether articulated pose is the optimal intermediate representation for activity recognition is unclear. For solving this issues we aim at providing a method for fine-grained human action recognition relying on RGB images only. Compared to common unconstrained approaches which rely on the modeling of the global context, as shown in Section 2.2.6, we explore a strategy which consists in learning a local representation of the video through a visual attention process.

We conjecture that the replacement of articulated pose should keep one important property, which is its collection of local entities, which can be tracked over time and whose motion is relevant to the activity at hand. Instead of fixing the semantic meaning of these entities to the definition of a subset of the joints in the human body, we learn it discriminatively. In our strategy, the attention process is completely free to attend to arbitrary locations at each time instant. In particular, we do not impose any constraints on spatio-temporal coherence of glimpse locations, which allows the model to vary its focus within and across frames. Certain similarities can be made to human gaze patterns which saccade to different points in a scene.

Activities are highly correlated with motion, and therefore tracking the motion of specific points of visual interest is essential, yielding a distributed representation of the collection of glimpses. Appearance and motion features need to be collected

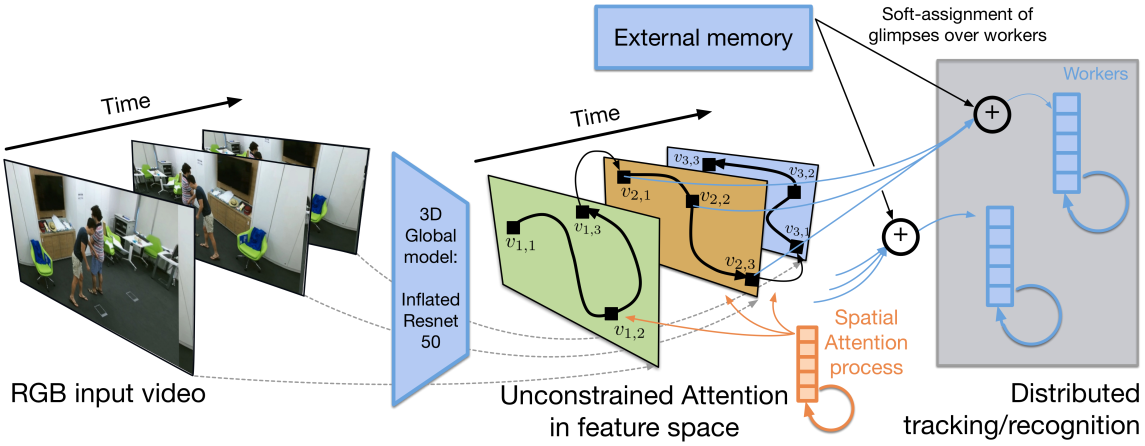

External memory
Soft-assignment of glimpses over workers
Workers
Time
Time
3D Global model: Inflated Resnet 50
$v_{1,1}$
$v_{1,2}$
$v_{1,3}$
$v_{2,1}$
$v_{2,2}$
$v_{2,3}$
$v_{3,1}$
$v_{3,2}$
$v_{3,3}$
Spatial Attention process
RGB input video
Unconstrained Attention in feature space
Distributed tracking/recognition

Figure 4.1 – **Overview of the proposed model.** We recognize human activities from unstructured collections of spatio-temporal glimpses with distributed recurrent tracking/recognition and soft-assignment among glimpse points and trackers.

over time from local points and integrated into a sequential decision model. However, tracking a set of glimpse points, whose location is not spatio-temporally smooth and whose semantic meaning can change from frame to frame, is a challenge. Our objective is to match new glimpses with past ones of the same (or a nearby) location in the scene. Due to the unconstrained nature of the attention mechanism, we do not know when a point in the scene has been last scrutinized, or if it has been attended to in the past.

We solve this issue by separating the problem into two distinct parts:

- Selecting a distributed and local representation of G glimpse points through a sequential recurrent attention model
- Tracking the set of glimpses by a set of C recurrent workers which sequentially integrate features, and participate in the final recognition of the activity (Figure 4.1)

In general, G can be different from C, and the assignment between glimpses and workers is *soft*. Each worker is potentially assigned to all glimpses, albeit to a varying degree. This assignment attention distribution is calculated with external memory based on regularities in space, time and feature space.

We summarize the main contributions of this chapter as follows:

- We present a method for human activity recognition which does not require articulated pose during testing and which models activities using two attentional processes; one extracting a set of glimpses per frame in Section 4.3.1 and one reasoning about entities over time in Section 4.3.2.

- This unstructured "cloud" of glimpses produced by the attention process is tracked over time using a set of trackers/recognizers, which are soft-assigned using external memory. Each tracker can potentially track multiple glimpses.
- Articulated pose is used during *training* time as an additional target, encouraging the attention process to focus on human structures.
- All attentional mechanisms are executed in feature space which is calculated jointly with a global model processing the full input image.
- In Section 4.4 we evaluate our method on the NTU RGB-D dataset, the largest available human activity dataset, where we outperform the State Of The Art (SOTA) by a large margin at the time of publication of this work.

4.2 Related work

Recurrent architectures for action recognition — Recurrent neural networks (and their variants) are employed in much contemporary work on activity recognition, and a recent trend is to make recurrent models local. Part-aware Long-Short Term Memory (LSTM)s (Shahroudy et al. 2016a) separate the memory cell of anLSTM network (Hochreiter et al. 1997) into part-based sub-cells and let the network learn long-term representations individually for each part, fusing the parts for output. Similarly, (Du et al. 2015b) use bi-directional LSTM layers that fit an anatomical hierarchy. Skeletons are split into anatomically-relevant parts (legs, arms, torso, etc.) and let subnetworks specialize on them. Lattice LSTMs partition the latent space over a grid that is aligned with the spatial input space (Sun et al. 2017).

On the other hand, our method soft-assigns parts of the scene over multiple recurrent workers, where each worker can potentially integrate all points of the scene.

Tracking and distributed recognition — Structural Recurrent Neural Network (RNN)s (Jain et al. 2016) bear a certain resemblance to our work. They handle the temporal evolution of tracked objects in videos with a set of RNNs, each of which corresponding to cliques in a graph that models the spatio-temporal relationships between these objects. However, this graph is hand-crafted manually for each application, and object tracking is performed using external trackers, which are not integrated into the neural model.

Our model does not rely on external trackers and does not require the manual creation of a graph, as the assignments between objects (glimpses) and trackers are learned automatically.

4.3 Model

We first introduce the following notation. We want to map our input video sequence $\boldsymbol{X} \in \mathbb{R}^{T \times H \times W \times 3}$ to a corresponding activity label y where H, W, T denote, respectively, the height, the width and the number of time steps. The sequence $\boldsymbol{X}$ is a set of RGB input images $\boldsymbol{X}_t \in \mathbb{R}^{H \times W \times 3}$ with $t = 1...T$. We do not assume any other kind of prior information on the input data. We do not use any external information during testing such as pose data nor depth nor motion. However, if pose data is available during *training time*, our method is able to integrate it in the form of additional inputs, which increases the performance of the system, as shown in Section 4.3.3.

4.3.1 Dynamic sequential attention

Most of the RGB-only SOTA methods, which do not use pose data, extract features at a frame level by feeding the entire video frame to a pre-trained deep network. This leads to global features, which do not capture local information that would be relevant to the activities at hand. Reasoning at a local level has, up till now, been achieved using pose features, or attention processes which were limited to attention maps (e.g. (Sharma et al. 2016; Li et al. 2017)). Here, we propose an alternative approach, where an attention process runs statically over each time instant ***and*** over time, creating sequences of sets of glimpse points, from which features are extracted.

Our model processes videos using several key components, also illustrated in Figure 4.1:

- A ***recurrent spatial attention model*** that extracts features from different local glimpses following an attention path in each video frame predicted by the same model

- ***Recurrent soft-tracking workers*** which process these spatial features sequentially. The input data being unstructured, the spatial glimpses are soft-assigned to the workers, such that no hard decisions are taken at any point.

- To this end, an ***external memory module*** keeps track of the glimpses seen in the past, their features, as well as of past soft-assignments, and produces new soft-assignments optimizing spatio-temporal consistency.

Our approach is fully-differentiable, such that the full model is trained end-to-end.

Joint global/local feature space

We recognize activities based on global and local features jointly. In order to speed up calculations and to avoid extracting redundant calculations, we use a single feature space computed by a global model. In particular, we map an input sequence $\boldsymbol{X}$ to a spatio-temporal feature map $\boldsymbol{Z} \in \mathbb{R}^{T \times H' \times W' \times C'}$ using a deep neural network $f(\cdot)$ with 3D convolutions. Pooling is performed on the spatial dimensions but, not in time. This allows retention of the original temporal scale of the video, and thus access to features in each frame. It should, however, be noted, that due to the 3D convolutions used, the temporal receptive field of a single "temporal" slice of the feature map is greater than a single frame. This is intended, as it allows the attention process to utilize motion. In an abuse of terminology, we will still use the term *frame* to specify the slice $\boldsymbol{Z}_t$ of a feature map with a temporal length of 1. More information on the architecture of $f(\cdot)$ is given in Section 4.3.4.

A recurrent model of spatial attention

Inspired by human behavior when scrutinizing a scene, we extract a fixed number of features from a series of G glimpses within each frame. The process of moving from one glimpse to another is achieved with a recurrent model. Glimpses are indexed by index g=$1 \ldots G$, and each glimpse $\boldsymbol{Z}_{t,g}$ corresponds to a sub-region of $\boldsymbol{Z}_t$ using coordinates and scale $\boldsymbol{l}_{t,g} = \left[x_g, y_g, s_g^x, s_g^y\right]_t^\top$ output by a differentiable glimpse function, which will be defined in Figure 4.3.1. Features are extracted from the glimpse region $\boldsymbol{Z}_{t,g}$ using Global Average Pooling (GAP), resulting in a 1D feature vector $\boldsymbol{z}_{t,g}$:

$$\boldsymbol{z}_{t,g} = \Gamma(\boldsymbol{Z}_{t,g}) = \frac{1}{H'W'} \sum_m \sum_n \boldsymbol{Z}_{t,g}(m,n) \tag{4.1}$$

where $W' \times H'$ is the size of the glimpse region. The glimpse locations and scales $\boldsymbol{l}_g$ for g=$1 \ldots G$ are predicted by a recurrent network, which runs over glimpses. As illustrated in Figure 4.2, the model predicts a fixed-length sequence of glimpse points for each frame. It runs over the video, i.e. it is not restarted/reinitialized after each frame. The hidden state thus carries information across frames and creates a globally coherent scrutinization process over the video. In Equation 4.2 and Equation 4.3 we index glimpses with a linear index g. The recurrent model is given as follows (we use Gated Recurrent Unit (GRU)s (Cho et al. 2014), for notational simplicity we omit gates and biases in the rest of the equations):

$$\boldsymbol{h}_g = \Omega(\boldsymbol{h}_{g-1}, \left[\boldsymbol{z}_{g-1}, \boldsymbol{r}_t\right] | \theta) \tag{4.2}$$

$$\boldsymbol{l}_g = W_l^\top \left[\boldsymbol{h}_g, \boldsymbol{c}_t\right] \tag{4.3}$$

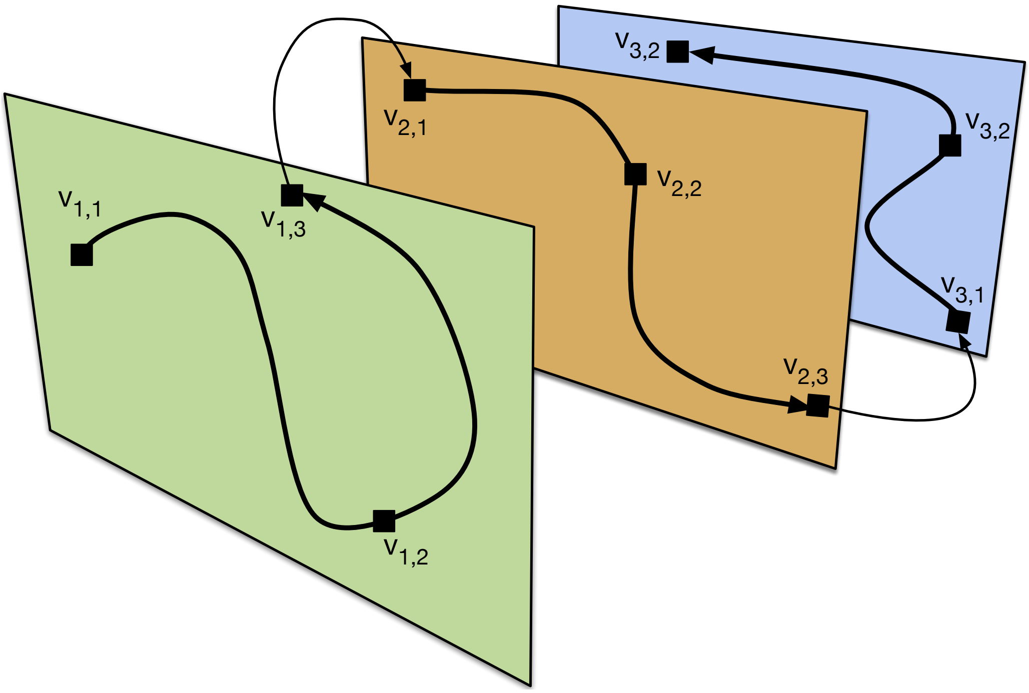


Figure 4.2 – **Dynamic attention process.** A dynamic attention process produces several glimpses for each frame and also runs over frames. The process is free to explore different scene points in different frames.

where $\boldsymbol{h}$ denotes the hidden state of the RNN running over glimpses g, $\boldsymbol{c}_t$ is a frame context vector for making the process aware of frame transitions (described in Equation 4.3.2) and $\boldsymbol{r}_t$ carries information about the high level classification task. In essence, $\boldsymbol{r}_t$ corresponds to the global hidden state of the recurrent workers performing the actual recognition, as described in Section 4.3.2, Equation 4.7.

Differentiable glimpse module

In order to create a model which can be trained end-to-end, we use a simple version of Spatial Transformer Network (STN) (Jaderberg et al. 2015) to perform a differentiable crop operation on each feature map. Given an input feature map $\boldsymbol{Z}_t \in \mathbb{R}^{H \times W \times C}$ and the glimpse parameters

$$\boldsymbol{l}_g = \left[x_g, y_g, s_g^x, s_g^y \right]$$

where (x_g, y_g) is the central focus point and (s_g^x, s_g^y) corresponds to the scale, we output a feature map $\boldsymbol{Z}_{t,g} \in \mathbb{R}^{H' \times W' \times C}$. Note that the output size can differ from the input size.

We constrain the STN to implement a simple 2D affine transformation $A_{\boldsymbol{l}_g}$ which allows cropping, translation and isotropic scaling on a regular grid point x_i^t, y_i^t according to the given glimpse parameters $\boldsymbol{l}_g$:

$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = A_{\boldsymbol{l}_g} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} s_g^x & 0 & x_g \\ 0 & s_g^y & y_g \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} \tag{4.4}$$

where x_i^t, y_i^t are the target coordinates of the regular grid in the output feature map $\boldsymbol{Z}_{t,g}$ and x_i^s, y_i^s are the source coordinates in the input feature map that define the sample points.

We must define a sampler which takes the set of sampling points (x_i^s, y_i^s), along with the input feature map $\boldsymbol{Z}_t$ and produces the sampled output feature map $\boldsymbol{Z}_{t,g}$. We employ bilinear interpolation which implements the following mapping:

$$\boldsymbol{Z}_{t,g}(x_i^s, y_i^s) = \\ \sum_n^{H'} \sum_m^{W'} \boldsymbol{Z}_t(m,n) \max(0, 1-|x_i^s-n|) \max(0, 1-|y_i^s-m|).$$

The STN is differentiable, which allows us to train the parameters W_l for the prediction of focus point parameters $\boldsymbol{l}_g$ together with the rest of the network using gradient descent.

4.3.2 Distributed Reasoning on Unstructured Glimpse Clouds

The attended points (glimpses) predicted in each frame $\boldsymbol{Z}_t$ have a semantic meaning in the input video (e.g. a patch around the hands or shoulders; an object held or pointed at by a person etc.). The goal is to reason about their positions, motion, changes in appearance, relationships or other properties. This is made difficult by the sequential attention process described in Section 4.3.1, which can provide very different glimpse sequences for each frame, since we avoid any direct supervision. This is intentional, in order to give the spatial attention process complete freedom. In particular, it can choose to jump to different glimpse points at each frame, and/or decide to revisit certain glimpses attended to in the past. Since frame features $\boldsymbol{Z}_t$ also encode motion due to the 3D convolutions in $f(\cdot)$, the attention process can learn to revisit attended points, compensating for their motion. In Section 4.4 we describe experiments performed which justify this kind of attention process compared to an alternate choice of spatio-temporally coherent attention.

As a consequence, extracting motion cues from semantic points in the scene requires associating glimpse points from different frames over time. Due to the freedom of the attention process and fixed number of glimpses, subsequent glimpses of the same point in the scene are generally not in subsequent frames, which excludes conventional tracking mechanisms known from the Computer Vision (CV) literature. Instead, we avoid hard tracking and hard assignments between glimpse points in a temporal manner. We propose a soft associative model for automatically associating similar spatial features over time.

Distributed soft-tracking workers

As given in Equation 4.1, we denote by $\boldsymbol{z}_{t,g}$ the features extracted from the g^{th} glimpse in feature map $\boldsymbol{Z}_t$ for $g = 1...G$ and $t = 1...T$. We are interested in a joint encoding of spatial dimensions and feature dimensions and employ *"what"* and *"where"* features $\boldsymbol{v}_{t,g}$ introduced in (Larochelle et al. 2010) defined by:

$$\boldsymbol{v}_{t,g} = \boldsymbol{z}_{t,g} \otimes \Lambda(\boldsymbol{l}_{t,g}|\theta_\Lambda) \tag{4.5}$$

where $\otimes$ is the Hadamard product and $\Lambda(\boldsymbol{l}_{t,g}|\theta_\Lambda)$ is a network which provides an embedding of the spatial patch coordinates into a space which is of the same dimensionality as the features $\boldsymbol{z}_{t,g}$. The vector $\boldsymbol{v}_{t,g}$ contains joint cues about motion and appearance, but also the spatial localization of those features.

Evolution over time of this information is modeled with a number (C) of so-called *soft-tracking workers* Ψ_c for $c = 1...C$. Each worker corresponds to a recurrent model capable of tracking entities over time. We *never* hard assign glimpses to workers. Inputs to each individual worker correspond to weighted contributions from all of the G glimpses. In general, the number of glimpse points G can be different from the number of workers C. At each time instant, focal points are thus soft-assigned to the workers on the fly but changing the weights of the contributions, which will be described further below.

A worker Ψ_c is a recurrent network following the usual update equations based on the past state $\boldsymbol{r}_{t-1,c}$ and its input $\tilde{\boldsymbol{v}}_{t,c}$:

$$\boldsymbol{r}_{t,c} = \Psi_c(\boldsymbol{r}_{t-1,c}, \tilde{\boldsymbol{v}}_{t,c}|\theta_{\Psi_c}) \tag{4.6}$$

$$\boldsymbol{r}_t = \sum_c \boldsymbol{r}_{t,c} \tag{4.7}$$

where Ψ_c is a GRU and $\boldsymbol{r}_t$ is carrying global information about the current state (needed as input of the recurrent model of spatial attention). The input $\tilde{\boldsymbol{v}}_{t,c}$ to each worker Ψ_c is a linear combination of the different glimpses $\{\boldsymbol{v}_{t,g}\}, g = 1 \dots G$ weighted by a soft attention distribution $\boldsymbol{p}_{t,c} = \{p_{t,g,c}\},\ g = 1 \dots G$:

$$\tilde{\boldsymbol{v}}_{t,c} = \boldsymbol{V}_t \boldsymbol{p}_{t,c} \tag{4.8}$$

where $\boldsymbol{V}_t$ is a matrix whose rows are the different glimpse features $\boldsymbol{v}_{t,g}$. Workers are independent from each other in the sense that they do not share parameters θ_{Ψ_c}. This can potentially lead to specialization of the workers on types of tracked and integrated scene entities.

Soft-assignment using External Memory

The role of the attention distribution $\boldsymbol{p}_{t,c}$ is to give higher weights to glimpses which have been soft-assigned to this worker in the past. Thus workers extract different kinds of features from each other. To do so, we employ an external

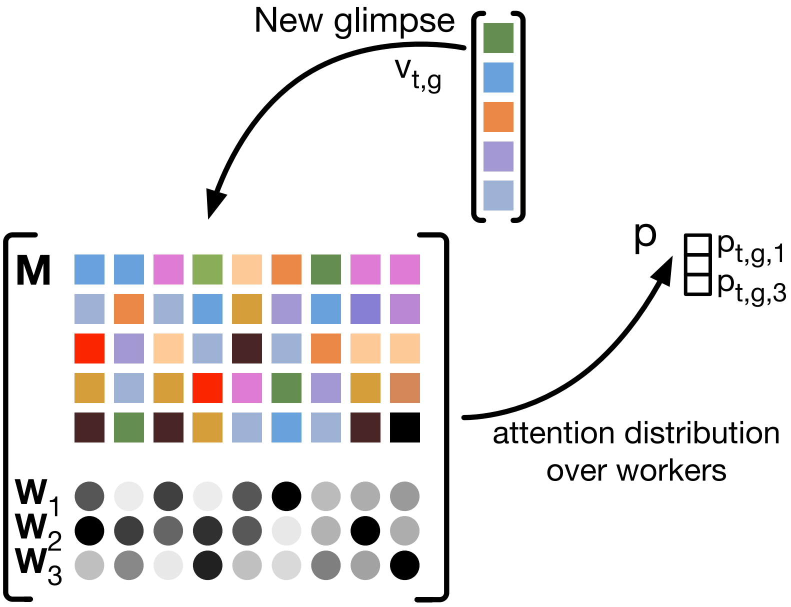


Figure 4.3 – **Memory network.** An external memory module determines an attention distribution over workers (a soft assignment) for each new glimpse $\boldsymbol{v}_{t,g}$ based on similarities with past glimpses M and their past attention probabilities w. Shown for a single glimpse and 3 workers.

memory bank denoted $\boldsymbol{M} = \{\boldsymbol{m}_k\}$ which is common to all workers. In particular, $\boldsymbol{M}$ is a fixed-length array of K entries $\boldsymbol{m}_k$ each capable of storing a feature vector $\boldsymbol{v}_{t,g}$. Even if the external memory is common to each worker, they have their own ability to extract information from it. Each worker Ψ_c has its own weight bank denoted $\boldsymbol{W}_c = \{w_{c,k}\}$. The scalar $w_{c,k}$ holds the importance of the entry $\boldsymbol{m}_{c,k}$ for worker Ψ_c . Hence the overall external memory is defined by the set $\{\boldsymbol{M}, \boldsymbol{W}_1, \ldots \boldsymbol{W}_c\}$.

Attention from memory reads — The attention distribution $\boldsymbol{p}_{t,c}$ is a distribution over glimpses g, i.e.

$$\boldsymbol{p}_{t,c} = \{p_{t,c,g}\}, 0 \leq p_{t,c,g} \leq 1$$

and

$$\sum_g p_{t,c,g} {=} 1.$$

We want the glimpses to get distributed appropriately across the workers, and encourage worker specialization. In particular, at each timestep we want to assign a glimpse high importance to a worker if this worker has been soft-assigned similar glimpses in the past with high importance. To this end, we define a fully trainable distance function $\phi(.,.)$ which is implemented as a quadratic form:

$$\phi(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(\boldsymbol{x} - \boldsymbol{y})^\top \boldsymbol{D} (\boldsymbol{x} - \boldsymbol{y})} \tag{4.9}$$

where $\boldsymbol{D}$ is a learned weight matrix. Within each batch we normalize $\phi(\cdot, \cdot)$ by min-max normalization to scale it to lie between 0 and 1.

A glimpse g is soft-assigned to a given worker c with a higher weight $p_{t,c,g}$ if $\boldsymbol{v}_{t,g}$ is similar to vectors $\boldsymbol{m}_k$ from the memory bank $\boldsymbol{M}$ which had a high importance for the worker in the past Ψ_c :

$$p_{t,c,g} = \sigma_\alpha \left(\sum_k e^{-t^{m_k}} \times w_{c,k} \left[1 - \phi(\boldsymbol{v}_{t,g}, \boldsymbol{m}_k) \right] \right) \tag{4.10}$$

where σ is the softmax function over the G glimpses and $e^{-t^{m_k}}$ is an exponential rate over time to give higher importance to recent feature vectors compared to those in the distant past. t^{m_k} is the corresponding timestep of the memory bank m_k. In practice we add a temperature term α to the softmax function σ. When $\alpha \rightarrow 0$ the output vector is sparser. The negative factor multiplied with ϕ is justified by the fact that ϕ is initially pre-trained as a Mahalanobis distance by setting $\boldsymbol{D}$ to the inverse covariance matrix of the glimpse data. The factor therefore transforms the distance into a similarity. After pre-training, $\boldsymbol{D}$ is trained end-to-end.

The attention distribution $\boldsymbol{p}_{t,c}$ is computed for each worker Ψ_c. Thus each glimpse g potentially contributes to each worker Ψ_c through its input vector $\tilde{\boldsymbol{v}}_{t,c}$ (c.f. Equation 4.8), albeit with different weights.

Memory writes — for each frame, the feature representations $\boldsymbol{v}_{t,g}$ are stored in the memory bank $\boldsymbol{M}$. However, the attention distribution $\boldsymbol{p}_{t,c} = \{p_{t,c,g}\}$ is used to weight these entries for each worker Ψ_c. If a glimpse feature $\boldsymbol{v}_{t,g}$ is stored in a slot $\boldsymbol{m}_k$, then its importance weight $w_{c,k}$ for worker Ψ_c is set to $p_{t,c,g}$. The only limitation is the size K of the memory bank. When the memory is full, we delete the oldest memory slot. More flexible storing processes, e.g. trained mappings, are left for future work.

Recognition

Since workers proceed in a independent manner through time, we need an aggregation strategy to perform classification. Each worker Ψ_c has its own hidden state $\{\boldsymbol{r}_{t,c}\}_{t=1...T}$ and is responsible for its own classification through a fully-connected layer. The final classification is done by averaging logits of the workers:

$$\boldsymbol{q}_c = W_c \cdot \boldsymbol{r}_c \tag{4.11}$$

$$\hat{\boldsymbol{y}} = \text{softmax}\left(\sum_c^C \boldsymbol{q}_c\right) \tag{4.12}$$

where $\hat{\boldsymbol{y}}$ is the probability vector of assigning the input video $\boldsymbol{X}$ to each class.

Context vector

In order to make the spatial attention process (Section 4.3.1) aware of frame transitions, we introduce a context vector $\boldsymbol{c}_t$ which contains high level information about humans present in the current frame t. $\boldsymbol{c}_t$ is obtained by GAP over the spatial domain of the penultimate feature maps of a given timestep. We regress the 2D pose coordinates of humans from the context vector $\boldsymbol{c}_t$ using the following mapping:

$$\boldsymbol{y}_t^p = W_p^\top \boldsymbol{c}_t \tag{4.13}$$

Pose $\boldsymbol{y}_t^p$ is linked to ground truth pose (during *training* only) using a supervised term described in Section 4.3.3. This leads to incorporate hierarchical feature learning in a sense that the penultimate feature maps have to detect human joints present in each frame.

4.3.3 Training

We train the model end-to-end with the sum of a collection of loss terms, which are explained below:

$$\mathcal{L} = \mathcal{L}_D(\hat{\boldsymbol{y}}, \boldsymbol{y}) + \mathcal{L}_P(\hat{\boldsymbol{y}}^p, \boldsymbol{y}^p) + \mathcal{L}_G(\boldsymbol{l}, \boldsymbol{y}^p) \tag{4.14}$$

Supervision — $\mathcal{L}_D(\hat{\boldsymbol{y}}, \boldsymbol{y})$ is a supervised loss term (cross-entropy loss on activity labels $\boldsymbol{y}$).

Pose prediction — Articulated pose $\boldsymbol{y}^p$ is available for many datasets. Our goal is to *not* depend on pose during testing; however, its usage during training can provide additional information to the learning process and reduce the tendency of activity recognition methods to memorize individual elements in the data for recognition. We therefore add an additional term $\mathcal{L}_P(\hat{\boldsymbol{y}}^p, \boldsymbol{y}^p)$, which encourages the model to perform pose regression during training only from intermediate feature maps (described in Equation 4.3.2). Pose regression over time leads to a faster convergence of the overall model.

Making glimpses similar to humans — $\mathcal{L}_G(\boldsymbol{l}, \boldsymbol{y}^p)$ is a loss encouraging the glimpse points to be as sparse as possible within a frame, but at the same time, close to humans in the scene. Recall that $\boldsymbol{l}_{t,g} = \left[x_{t,g}, y_{t,g}, s^x_{t,g}, s^y_{t,g}\right]^T$, so $\mathcal{L}_G$ is defined by:

$$\mathcal{L}^t_{G_1}(\boldsymbol{l}, \boldsymbol{y}^p) = \frac{1}{1 + \sum_{g_1}^G \sum_{g_2}^G ||\boldsymbol{l}_{t,g_1}, \boldsymbol{l}_{t,g_2}||} \tag{4.15}$$

$$\mathcal{L}^t_{G_2}(\boldsymbol{l}, \boldsymbol{y}^p) = \sum_g^G \min_j ||\boldsymbol{l}_{t,g}, \boldsymbol{y}^p_j|| \tag{4.16}$$

$$\mathcal{L}_G(\boldsymbol{l}, \boldsymbol{y}^p) = \sum_t^T \left(\mathcal{L}^t_{G_1}(\boldsymbol{l}, \boldsymbol{y}^p) + \mathcal{L}^t_{G_2}(\boldsymbol{l}, \boldsymbol{y}^p)\right) \tag{4.17}$$

where $\boldsymbol{y}^p_j$ denotes the 2D coordinates of joints j, and Euclidean distance on $\boldsymbol{l}_{t,g}$ is computed using the central focus point $(x_{t,g}, y_{t,g})$. $\mathcal{L}_{G_1}$ encourages diversity between glimpses within a frame. $\mathcal{L}_{G_2}$ ensures that all the glimpses are not taken too far away from the subjects.

4.3.4 Pretraining

We designed the 3D Convolutional Neural Network (CNN) $f(\cdot)$ by computing the global feature maps in Section 4.3.1, such that the temporal dimension is maintained (i.e. without any temporal subsampling). We advance from the Resnet-50 network(He et al. 2016) and inflate the 2D spatial convolutional kernels into 3D kernels, artificially creating new a temporal dimension, as described by Carreira et al. (Carreira et al. 2017). This allows us to take advantage of the 2D kernels learned by pre-training on image classification on the Imagenet dataset. The Inflated ResNet $f(\cdot)$ is then trained as a first step by minimizing the loss $\mathcal{L}_D + \mathcal{L}_P$. The supervised loss $\mathcal{L}_D$ on the global model is applied on a path attached to GAP on the last feature maps, followed by a fully-connected layer that is subsequently removed.

The recurrent spatial attention module $\Omega(\cdot)$ is a GRU with a hidden state of size 1024; $\Lambda(\cdot)$ is an Multi-Layer Perceptron (MLP) with a single hidden layer of size 256 and a Rectified Linear Unit (ReLU) activation; the soft-trackers Ψ_c are GRU with a hidden state of size 512. There is no parameter sharing between them.

4.4 Experiments

This section contains a description of the datasets, a summary of the implementation parameters, and the results of several experiments.

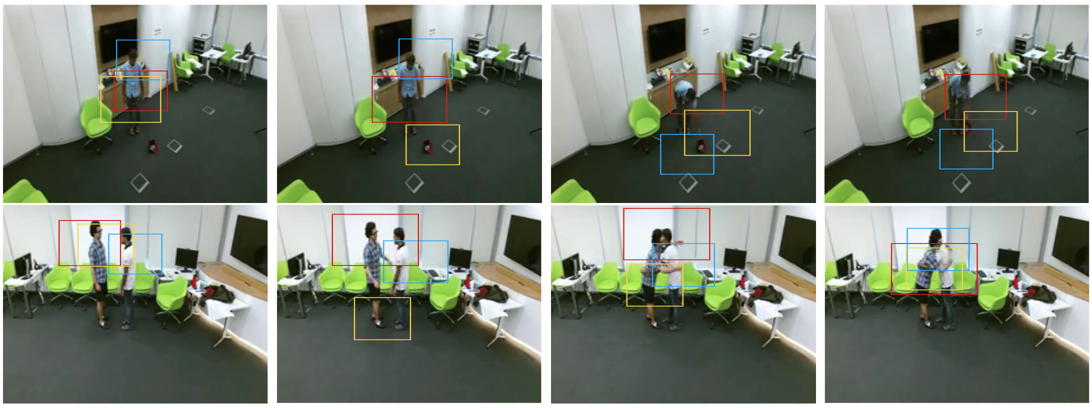


Figure 4.4 – **Illustration of the glimpse distribution.** Shown for several sequences of the NTU dataset. Here we set 3 glimpses per frame (G=3, Red: first, Blue: second, Yellow: third).

4.4.1 Datasets

The proposed method has been evaluated on two human action recognition datasets: NTU RDB+D Dataset (Shahroudy et al. 2016a) and Northwestern-UCLA Multiview Action 3D Dataset (Wang et al. 2014).

NTU RDB+D Dataset (NTU) — NTU was acquired with a Kinect v2 sensor and contains more than 56K videos and 4 million frames with 60 different activities including individual activities, interactions between multiple people, and health-related events. The actions were performed by 40 subjects and recorded from 80 viewpoints. We follow the cross-subject and cross-view split protocol from (Shahroudy et al. 2016a). Due to the large number of videos, this dataset is highly suitable for Deep Learning (DL) modeling.

Northwestern-UCLA Multiview Action 3D Dataset (N-UCLA) — This dataset contains 1494 sequences, covering ten action categories, such as *drop trash* or *sit down* (Wang et al. 2014). Each sequence is captured simultaneously by 3 Kinect v1 cameras. RGB, depth and human pose are available for each video, and each action is performed one to six times by ten different subjects. Most actions involve human-object interaction, making this dataset challenging. We followed the cross-view protocol defined by (Wang et al. 2014), and we trained our method on samples from two camera views, and tested it on samples from the remaining view. This produced three possible cross-view combinations: $V^3_{1,2}$, $V^2_{1,3}$, $V^1_{2,3}$. The combination $V^3_{1,2}$ means that samples from view 1 and 2 are used for training, and samples from view 3 are used for testing.

4.4.2 Implementation details

Similar to (Shahroudy et al. 2016a), we cut videos into sub-sequences of 8 frames and sample sub-sequences. During training, a single sub-sequence is sampled. During testing, 5 sub-sequences and logits are averaged. RGB videos are rescaled to 256 × 256 and random cropping of size 224 × 224 is done during training and testing.

Training is performed using the Adam Optimizer (Kingma et al. 2015) with an initial learning rate of 0.0001. We use minibatches of size 40 on 4 Graphics Processing Unit (GPU)s. Following (Shahroudy et al. 2016a), we sample 5% of the initial training set as a validation set, which is used for hyper-parameter optimization and for early stopping. All hyperparameters have been optimized on the validation sets of the respective datasets. We used the model trained on NTU as a pre-trained model and fine-tuned it on N-UCLA.

4.4.3 Results

Comparison with the state of the art — At the time of publication of this wok, our method outperformed SOTA methods on NTU and N-UCLA by a large margin, and this also includes several methods which use multiple modalities, in addition to RGB, depth and pose. Table 4.1 and Table 4.2 provide detailed results compared to the SOTA on the NTU dataset. Sample visual results can be seen in Figure 4.4.

Ablation study — Table 4.3 shows several experiments to study the effect of our design choices. Classification from the Global Model (GM) alone (Inflated-Resnet-50) is clearly inferior to the distributed recognition strategy using the set of workers (+1.9 points on NTU and +4.4 points on N-UCLA). The bigger gap obtained on N-UCLA can be explained by the larger portion of the frame occupied by people and therefore higher efficiency of a local representation. The additional loss predicting pose during training helps, even though pose is not used during testing. An important question is whether the Glimpse Cloud could be integrated with an easier mechanism than a soft-assignment. We tested a baseline which sums glimpse features for each time step and which integrates them temporally (row #3). This gave only a very small improvement over the global model. Distributed recognition from Glimpse Clouds with soft-assignment clearly outperforms the simpler baselines. Adding the global model does not gain any improvement.

Importance of losses — Table 4.3 also shows the importances of our three loss functions. Cross-entropy only L_D gives 89.1%. Adding pose prediction L_P

Methods	Data	$V_{1,2}^{3}$	$V_{1,3}^{2}$	$V_{2,3}^{1}$	Avg
DVV (Li et al. 2012b)	D	58.5	55.2	39.3	51.0
CVP (Zhang et al. 2013)	D	60.6	55.8	39.5	52.0
AOG (Wang et al. 2014)	D	45.2	-	-	-
HPM+TM (Rahmani et al. 2016)	D	**91.9**	75.2	71.9	79.7
Lie group (Vemulapalli et al. 2014)	P	74.2	-	-	-
HBRNN-L (Du et al. 2015b)	P	78.5	-	-	-
Enhanced viz. (Liu et al. 2017c)	P	86.1	-	-	-
Ensemble TS-LSTM (Lee et al. 2017)	P	89.2	-	-	-
Hankelets (Li et al. 2012a)	V	45.2	-	-	-
nCTE (Gupta et al. 2014)	V	68.6	68.3	52.1	63.0
NKTM (Rahmani et al. 2015)	V	75.8	73.3	59.1	69.4
Global model	V	85.6	84.7	79.2	83.2
Glimpse Clouds	V	90.1	**89.5**	**83.4**	**87.6**

Table 4.1 – **Results on the Northwestern-UCLA Multiview Action 3D dataset.** With Cross-View Setting (accuracy as a percent). V, D, and P mean Visual (RGB), Depth, and Pose, respectively.

we gain 0.6 points and adding pose attraction L_G we gain 0.4 points, which are complementary.

Unstructured vs. coherent attention — We also evaluated the choice of unstructured attention, i.e. the decision to give the attention process complete freedom to attend to a new (and possibly unrelated) set of scene points in each frame. We compared this with an alternative choice, where glimpses are axis-aligned space-time tubes over the whole temporal length of the video. In this baseline, the attention process is not aligned with time. At each iteration, a new tube is attended in the full space-time volume, and no tracking or soft-assignment to worker modules is necessary. As indicated in Table 4.4, this choice is sub-optimal. We conjecture that tubes cannot cope with moving objects and object parts in the video.

Attention vs. saliency vs. random — We evaluated whether a sequential attention process contributes to performance, or whether the gain is solely explained from the sampling of local features in the space-time volume. We compared our choice with two simple baselines: (i) complete random sampling of local features, which leads to a drop of more than 6 points, indicating that the location of the glimpses is clearly important; and (ii) with a saliency model, which predicts glimpse locations in parallel through different outputs of the location network.

Methods	Pose	RGB	CS	CV	Avg
Lie Group (Vemulapalli et al. 2014)	✓	-	50.1	52.8	51.5
Skeleton Quads (Evangelidis et al. 2014)	✓	-	38.6	41.4	40.0
Dynamic Skeletons (Hu et al. 2015)	✓	-	60.2	65.2	62.7
HBRNN (Du et al. 2015b)	✓	-	59.1	64.0	61.6
DeepLSTM(Shahroudy et al. 2016a)	✓	-	60.7	67.3	64.0
Part-awareLSTM(Shahroudy et al. 2016a)	✓	-	62.9	70.3	66.6
ST-LSTM + TrustG. (Liu et al. 2016a)	✓	-	69.2	77.7	73.5
STA-LSTM (Song et al. 2016)	✓	-	73.2	81.2	77.2
Ensemble TS-LSTM (Lee et al. 2017)	✓	-	74.6	81.3	78.0
GCA-LSTM (Liu et al. 2017a)	✓	-	74.4	82.8	78.6
JTM (Wang et al. 2016b)	✓	-	76.3	81.1	78.7
MTLN (Ke et al. 2017)	✓	-	79.6	84.8	82.2
VA-LSTM (Zhang et al. 2017)	✓	-	79.4	87.6	83.5
View-invariant (Liu et al. 2017c)	✓	-	80.0	87.2	83.6
DSSCA - SSLM (Shahroudy et al. 2016b)	✓	✓	74.9	-	-
STA-Hands (Baradel et al. 2017b)	X	X	82.5	88.6	85.6
Hands Attention (Baradel et al. 2017a)	✓	✓	84.8	90.6	87.7
C3D†	-	✓	63.5	70.3	66.9
Resnet50+LSTM†	-	✓	71.3	80.2	75.8
Glimpse Clouds	-	✓	**86.6**	**93.2**	**89.9**

Table 4.2 – **Results on the NTU RGB+D dataset.** With Cross-Subject and Cross-View settings (accuracies in %); († indicates method has been re-implemented).

Methods	Spatial Attention	Soft Workers	L_D	L_P	L_G	CS	CV	Avg
GM	-	-	✓	-	-	84.5	91.5	88.0
GM	-	-	✓	✓	-	85.5	92.1	88.8
GM+$\sum$ Glimpses + GRU	-	-	✓	✓	-	85.8	92.4	89.1
Glimpse Clouds	✓	✓	✓	-	-	85.7	92.5	89.1
Glimpse Clouds	✓	✓	✓	✓	-	86.4	93.0	89.7
Glimpse Clouds	✓	✓	✓	-	✓	86.1	92.9	89.5
Glimpse Clouds	✓	✓	✓	✓	✓	**86.6**	**93.2**	**89.9**
Glimpse Clouds + GM	✓	✓	✓	✓	✓	86.6	93.2	89.9

Table 4.3 – **Ablation study.** Results on NTU. Note: GM means Global model.

This is not a full attention process in that a glimpse prediction does not depend on what the model has seen in the past. This choice is also sub-optimal.

Glimpse	Type of attention	CS	CV	Avg
3D tubes	Attention	85.8	92.7	89.2
Seq. 2D	Random sampling	80.3	87.8	84.0
Seq. 2D	Saliency	86.2	92.9	89.5
Seq. 2D	**Attention**	**86.6**	**93.2**	**89.9**

Table 4.4 – **Different attention and alternative strategies.** Results on the NTU.

Learned weight matrix — Random initialization and fine-tuning of D matrix in Equation 4.9 loses 0.4 points and leads to slower convergence by a factor of 1.5. Fixing D (to inverse covariance) w/o any training loses 0.8 points.

The Joint encoding — "What and where" features are important for correctly weighting their respective contribution. Plainly adding concatenating coordinates and features loses 1.1 points.

Hyper-parameters C, G, T — Number of glimpses and workers: C and G were selected by cross-validation on the validation set by varying them from 1 to 4, giving an optimum of G=C=3 over all 16 combinations. More leads the model to overfit. The size of the memory bank K is set to T where T=8 is the length of the sequence.

Runtime — The model has been trained on a GPU cluster with a single job spread over 4 Titan Xp GPUs. Pre-training the global model on the NTU dataset takes 16h. Training the Glimpse Cloud model end-to-end then takes a further 12h. A single forward pass over the full model takes 97ms on 1 GPU. The method has been implemented in PyTorch.

4.5 Conclusion

In this chapter, we proposed a method for human activity recognition that does not rely on depth images or articulated pose, though it is able to leverage pose information during training. The method achieves SOTA performance on the NTU and N-UCLA datasets even when compared to methods that use pose, depth, or both at test time. An attention process over space and time produces an unstructured *Glimpse Cloud*, which is soft-assigned to a set of tracking/recognition workers. In our experiments, we showed that this distributed recognition outperforms a global convolutional model, as well as local models with simple baselines for the localization of glimpses.

The method proposed in this chapter works well in constrained environment such as demonstrated on the two datasets that we have used. However in real life scenarios, the camera may be moving or the background could vary over time which could potentially affect the performance of our method. In the next chapter, we focus on building a more structured video representation based on object-interactions with the goal to extract the semantic of daily life videos.

Chapter 5

OBJECT LEVEL VISUAL REASONING IN VIDEOS

Chapter abstract

Human activity recognition is typically addressed by detecting key concepts like global and local motion, features related to object classes present in the scene, as well as features related to the global context. The next open challenges in activity recognition require a level of understanding that pushes beyond this and call for models with capabilities for fine distinction and detailed comprehension of interactions between actors and objects in a scene.

We propose a model capable of learning to reason about semantically meaningful spatio-temporal interactions in videos. The key to our approach is a choice of performing this reasoning at the object level through the integration of state-of-the-art object detection networks. This allows the model to learn detailed spatial interactions that exist at a semantic, object-interaction relevant level.

We evaluate our method on three standard datasets (Twenty-BN Something-Something, VLOG and EPIC Kitchens) and achieve state-of-the-art results on all of them. We also show visualizations of the interactions learned by the model, which illustrate object classes and their interactions corresponding to different activity classes.

The work in this chapter has led to the publication of a conference paper:

- Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori (2018a). “Object Level Visual Reasoning in Videos”. In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV).*

Contents

5.1 Introduction

In the previous chapters, we developed end-to-end attention-based mechanisms for human action recognition. We show in Chapter 3 that the articulated human pose is an important information for selecting points of interest in the video stream. Furthermore, we show in Chapter 4 that attention-free mechanisms can be developed without relying on internal pose information. In this chapter, we push the modeling of video content one step further by focusing on human-objects interactions. We focus on a human-centric viewpoint of activity recognition where it is not only the human positions or the presence of certain objects / scenes that dictate the current activity, but the manner, order, and effects of human interactions with these scene elements that are necessary for understanding.

Humans are able to infer what happened in a video given only a few sample frames. This faculty is called *reasoning* and is a key component of human intelligence. As an example we can consider the pair of images in Figure 5.1, which shows a complex situation involving articulated objects (human, carrots and knife), the change of location and composition of objects. For humans it is straightforward to draw a conclusion on what happened (a carrot was chopped by the human). Humans have this extraordinary ability to perform visual reasoning on very complicated tasks while it remains unattainable for contemporary Computer Vision (CV) algorithms (Stabinger et al. 2016; Fleuret et al. 2011).

We describe in Figure 2.2.7 that there are several attempts to equip neural models with reasoning abilities.

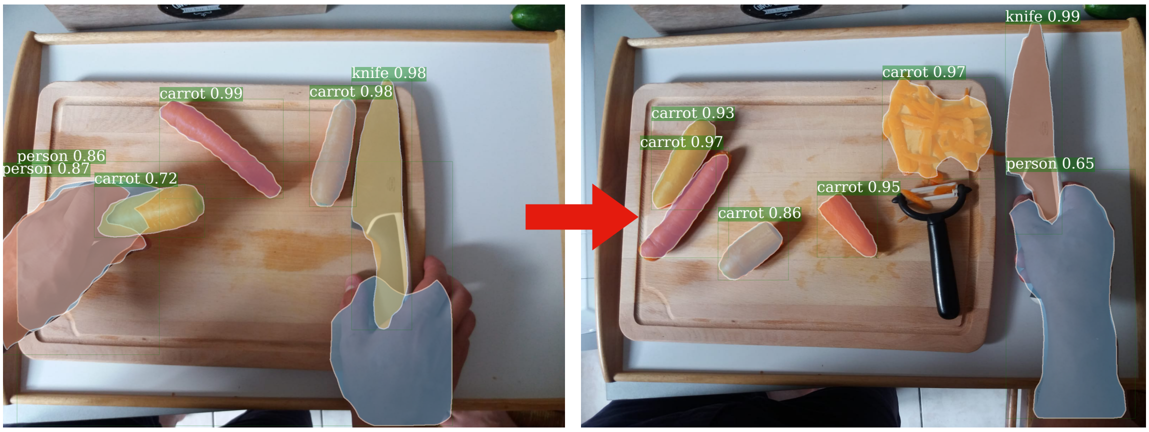


Figure 5.1 – **Reasoning from two consecutive frames.** Humans can understand what happened in a video ("the leftmost carrot was chopped by the person") given only a pair of frames. Along these lines, the goal of this work is to explore the capabilities of higher-level *reasoning* in neural models operating at the semantic level of objects and interactions.

We extend these efforts to *object level reasoning in videos*. Since a video is a temporal sequence, we leverage time as an explicit causal signal to identify causal object relations. Our approach is related to the concept of the *"arrow of the time"* (Pickup et al. 2014) involving the "one-way direction" or "asymmetry" of time. In Figure 5.1 the knife was used before the carrot switched over to the chopped-up state on the right side. For a video classification problem, we want to identify a causal event A happening in a video that affects its label B. But instead of identifying this causal event directly from pixels we want to identify it from an object level perspective.

Following this hypothesis, we propose to make a bridge between object detection and activity recognition. Object detection allows us to extract low-level information from a scene with all the present object instances and their semantic meanings. However, detailed activity understanding requires reasoning over these semantic structures, determining which objects were involved in interactions, of what nature, and what were the results of these. To compound problems, the semantic structure of a scene may change during a video (e.g. a new object can appear, a person may make a move from one point to another one of the scene).

We propose an **Object Relation Network** (ORN), a neural network module for reasoning between detected semantic object instances through space and time. The ORN has potential to address these issues and conduct relational reasoning over object interactions for the purpose of activity recognition. A set of object detection masks ranging over different object categories and temporal occurrences is input to the ORN. The ORN is able to infer pairwise relationships between objects detected at different moments in time.

Code and object masks predictions is publicly available [1].

5.2 Related work

Action Recognition — Pre-deep learning approaches in action recognition focused on handcrafted spatio-temporal features including space-time interest points like SIFT-3D, HOG3D, IDT and aggregated them using bag-of-words techniques. Some hand-crafted representations, like dense trajectories (Wang et al. 2011), still give competitive performance and are frequently combined with deep learning.

In the recent past, work has shifted to deep learning. Early attempts adapt 2D Convolutional Neural Network (CNN)s to videos through temporal pooling and 3D convolutions (Baccouche et al. 2011; Tran et al. 2015). 3D convolutions are now widely adopted for activity recognition with the introduction of feature transfer by inflating pre-trained 2D convolutional kernels from image classification models trained on ImageNet/ILSVRC (Russakovsky et al. 2015) through 3D kernels (Carreira et al. 2017). The downside of 3D kernels is their computational complexity and the large number of learnable parameters, leading to the introduction of 2.5D kernels, i.e. separable filters in the form of a 2D spatial kernel followed by a temporal kernel (Xie et al. 2017). An alternative to temporal convolutions are Recurrent Neural Network (RNN)s in their various gated forms (Gated Recurrent Unit (GRU), Long-Short Term Memory (LSTM)) (Hochreiter et al. 1997; Chung et al. 2014).

Karpathy et al. (2014) presented a wide study on different ways of connecting information in spatial and temporal dimensions through convolutions and pooling. On very general datasets with coarse activity classes they have showed that there was a small margin between classifying individual frames and classifying videos with more sophisticated temporal aggregation.

Simonyan et al. (2014) proposed a widely adopted two-stream architecture for action recognition which extracts two different streams, one processing raw RGB input and one processing pre-computed optical flow images.

In slightly narrower settings, prior information on the video content can allow more fine-grained models. Articulated pose is widely used in cases where humans are guaranteed to be present (Shahroudy et al. 2016a). Pose estimation and activity recognition as a joint (multi-task) problem has recently shown to improve both tasks (Luvizon et al. 2018).

Attention models are a way to structure deep networks in an often generic way. They are able to iteratively focus attention to specific parts in the data without requiring prior knowledge about part or object positions. In activity recognition, they have gained some traction in recent years, either as soft-attention

[1] https://github.com/fabienbaradel/object_level_visual_reasoning

on articulated pose (joints) (Song et al. 2016), on feature map cells (Sharma et al. 2016), on time (Yeung et al. 2015) or on parts in raw RGB input through differentiable crops (Baradel et al. 2018c).

When raw video data is globally fed into deep neural networks, they focus on extracting spatio-temporal features and perform aggregations. It has been shown that these techniques fail on challenging fine-grained datasets, which require learning long temporal dependencies and human-object interactions. A concentrated effort has been made to create large scale datasets to overcome these issues (Goyal et al. 2017; Fouhey et al. 2018; Krishna et al. 2017; Gu et al. 2017).

Relational Reasoning — Relational reasoning is a well studied field for many applications ranging from visual reasoning (Santoro et al. 2017) to reasoning about physical systems (Battaglia et al. 2016). Battaglia et al. (2016) introduce a fully-differentiable network physics engine called Interaction Network (IN). IN learns to predict several physical systems such as gravitational systems, rigid body dynamics, and mass-spring systems. It shows impressive results; however, it learns from a virtual environment, which provides access to virtually unlimited training examples. Following the same perspective, Santoro et al. (2017) introduced Relation Network (RN), a plug-in module for reasoning in deep networks. RN shows human-level performance in Visual Question Answering (VQA) by inferring pairwise "object" relations. However, in contrast to our work, the term "object" in (Santoro et al. 2017) does not refer to semantically meaningful entities, but to discrete cells in feature maps. The number of interactions therefore grows with feature map resolutions, which makes it difficult to scale. Furthermore, a recent study (Kim et al. 2018) has shown that some of these results are subject to dataset bias and do not generalize well to small changes in the settings of the dataset.

In the same line, a recent work (Steenkiste et al. 2018) has shown promising results on discovering objects and their interactions in an unsupervised manner using training examples from virtual environments. In (Veličković et al. 2018), attention and relational modules are combined on a graph structure. From a different perspective, Perez et al. (2017) show that relational reasoning can be learned for visual reasoning in a data driven way without any prior using conditional batch normalization with a feature-wise affine transformation based on conditioning information. In an opposite approach, a strong structural prior is learned in the form of a complex attention mechanism: in (Hudson et al. 2018b), an external memory module combined with attention processes over input images and text questions, performing iterative reasoning for VQA.

While most of the discussed work has been designed for VQA and for predictions on physical systems and environments, extensions have been proposed for video understanding. Reasoning in videos on a mask or segmentation level has been attempted for video prediction (Luc et al. 2017), where the goal was to leverage semantic information to be able predict further into the future. Bolei et al.

(2017) have recently shown state-of-the-art performance on challenging datasets by extending Relation Network to video classification. Their chosen entities are frames, on which they employ RN to reason on a temporal level only through pairwise frame relations. The approach is promising, but restricted to temporal contextual information without an understanding on a local object level, which is provided by our approach.

5.3 Model

Our goal is to extract multiple types of cues from a video sequence: interactions between predicted objects and their semantic classes, as well as local and global motion in the scene. We formulate this objective as a neural architecture with two heads: an *activity head* and an *object head*. Figure 5.2 gives a functional overview of the model. Both heads share common features up to a certain layer shown in red in the figure. The *activity head*, shown in orange in the figure, is a CNN-based architecture employing convolutional layers, including spatio-temporal convolutions, able to extract global motion features. However, it is not able to extract information from an object level perspective. We leverage the *object head* to perform reasoning on the relationships between predicted object instances.

Our main contribution is a new structured module called **Object Relation Network** (ORN), which is able to perform spatio-temporal reasoning between detected object instances in the video. ORN is able to reason by modeling how objects move, appear and disappear and how they interact between two frames.

In this section, we will first describe our main contribution, the ORN network. We then provide details about object instance features, about the activity head, and finally about the final recognition task. In what follows, lowercase letters denote 1D vectors while uppercase letters are used for 2D and 3D matrices or higher order tensors. We assume that the input of our system is a video of T frames denoted by $X_{1:T} = (X_t)_{t=1}^{T}$ where X_t is the RGB image at timestep t. The goal is to learn a mapping from $X_{1:T}$ to activity classes y.

5.3.1 Object Relation Network

ORN (Object Relation Network) is a module for reasoning between semantic objects through space and time. It captures object moves, arrivals and interactions in an efficient manner. We suppose that for each frame t, we have a set of objects k with associated features o_t^k. Objects and features are detected and computed by the object head described in Section 5.3.2.

Reasoning about activities in videos is inherently temporal, as activities follow the *arrow of time* (Pickup et al. 2014), i.e. the causality of the time dimension imposes that past actions have consequences in the future but *not* vice-versa. We

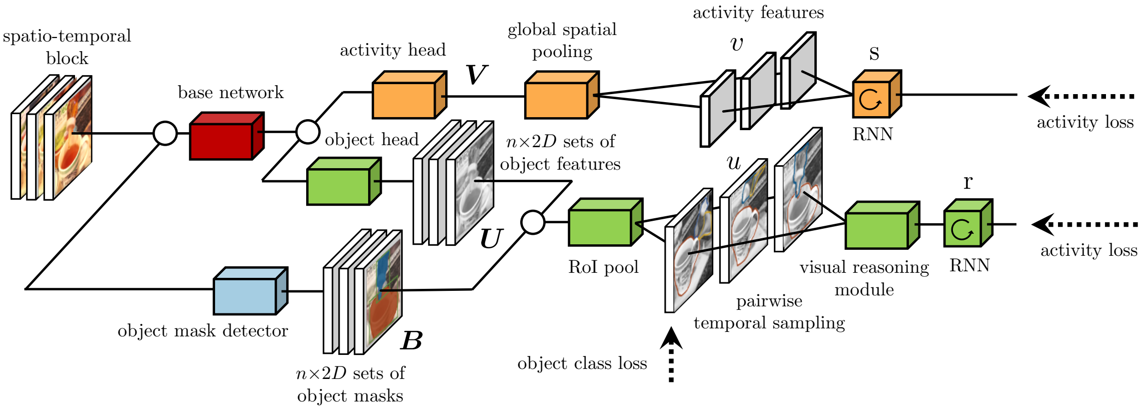


Figure 5.2 – **Functional overview of the model**. A global convolutional model extracts features and splits into two heads trained to predict, respectively activity classes and object classes. The latter are predicted by pooling over object instance masks, which are predicted by an additional convolutional model. The object instances are passed through a visual reasoning module.

handle this by sampling: running a process over time t, and for each instant t, sampling a second frame t' with $t'{<}t$. Our network reasons on objects which interact between pairs of frames and their corresponding sets of objects $O_{t'} = \{o_{t'}^k\}_{k=1}^{K'}$ and $O_t = \{o_t^k\}_{k=1}^{K}$. The goal is to learn a general function defined on the set of all input objects from the combined set of both frames:

$$g_t = g(o_{t'}^1, \ldots, o_{t'}^{K'}, o_t^1, \ldots, o_t^K). \tag{5.1}$$

The objects in this set are unordered, aside for the frame they belong to.

Inspired by relational networks (Santoro et al. 2017), we chose to directly model inter-frame interactions between pairs of objects (j,k) and leave modeling of higher-order interactions to the output space of the mappings h_θ and the global mapping f_ϕ:

$$g_t = \sum_{j,k} h_\theta(o_{t'}^j, o_t^k) \tag{5.2}$$

It is interesting to note that $h_\theta(\cdot)$ could have been evaluated over arbitrary cliques, like singletons and triplets — this has been evaluated in the experimental section. In order to better directly model long-range interactions, we make the global mapping $f_\phi(\cdot,\cdot)$ recurrent, which leads to the following form:

$$r_t = f_\phi(g_t, r_{t-1}) \tag{5.3}$$

where r_t represents the recurrent *object reasoning state* at time t and g_t is the global inter-frame interaction inferred at time t such as described in Equation 5.2. In

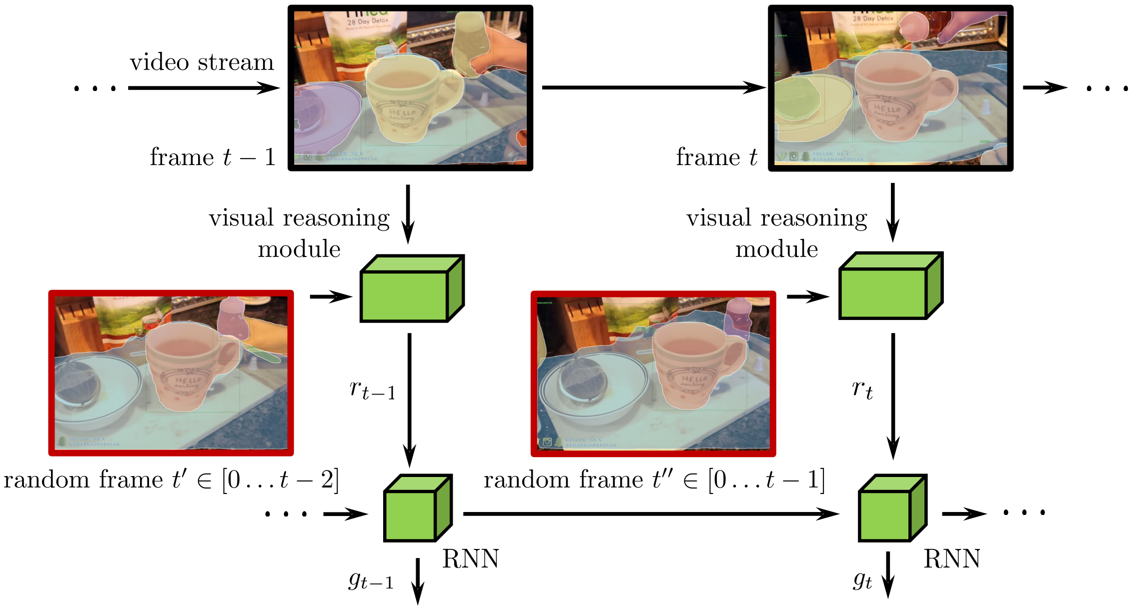


Figure 5.3 – **Object Relation Network.** ORN in the object head operating on detected instances of objects.

practice, this is implemented as a GRU, but for simplicity we omitted the gates in Equation 5.3. The pairwise mappings $h_\theta(\cdot,\cdot)$ are implemented as an Multi-Layer Perceptron (MLP). Figure 5.3 provides a visual explanation of the object head's operating through time.

Our proposed ORN differs from (Santoro et al. 2017) in three main points:

- **Objects have a semantic definition** — we model relationships with respect to semantically meaningful entities (object instances) instead of feature map cells which do not have a semantically meaningful spatial extent. We will show in the experimental section that this is a key difference.

- **Objects are selected from different frames** — we infer object pairwise relations only between objects present in two different sets. This is a key design choice which allows our model to reason about changes in object relationships over time.

- **Long range reasoning** — integration of the object relations over time is recurrent by using a RNN for $f_\phi(\cdot)$. Since reasoning from a full sequence cannot be done by inferring the relations between two frames, $f_\phi(\cdot)$ allows long range reasoning on sequences of variable length.

5.3.2 Object instance features

The object features $O_t = \{o_t^k\}_{k=1}^K$ for each frame t used for the ORN module described above are computed and collected from local regions predicted by a mask predictor. Independently for each frame X_t of the input data block, we predict object instances as binary masks B_t^k and associated object class predictions c_t^k, a distribution over C classes. We use Mask-RCNN (He et al. 2017), which is able to detect objects in a frame using Region Proposal Network (RPN) (Ren et al. 2015) and produces a high quality segmentation mask for each object instance.

The objective is to collect features for each object instance, which jointly describe its appearance, the change in its appearance over time, and its shape, i.e. the shape of the binary mask. In theory, appearance could also be described by pooling the feature representation learned by the mask predictor (Mask R-CNN). However, in practice we choose to pool features from the dedicated *object head* such as shown in Figure 5.2, which also include motion through the spatio-temporal convolutions shared with the activity head:

$$u_t^k = \text{ROI-Pooling}(U_t, B_t^k) \tag{5.4}$$

where U_t is the feature map output by the *object head*, u_t^k is a D-dimensional vector of appearance and appearance change of object k.

Shape information from the binary mask B_t^k is extracted through the following mapping function: $b_t^k = g_\phi(B_t^k)$, where $g_\phi(\cdot)$ is a MLP. Information about object k in image X_t is given by a concatenation of appearance, shape, and object class: $o_t^k = [\ b_t^k \ \ u_t^k \ \ c_t^k\ ]$.

5.3.3 Global Motion and Context

Current approaches in video understanding focus on modeling the video from a high-level perspective. By a stack of spatio-temporal convolution and pooling they focus on learning global scene context information. Effective activity recognition requires integration of both of these sources: global information about the entire video content in addition to relational reasoning for making fine distinctions regarding object interactions and properties.

In our method, local low-level reasoning is provided through object head and the ORN module such as described above in Section 5.3.1. We complement this representation by high-level context information described by V_t which are feature outputs from the activity head (orange block in Figure 5.2).

We use spatial global average pooling over V_t to output T D-dimensional feature vectors denoted by v_t, where v_t corresponds to the context information of the video at timestep t.

We model the dynamics of the context information through time by employing a RNN $f_\gamma(\cdot)$ given by:

$$s_t = f_\gamma(v_t, s_{t-1}) \tag{5.5}$$

where s is the hidden state of $f_\gamma(\cdot)$ and gives cues about the evolution of the context though time.

5.3.4 Recognition

Given an input video sequence $X_{1:T}$, the two different streams corresponding to the activity head and the object head result in the two representations h and r, respectively where $h = \sum_t \mathrm{h}_t$ and $r = \sum_t \mathrm{r}_t$. Each representation is the hidden state of the respective GRU, which were described in the preceding subsections. Recall that h provides the global motion context while r provides the object reasoning state output by the ORN module. We perform independent linear classification for each representation:

$$y^1 = W\,h \tag{5.6}$$

$$y^2 = Z\,r \tag{5.7}$$

where y^1, y^2 correspond to the logits from the *activity head* and the *object head*, respectively, and W and Z are trainable weights (including biases). The final prediction is done by averaging logits y^1 and y^2 followed by softmax activation.

5.3.5 Architectures

The input RGB images X_t are of size $\mathbf{R}^{3\times W\times H}$ where W and H correspond to the width and height and are of size 224 each. The object and activity heads (orange and green in Figure 5.2) are a joint convolutional neural network with Resnet50 architecture pre-trained on ImageNet/ILSVRC (Russakovsky et al. 2015), with Conv1 and Conv5 blocks being inflated to 2.5D convolutions (Xie et al. 2017) (3D convolutions with a separable temporal dimension). This choice has been optimized on the validation set, as explained in Section 5.4 and shown in Table 5.5.

The last *conv5* layers have been split into two different heads (activity head and object head). The intermediate feature representations U_t and V_t are of dimensions $2048\times T\times 7\times 7$ and $2048\times T\times 14\times 14$, respectively. We provide a higher spatial resolution for the feature maps U_t of the object head to get more precise local descriptors. This can be done by changing the stride of the initial *conv5* layers from 2 to 1. Temporal convolutions have been configured to keep the same time temporal dimension through the network.

Global spatial pooling of activity features results in a 2048 dimensional feature vector fed into a GRU with 512 dimensional hidden state s_t. Region of Interest (RoI)-Pooling of object features results in 2048 dimensional feature vectors u_t^k. The encoder of the binary mask is a MLP with one hidden layer of size 100 and outputs a mask embedding b_t^k of dimension 100. The number of object classes is 80, which leads in total to a 2229 dimensional object feature vector o_t^k.

The non-linearity $h_\theta(\cdot)$ is implemented as an MLP with 2 hidden layers each with 512 units and produces an 512 dimensional output space. $f_\phi(\cdot)$ is implemented as a GRU with a 256 dimension hidden state r_t. We use Rectified Linear Unit (ReLU) as the activation function after each layer for each network.

5.3.6 Training

We train the model with two different losses:

$$\mathcal{L} = \mathcal{L}_1\Big(\frac{\hat{y}^1 + \hat{y}^2}{2}, y\Big) + \sum_t \sum_k \mathcal{L}_2(\hat{c}_t^k, c_t^k). \tag{5.8}$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ are the cross-entropy loss. The first term corresponds to supervised activity class losses comparing two different activity class predictions to the class ground truth: $\hat{y}^1$ is the prediction of the activity head, whereas $\hat{y}^2$ is the prediction of the object head, as given by Equation 5.6 and Equation 5.7, respectively.

The second term is a loss which pushes the features U of the object towards representations of the semantic object classes. The goal is to obtain features related to, both, motion (through the layers shared with the activity head), as well as as object classes. As ground-truth object classes are not available, we define the loss as the cross-entropy between the class label c_t^k predicted by the mask predictor and a dedicated linear class prediction $\hat{c}_t^k$ based on features u_t^k, which, as we recall, are **ROI!** (**ROI!**)-pooled from U:

$$c_t^k = R\ u_t^k \tag{5.9}$$

where R trainable parameters (biases integrated) learned end-to-end together with the other parameters of the model.

We found that first training the object head only and then the full network was performing better. A ResNet50 network pretrained on ImageNet is modified by inflating some of its filters to 2.5 convolutions (3D convolutions with the time dimension separated), as described in Section 5.3.5; then by fine-tuning.

We train the model using the Adam optimizer (Kingma et al. 2015) with an initial learning rate of 10^{-4} on 30 epochs and use early-stopping criterion on the validation set for hyper-parameter optimization. Training takes ∼50 minutes per epoch on 4 Titan XP Graphics Processing Unit (GPU)s with clips of 8 frames.

5.4 Experiments

5.4.1 Comparison with existing methods

We evaluated the method on three standard datasets, which represent difficult fine-grained activity recognition tasks: the Something-Something dataset, the VLOG dataset and the recently released EPIC Kitchens dataset.

Something-Something (SS) — is a recent video classification dataset with 108,000 example videos and 157 classes (Goyal et al. 2017). It shows humans performing different actions with different objects, actions and objects being combined in different ways. Solving SS requires common sense reasoning and the State Of The Art (SOTA) methods in activity recognition tend to fail, which makes this dataset challenging.

VLOG — is a multi-label binary classification of human-object interactions recently released with 114,000 videos and 30 classes (Fouhey et al. 2018). Classes correspond to objects, and labels of a class are 1 if a person has touched a certain object during the video, otherwise they are 0. It has recently been shown, that SOTA video based methods (Carreira et al. 2017) are outperformed on VLOG by image based methods like ResNet-50 (He et al. 2016), although these video methods outperform image based ResNet-50 on large-scale video datasets like the Kinetics dataset (Carreira et al. 2017). This suggests a gap between traditional datasets like Kinetics and the fine-grained dataset VLOG, making it particularly difficult.

EPIC Kitchens (EPIC) — is an egocentric video dataset recently released containing 55 hours recording of daily activities (Damen et al. 2018). This is the largest in first-person vision and the activities performed are non-scripted, which makes the dataset very challenging and close to real world data. The dataset is densely annotated and several tasks exist such as object detection, action recognition and action prediction. We focus on action recognition with 39'594 action segments in total and 125 actions classes (i.e verbs). Since the test set is not available yet we conducted our experiments on the training set (28'561 videos). We use the videos recorded by person 01 to person 25 for training (22'675 videos) and define the validation set as the remaining videos (5'886 videos).

For all datasets we rescale the input video resolution to 256×256. While training, we crop space-time blocks of 224×224 spatial resolution and L frames, with L=8 for the SS dataset and L=4 for VLOG and EPIC. We do not perform any other data augmentation. While training we extract L frames from the entire video by splitting the video into L sub-sequences and randomly sampling one frame

Methods	Top1
C3D + Avg (Goyal et al. 2017)	21.50
I3D (Goyal et al. 2017)	27.63
MultiScale TRN (Bolei et al. 2017)	33.60
Ours	**35.97**

Table 5.1 – **the Something-Something dataset.** Classification accuracy in % on the test set.

Methods	Top1
R18 (He et al. 2016)*	32.05
I3D-18 (Carreira et al. 2017)*	34.20
Ours	**40.89**

Table 5.2 – **Results on the EPIC Kitchens dataset.** Accuracy in % on the validation set – methods with * have been re-implemented).

per sub-sequence. The output sequence of size L is called a *clip*. A clip aims to represent the full video with less frames. For testing we aggregate results of 10 clips [2].

The ablation study is done by using the train set as training data and we report the result on the validation set. We compare against other SOTA approaches on the test set. For the ablation studies, we slightly decreased the computational complexity of the model: the base network (including activity and object heads) is a ResNet-18 instead of ResNet-50, a single clip of 4 frames is extracted from a video at test time.

Comparison with other approaches — Table 5.3 shows the performance of the proposed approach on the VLOG dataset. At the time of publication, we outperformed the state of the art on this challenging dataset by a margin of ≈4.2 points (44.7% accuracy against 40.5% by (He et al. 2016)). As mentioned above, traditional video approaches tend to fail on this challenging fine-grained dataset, providing inferior results. Table 5.1 shows performance on SS where we outperform the state of the art given by very recent methods (+2.3 points). On EPIC we re-implement standard baselines and report results on the validation set (Table 5.2) since the test set is not available. Our full method reports an accuracy of 40.89 and outperforms baselines by a large margin (≈+6.4 and ≈+7.9 points respectively for against CNN-2D and I3D based on a ResNet-18).

[2] We use *lintel* (Duke 2018) for decoding video on the fly.

	mAP	bag	bed	bedding	book/papers	bottle/tube	bowl	box	brush	cabinet	cell-phone	clothing	cup	door	drawers	food	fork	knife	laptop	microwave	oven	pen/pencil	pillow	plate	refrigerator	sink	spoon	stuffed animal	table	toothbrush	towel
R50 (He et al. 2016)	40.5	29.7	68.9	65.8	64.5	58.2	33.1	22.1	19.0	23.9	54.0	45.5	28.6	49.2	28.7	49.6	19.4	37.5	62.9	48.8	23.0	36.9	39.2	12.5	55.9	58.8	31.1	57.4	26.8	39.6	22.9
I3D (Carreira et al. 2017)	39.7	24.9	71.7	71.4	62.5	57.1	27.1	19.2	33.9	20.7	50.6	45.8	24.7	54.7	19.1	50.8	19.3	41.9	54.0	27.5	21.4	37.4	42.9	12.6	42.5	60.4	33.9	46.0	23.5	59.6	34.7
Ours	44.7	30.2	72.3	70.7	64.9	59.8	38.2	24.6	26.3	22.4	64.5	47.2	35.4	57.9	25.2	48.5	24.5	40.2	72.0	54.1	26.5	39.9	48.6	15.2	53.5	60.7	36.8	52.8	27.9	64.0	37.6

Table 5.3 – **Results on VLOG.** Results on Hand/Semantic Object Interaction Classification (**Average Precision!** (**Average Precision!**) in % on the test set) on VLOG dataset. R50 and I3D implemented by (Fouhey et al. 2018).

Method	Object type	EPIC		VLOG		SS	
		obj. head	2 heads	obj. head	2 heads	obj. head	2 heads
Baseline	-	-	*38.33*	-	*35.03*	-	*31.31*
ORN	pixel	23.71	38.83	14.40	35.18	2.51	31.43
ORN	**COCO**	**29.94**	**40.89**	**27.14**	**37.49**	10.26	**32.12**
ORN-MLP	COCO	28.15	39.41	25.40	36.35	-	-
ORN	COCO-visual	28.45	38.92	22.92	35.49	-	-
ORN	COCO-shape	21.92	37.16	7.18	35.39	-	-
ORN	COCO-class	21.96	37.75	13.40	35.94	-	-
ORN	COCO-intra	29.25	38.10	26.78	36.28	-	-
ORN clique-1	COCO	28.25	40.18	26.48	36.71	-	-
ORN clique-3	COCO	22.61	37.67	27.05	36.04	-	-

Table 5.4 – **Ablation study.** With ResNet-18 backbone. Results in %: Top-1 accuracy for EPIC and SS datasets, and mAP for VLOG dataset.

5.4.2 Further analysis

Effect of object-level reasoning — Table 5.4 shows the importance of reasoning on the performance of the method. The baseline corresponds to the performance obtained by the activity head trained alone (inflated ResNet, in the ResNet-18 version for this table). No object level reasoning is present in this baseline. The proposed approach (third line) including an object head and the ORN module gains 0.8, 2.5 and 2.4 points compared to our baseline respectively on SS, on EPIC and on VLOG. This indicates that the reasoning module is able to extract complementary features compared to the activity head.

Using *semantically defined objects* proved to be important and led to a gain of 2 points on EPIC and 2.3 points on VLOG for the full model (6/12.7 points using the object head only) compared to an extension of (Santoro et al. 2017) operating on pixel level. This indicates importance of object level reasoning. The gain on SS is smaller (0.7 point with the full model and 7.8 points with the object head only) and can be explained by the difference in spatial resolution of the videos.

Object detections and predictions of the binary masks are done using the initial video resolution. The mean video resolution for VLOG is 660×1183 and for EPIC is 640×480 against 100×157 for SS. Mask-RCNN has been trained on images of resolution 800×800 and thus performs best on higher resolutions. The quality of the object detector is important for leveraging object level understanding then for the rest of the ablation study we focus on EPIC and VLOG datasets.

The function f_ϕ in Equation 5.3 is an important design choice in our model. In our proposed model, f_ϕ is recurrent over time to ensure that the ORN module captures long range reasoning over time, as shown in Equation 5.3. Removing the recurrence in this equation leads to an MLP instead of a (gated) RNN, as evaluated in row 4 of Table 5.4. Performance decreases by 1.1 point on VLOG and 1.4 points on EPIC. The larger gap for EPIC compared to VLOG and can arguably be explained by the fact that in SS actions cover the whole video, while solving VLOG requires detecting the right moment when the human-object interaction occurs and thus long range reasoning plays a less important role.

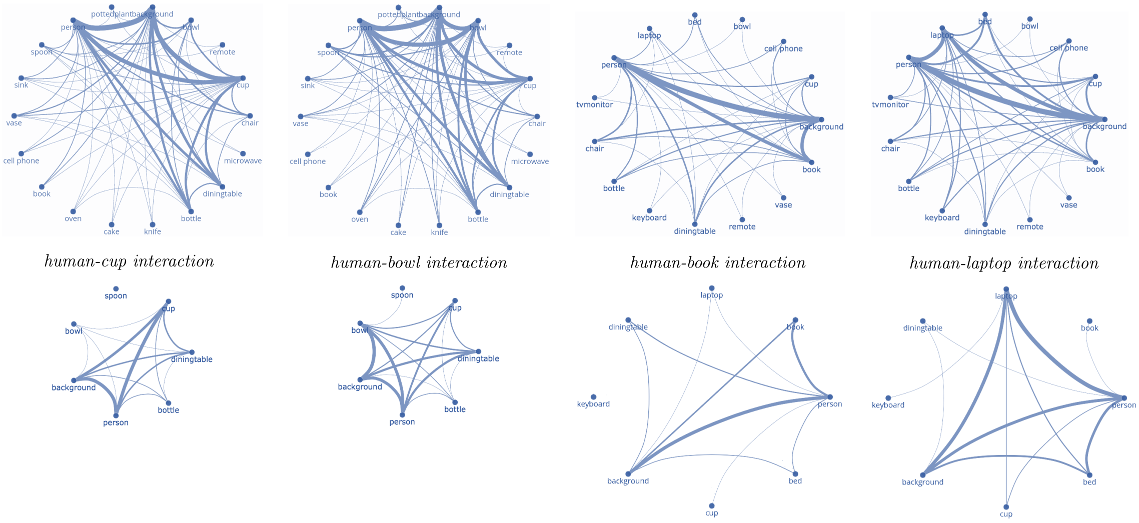
Visual features extracted from object regions are the most discriminative, however object shapes and labels also provide complementary information. Finally, the last part of Table 5.4 evaluates the effect of the cliques size for modeling the interactions between objects and show that pairwise cliques outperform cliques of size 1 and 3.

CNN architecture and kernel inflations — The convolutional architecture of the model was optimized over the validation set of the SS dataset, as shown in Table 5.5. The architecture itself (in terms of numbers of layers, filters etc.) is determined by pre-training on image classification. We optimized the choice of filter inflations from 2D to 2.5D or 3D for several convolutional blocks. This has been optimized for the single head model and using a ResNet-18 variant to speed up computation. Adding temporal convolutions increases performance up to 100% w.r.t. to pure 2D baselines. This indicates, without surprise, that motion is a strong cue. Inflating kernels to 2.5D on the input side and on the output side provided best performances, suggesting that temporal integration is required at a very low level (motion estimation) as well as on a very high level, close to reasoning. Our study also corroborates recent research in activity recognition, indicating that 2.5D kernels provide a good trade-off between high-capacity and learnable numbers of parameters. Finally temporal integration via RNN outperforms global average pooling over space and time.

Visualizing the learned object interactions — Figure 5.4 shows visualizations of the pairwise object relationships the model learned from data, in particular from the VLOG dataset. Each graph is computed for a given activity class.

Conv1			Conv2			Conv3			Conv4			Conv5			Aggreg		SS
2D	3D	2.5D	2D	3D	2.5D	2D	3D	2.5D	2D	3D	2.5D	2D	3D	2.5D	GAP	RNN	
✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	15.73
✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	-	-	✓	15.88
-	✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	✓	-	31.42
-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	✓	-	27.58
✓	-	-	✓	-	-	✓	-	-	✓	-	-	-	✓	-	✓	-	31.28
✓	-	-	✓	-	-	✓	-	-	-	✓	-	-	✓	-	✓	-	32.06
✓	-	-	✓	-	-	-	✓	-	-	✓	-	-	✓	-	✓	-	32.25
✓	-	-	✓	-	-	✓	-	-	✓	-	-	-	-	✓	✓	-	31.31
✓	-	-	✓	-	-	✓	-	-	-	-	✓	-	-	✓	✓	-	32.79
✓	-	-	✓	-	-	-	-	✓	-	-	✓	-	-	✓	✓	-	**33.77**
-	✓	-	✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	28.71
-	✓	-	-	✓	-	✓	-	-	✓	-	-	✓	-	-	✓	-	31.42
-	-	✓	✓	-	-	✓	-	-	✓	-	-	✓	-	-	✓	-	20.05
-	-	✓	-	-	✓	✓	-	-	✓	-	-	✓	-	-	✓	-	22.52

Table 5.5 – **Effect of the CNN architecture.** Choice of kernel inflations on a single head ResNet-18 network. Accuracy in % on the validation set of Something-Something is shown. 2.5D kernels are separable kernels: 2D followed by a 1D temporal.



human-cup interaction

human-bowl interaction

human-book interaction

human-laptop interaction

Figure 5.4 – **Object pairwise interactions.** Example of object pairwise interactions learned by our model on VLOG for four different classes. Objects co-occurrences are at the top and learned pairwise objects interactions are at the bottom. Line thickness indicates learned importance of a given relation. Interactions have been normalized by the object co-occurrences.

Visualizing of failure cases. Figure 5.5 and Figure 5.6 show failure cases. The model is either making confusion between semantically similar objects or having difficulties with small objects.

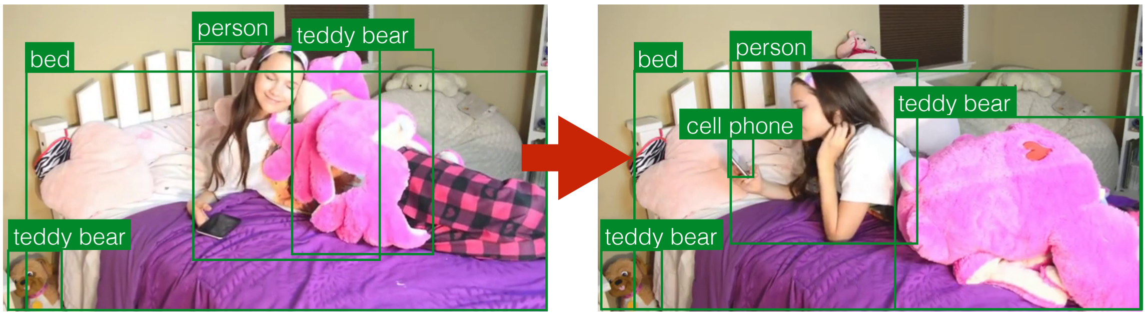


Figure 5.5 – **Failure cases 1.** Small sized objects. Our model detects a *cell phone* and a *person* but fails to detect *hand-cell-phone contact*.

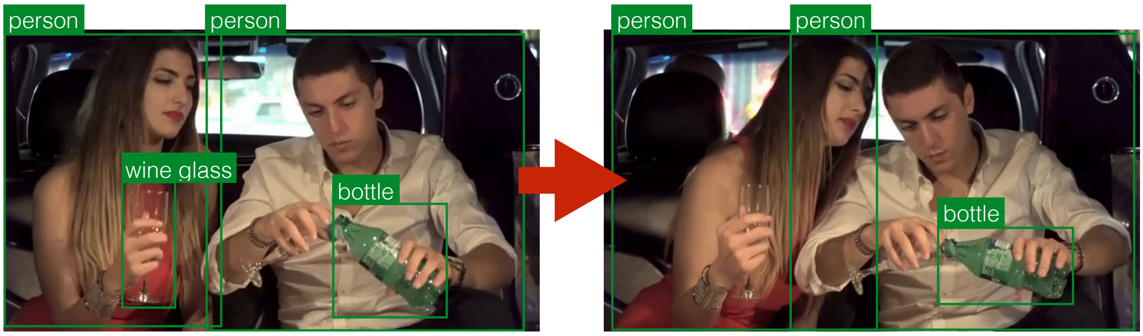


Figure 5.6 – **Failure cases 2.** Confusion between semantically similar objects. The model falsly predicts *hand-cup contact* instead of *hand-glass-contact* even though the *wine glass* is detected.

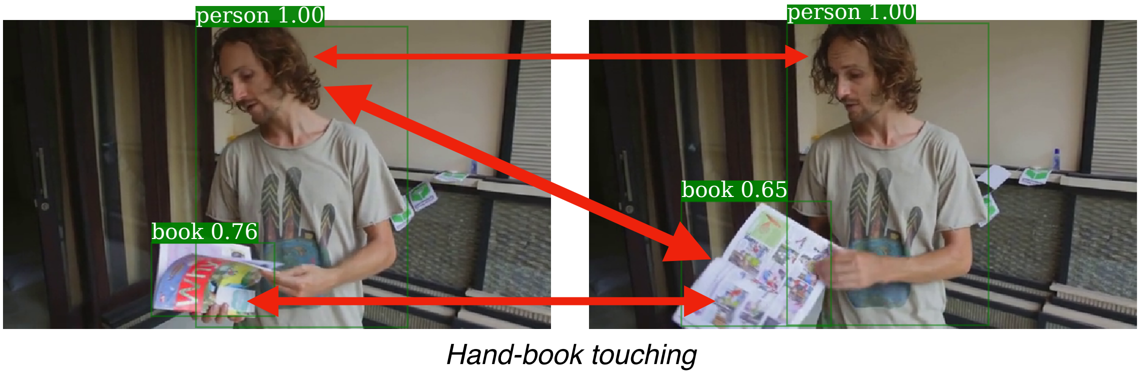


Figure 5.7 – **Learned object interactions 1.** The model learned that the most important object interactions for the class *hand-book touching* corresponds to the relation between a *person* and a *book*.

Visualizing of the important object interactions — Finally we visualize in Figure 5.7 and Figure 5.8 the most important learned object interactions. We apply a threshold for showing only the top object interactions.

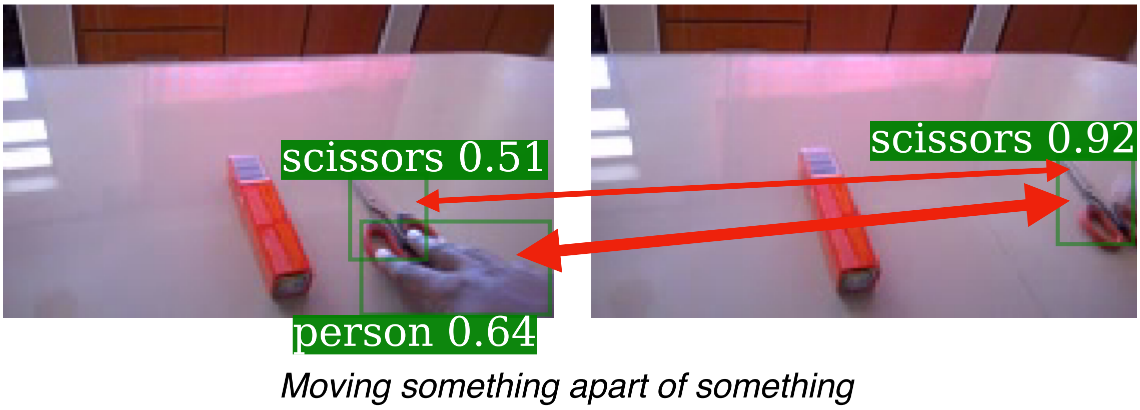


Figure 5.8 – **Learned object interactions 2.** The model detects that the most important object interactions correspond to relation between *person* and *scissors* for the high-level class *moving something apart of something*.

5.5 Conclusion

In this chapter, we present a method for activity recognition in videos which leverages object instance detections for visual reasoning on object interactions over time. The choice of reasoning over semantically well-defined objects is key to our approach and outperforms state of the art methods which reason on grid-levels, such as cells of convolutional feature maps. Temporal dependencies and causal relationships are dealt with by integrating relationships between different time instants. We evaluated the method on three difficult datasets, on which standard approaches do not perform well, and report SOTA results.

In the next chapter, we go one step further by moving towards a more challenging task towards visual reasoning. We still model the video content from an object-level perspective but with the underlying goal to estimate each object properties in an unsupervised manner for being able to estimate the causal relations between them.

Chapter 6

COPHY: COUNTERFACTUAL LEARNING OF PHYSICAL DYNAMICS

Chapter abstract

Understanding causes and effects in mechanical systems is an essential component of reasoning in the physical world. This chapter poses a new problem of counterfactual learning of object mechanics from visual input.

We develop the CoPhy benchmark to assess the capacity of the State Of The Art (SOTA) models for causal physical reasoning in a synthetic 3D environment and propose a model for learning the physical dynamics in a counterfactual setting. Having observed a mechanical experiment that involves, for example, a falling tower of blocks, a set of bouncing balls or colliding objects, we learn to predict how its outcome is affected by an arbitrary intervention on its initial conditions, such as displacing one of the objects in the scene. The alternative future is predicted given the altered past and a latent representation of the confounders learned by the model in an end-to-end fashion with no supervision of confounders.

We compare against feedforward video prediction baselines and show how observing alternative experiences allows the network to capture latent physical properties of the environment, which results in significantly more accurate predictions at the level of super human performance.

The work in this chapter has led to the publication of a conference paper:

- Fabien Baradel, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf (2020a). "CoPhy: Counterfactual Learning of Physical Dynamics". In: *Proceedings of the International Conference on Learning Representations (ICLR) - (spotlight presentation).*

Contents

6.1 Introduction

In Chapter 5, we demonstrated that describing video stream from an object-level point of view is helpful for producing more robust video representation. It allows us to discover important object interactions patterns. In this chapter, we extend this idea of modeling object interactions by moving towards a reasoning task.

Reasoning is an essential ability of intelligent agents that enables them to understand complex relationships between observations, detect affordances, interpret knowledge and beliefs, and to leverage this understanding to anticipate future events and act accordingly. The capacity for observational discovery of *causal effects* in physical reality and making sense of fundamental physical concepts, such as *mass*, *velocity*, *gravity*, *friction*, etc., may be one of differentiating properties of human intelligence that ensures our ability to leverage such experiences for robust *generalization* to new scenarios (Martin-Ordas et al. 2008).

In Section 2.1.3, we introduce the concept of *counterfactual reasoning* for expressing the causality that deals with a problem containing an *if* statement, which is untrue or unrealized.

Predicting the effect of the interventions based on the given observations without explicitly observing the effect of the intervention on data is a hard task and requires modeling of the causal relationships between the variable on which the intervention is performed and the variable whose alternative future should be predicted (Balke et al. 1994). Using counterfactuals has been shown to be a way to perform reasoning over causal relationships between variables in low dimensional spaces. However, it has been an unexplored direction for high dimensional signals such as videos.

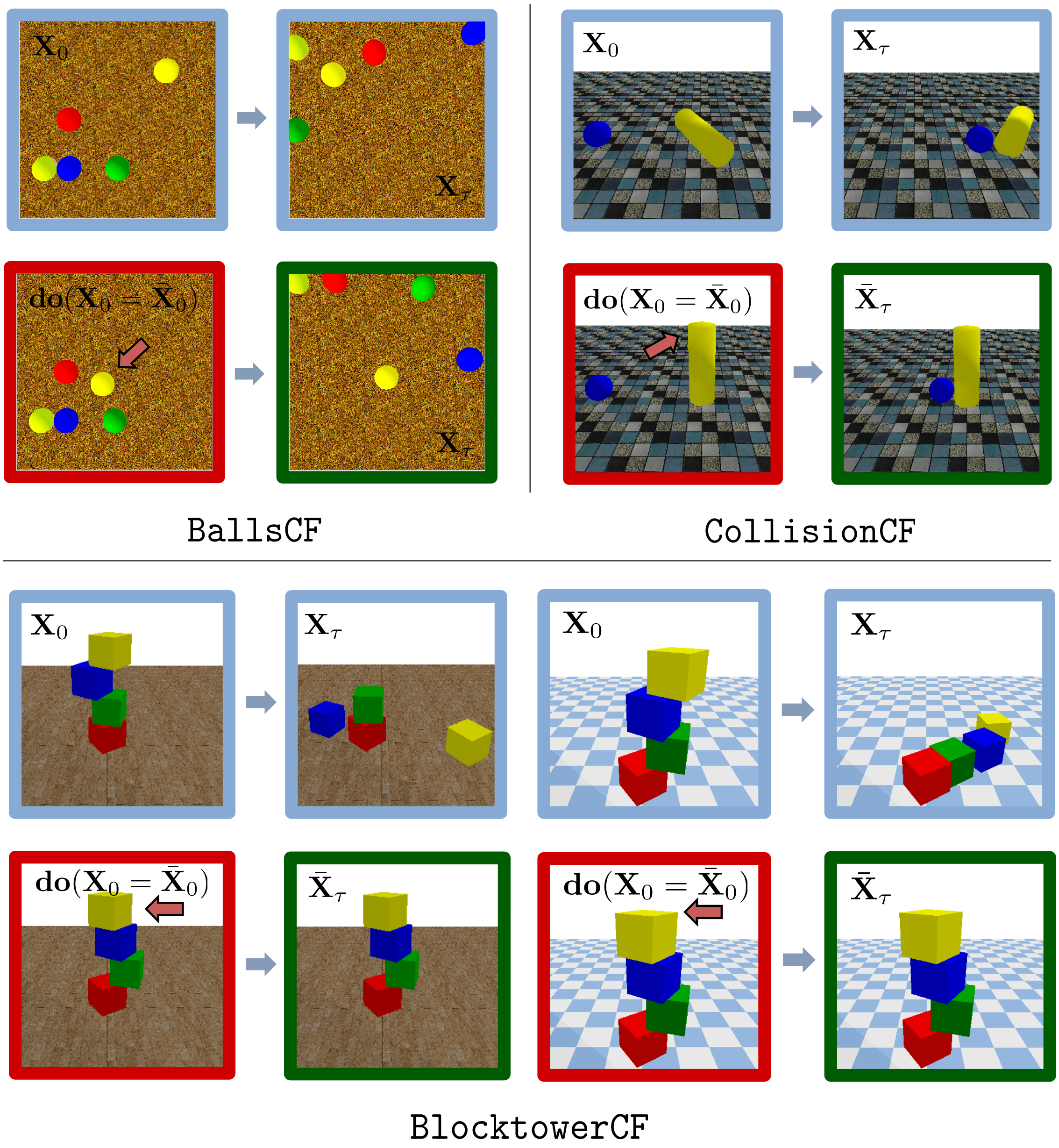


Figure 6.1 – **Overview of the proposed benchmark.** We train a model for performing counterfactual learning of physical dynamics. Given an observed frame $\mathbf{A} = X_0$ and a sequence of future frames $\mathbf{B} = X_{1:\tau}$, we ask how the outcome $\mathbf{B}$ would have changed if we changed X_0 to $\bar{X}_0$ by performing a *do*-intervention (e.g. changing the initial positions of objects in the scene).

In this chapter, we develop the **Co**unterfactual **Phy**sics benchmark (**CoPhy**) and propose a framework for causal learning of dynamics in mechanical systems with multiple degrees of freedom, as illustrated in Figure 6.1.

For a number of scenarios, such as *tower of blocks falling*, *balls bouncing against walls* or *objects colliding*, we are given the starting frame $\mathbf{A} = X_0$ and a sequence of following frames $\mathbf{B} = X_{1:\tau}$, where τ covers the range of 6 sec. The observed sequence $\mathbf{B}$, conditioned on the initial state $\mathbf{A}$, is a direct effect of the physical principles (such as *inertia*, *gravity* or *friction*) applied to the closed system, that cause the objects to change their positions and 3D poses over time.

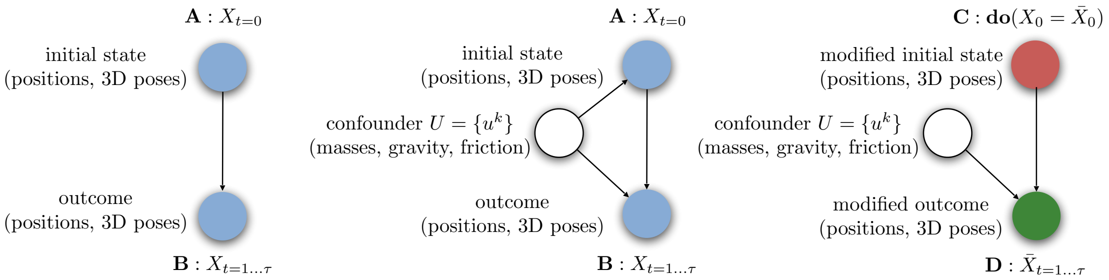
The task is formulated as follows: having observed the tuple $(\mathbf{A}, \mathbf{B})$, we wish to predict positions and poses of all objects in the scene at time $t{=}\tau$, *if we had changed the initial frame* X_0 *by performing an intervention*. The intervention is formalized by the *do-operator* introduced by Pearl *et al.* (Pearl 2009; Pearl et al. 2018) for dealing with causal inference (Spirtes 2010). In our case, it implies modification of the variable $\mathbf{A}$ to $\mathbf{C}$, defined as $\mathbf{C} = \mathbf{do}(X_0{=}\bar{X}_0)$. Accordingly, for each experiment in the CoPhy benchmark, we provide pairs of original sequences $X_{0:\tau}$ and their modified counterparts $\bar{X}_{0:\tau}$ sharing the same values of all confounders.

We note the fundamental difference between this problem of *counterfactual future forecasting* and the conventional setup of *feedforward future forecasting*, like video prediction (Mathieu et al. 2016). The latter involves learning spatio-temporal regularities and thereby predicting future frames $X_{1...\tau}$ from one or several past frame(s) X_0 (the causal chain of this problem is shown in Figure 6.2a). On the other hand, counterfactual forecasting benefits from *additional observations* in the form of the original outcome $X_{1:\tau}$ *before* the *do-operator*. This adds a *confounder variable* U into the causal chain (Figure 6.2b), which provides information not observable in frame X_0. For instance, in the case of the CoPhy benchmark, observing the pair $(\mathbf{A}, \mathbf{B})$ might give us information on the masses, velocities or friction coefficients of the objects in the scene, which otherwise cannot be inferred from frame $\bar{X}_0$ alone. Therefore, predicting the alternative outcome after performing counterfactual intervention then involves using the estimate of the confounder U together with the modified past $\mathbf{do}(X_0{=}\bar{X}_0)$.

Overall, we employ the idea of **counterfactual intervention in predictive models** and argue that counterfactual reasoning is an essential step towards human-like reasoning and general intelligence. More specifically, the key contributions of this chapter include:

- **a new task of counterfactual prediction** of physical dynamics from high-dimensional visual input, as a way to access capacity of intelligent agents for causal discovery;

- **a large-scale CoPhy benchmark** with three physical scenarios and 300k synthetic experiments including rendered sequences of frames, metadata (*object positions*, *angles*, *sizes*) and values of confounders (*masses*, *frictions*, *gravity*). This benchmark was specifically designed in **bias-free** fashion to make the counterfactual reasoning task challenging by optimizing the impact of the confounders on the outcome of the experiment. The dataset will be made publicly available.


A : $X_{t=0}$
initial state (positions, 3D poses)
outcome (positions, 3D poses)
B : $X_{t=1...\tau}$
A : $X_{t=0}$
initial state (positions, 3D poses)
confounder $U = \{u^k\}$ (masses, gravity, friction)
outcome (positions, 3D poses)
B : $X_{t=1...\tau}$
C : do($X_0 = \bar{X}_0$)
modified initial state (positions, 3D poses)
confounder $U = \{u^k\}$ (masses, gravity, friction)
modified outcome (positions, 3D poses)
D : $\bar{X}_{t=1...\tau}$

(a) feedforward future forecasting

(b) counterfactual future forecasting

Figure 6.2 – **Causal graph.** The difference between conventional video prediction (a) and counterfactual video prediction (b). The causal graph of the latter includes a confounder variable, which passes information from the original outcome to the outcome after *do-intervention*. The initially observed sequence $(\mathbf{A}, \mathbf{B})$ (on the left) and the counterfactual sequence after the *do*-intervention (on the right).

- **a counterfactual neural model** predicting an alternative outcome of a physical experiment given an intervention, by estimating the latent representation of the confounders. The model outperforms State Of The Art (SOTA) solutions implementing feedforward video prediction. It successfully extrapolates its behavior to unseen initial states and does not require supervision on the confounders. We provide extensive ablations on the different effects of key design choices and compare our results with human performance as evaluated in our studies, that show that the task is hard for humans to solve.

6.2 Related work

This chapter is inspired by a significant number of prior studies from several subfields, including visual reasoning, learning intuitive physics and perceived causality.

Intuitive physics — Fundamental studies on cognitive psychology have shown that humans perform poorly when asked to reason about expected outcomes of a dynamic based event, demonstrating striking deviations from Newtonian physics in their intuitions (McCloskey et al. 1983; McClooskey et al. 1980; McClooskey et al. 1983; Kubricht et al. 2017). The questions of approximating these mechanisms, learning from noisy observed and non-observed physical quantities (such as sizes or velocities vs masses or gravity), as well as justifying importance of explicit physical concepts vs cognitive constructs in intelligent agents have been raised and explored in recent works on deep learning (Wu et al. 2015). We summarized these works in Section 2.2.7.

Other physics benchmarks and simulators — The main objective for the creation of our benchmark is (a) to focus specifically on evaluating capabilities of state-of-the-art models for performing counterfactual reasoning, (b) to be unbiased in terms of distributions of parameters to be estimated and balanced with respect to possible outcomes, and (c) to have sufficient variety in terms of scenarios and latent physical characteristics of the scene that are not visually observed and therefore can act as confounders. To the best of our knowledge, none of existing intuitive physics benchmarks have these properties. IntPhys (Riochet et al. 2018) focuses on a high level task of estimating physical plausibility in a black box fashion and modeling out distribution events at test time. Phyre (Bakhtin et al. 2019) is an environment for solving physics-based puzzles, where achieving sample efficiency may implicitly require counterfactual reasoning. However, this component is not explicitly evaluated. Construction of parallel data with several alternative outcomes is not straightforward, and the trivial baseline performance levels are not easy to estimate. Adapting these benchmarks to counterfactual reasoning would require significant refactoring and changing the logic of the data sampling.

Perceptual causality — As already discussed in Section 2.1.3, causal reasoning gained mainstream attention relatively recently in the Machine Learning (ML) community (Lopez-Paz et al. 2017a; Lopez-Paz et al. 2017b; Kocaoglu et al. 2018; Rojas-Carulla et al. 2018; Mooij et al. 2016; Schölkopf et al. 2012), due to limitations of statistical learning becoming increasingly apparent (Pearl 2018; Lake et al. 2017). The concept of *perceived causality* has been however explored in cognitive psychology (Michotte 1963), where human subjects have be shown to consistently report causal impressions not aligned with underlying physical principles of the events (Gerstenberg et al. 2015; Kubricht et al. 2017). Exploiting the *colliding objects* scenario as a standard testbed for these studies led to discovery of a number of cognitive biases, e.g. Motor Object Bias (i.e. false perceived association of object's velocity with its mass).

In this chapter, we bring the domains of visual reasoning, intuitive physics and perceived causality together in a single framework to tackle the new problem of counterfactual learning of physical dynamics. Following prior literature (Battaglia et al. 2013), we also compare counterfactual learning with human performance and expect that, similarly to learning intuitive vs Newtonian physics, modeling perceived vs true causality will get more attention from the ML community in the future.

6.3 Benchmark

In this paper we investigate visual reasoning problems involving a set of K physical objects and their interactions, while considering a specific setting of **learning counterfactual prediction** with the objective of estimating objects' *alternative* 3D positions from images after the *do-intervention*.

We introduce the **Co**unterfactual **Phy**sics benchmark suite (**CoPhy**) for counterfactual reasoning of physical dynamics from raw visual input. It is composed of three tasks based on three physical scenarios: `BlocktowerCF`, `BallsCF` and `CollisionCF`, defined similarly to existing SOTA environments for learning intuitive physics: *Shape Stack* (Groth et al. 2018), *Bouncing balls* environment (Chang et al. 2017) and *Collision* (Ye et al. 2018) respectively. This was done to ensure natural continuity between the prior work in the field and the proposed counterfactual formulation.

Each scenario includes training and test samples, that we call *experiments*. Each experiment is represented by two sequences of τ synthetic RGB images (covering the time span of 6 sec at 5 fps):

- an **observed sequence** $X=\{X_0, \ldots, X_\tau\}$ demonstrates evolution of the dynamic system under the influence of laws of physics (gravity, friction, etc.), from its initial state X_0 to its final state X_τ. For simplicity, we denote **A** the initial state X_0 and **B** the observed outcome $\{X_1, \ldots, X_\tau\}$;
- a **counterfactual sequence** $\bar{X}=\{\bar{X}_0, \ldots, \bar{X}_\tau\}$, where $\bar{X}_0$ (**C**) corresponds to the initial state X_0 after the *do-intervention*, and $\bar{X}_1, \ldots, \bar{X}_\tau$ (**D**) correspond to the counterfactual outcome.

A **do-intervention** is a *visually observable* change introduced to the initial physical setup X_0 (such as, for instance, object displacement or removal).

Finally, the physical world in each experiment is parameterized by a set of *visually unobservable* quantities, or **confounders** (such as object masses, friction coefficients, direction and magnitude of gravitational forces), that cannot be uniquely estimated from a single time step. Our dataset provides ground truth values of all confounders for evaluation purposes. However, we do not assume access to this information during training or inference, and do not encourage it.

Each of the three scenarios in the CoPhy benchmark is defined as follows (see Figure 6.1 for illustrations).

`BLOCKTOWERCF` — Each experiment involves K=3 or K=4 stacked cubes, which are initially at resting (but potentially unstable) positions. We define three different confounder variables: *masses*, $m \in \{1, 10\}$ and *friction coefficients*, $\mu \in \{0.5, 1\}$, for each block, as well as *gravity components* in x and y direction, $g_{x,y} \in \{-1, 0, 1\}$. The *do-interventions* include block displacement or removal. This set contains

146k sample experiments corresponding to 73k different geometric block configurations.

BALLSCF — Experiments show K bouncing balls (K=2 . . . 6). Each ball has an initial random velocity. The confounder variables are the *masses*, $m \in \{1, 10\}$, and the *friction coefficients*, $\mu \in \{0.5, 1\}$, of each ball. There are two *do-operators*: block displacement or removal. There are in total 100k experiments corresponding to 50k different initial geometric configurations.

COLLISIONCF — This set is about moving objects colliding with static objects (balls or cylinders). The confounder variables are the *masses*, $m \in \{1, 10\}$, and the *friction coefficients*, $\mu \in \{0.5, 1\}$, of each object. The *do-interventions* are limited to object displacement. This scenario includes 40k experiments with 20k unique geometric object configurations.

Given this data, the problem can be formalized as follows. During *training*, we are given the quadruplets of visual observations $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ (through sequences X and $\bar{X}$, including GT object positions for supervision), but do *not* have access to the values of the confounders. During *testing*, the objective is to reason on new visual data unobserved at training time and to predict the counterfactual outcome $\mathbf{D}$, having observed the first sequence $(\mathbf{A}, \mathbf{B})$ and the modified initial state $\mathbf{C}$ after the do-intervention, which is known.

The CoPhy benchmark is by construction **balanced and bias free** w.r.t. (1) global statistics of all confounder values within each scenario, (2) distribution of possible outcomes of each experiment over the whole set of possible confounder values (for a given do-intervention). We make sure that the data does not degenerate to simple regularities which are solvable by conventional methods predicting the future from the past. In particular, for each experimental setup, we enforce existence of at least two different confounder configurations resulting in significantly different object trajectories. This guarantees that *estimating the confounder variable is necessary for visual reasoning on this dataset*.

More specifically, we ensure that, for each experiment, the set of possible counterfactual outcomes is balanced w.r.t. (1) *tower stability* for `BlocktowerCF` and (2) *distribution of object trajectories* for `BallsCF` and `CollisionCF`. As a result, the `BlocktowerCF` set, for example, has $50 \pm 5\%$ of stable and unstable counterfactual configurations. The exact distribution of stable/unstable examples for each confounder in this scenario is shown in Figure 6.3.

All images for this benchmark have been rendered into the visual space (RGB, depth and instance segmentation) at a resolution of 448×448 px with PyBullet (only RGB images are used in this chapter). We ensure diversity in visual appearance between experiments by rendering the pairs of sequences over a set of randomized backgrounds. The ground truth physical properties of each object (3D pose, 4D quaternion angles, velocities) are sampled at a higher frame rate (20 fps) and also stored. The training / validation / test split is defined as 0.7 : 0.2 : 0.1 for each of the three scenarios.

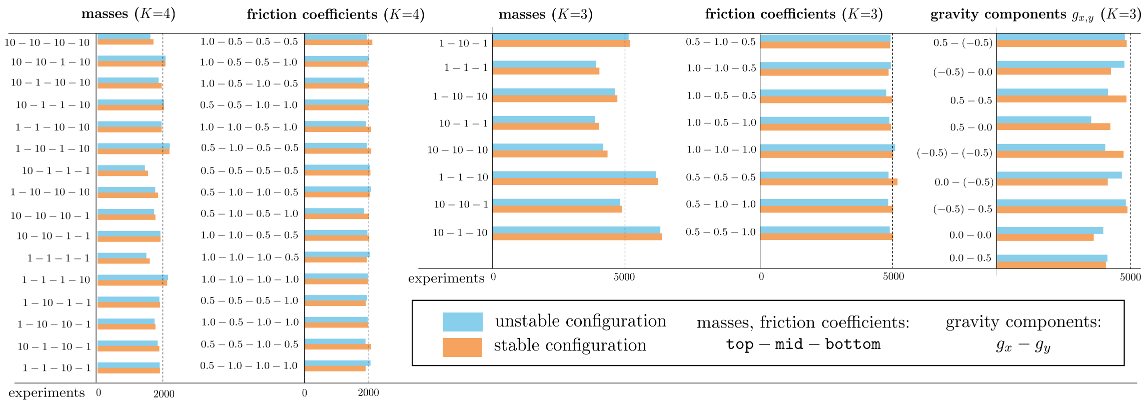


Figure 6.3 – **Stability distribution for each confounder variable.** For heights K=3 and K=4 of the `BlockTowerCF` task. Masses, friction cooefficients: 2 configurations per block, 2^K total; gravity: 3 configurations for each axis $\in\{x, y\}$, 9 total.

6.4 Model

The task as described in Section Section 6.3 requires reasoning from visual inputs.We propose a single neural model which can be trained end-to-end, as shown in Figure 6.4. We address this problem by adding strong inductive biases to a deep neural network, structuring it in a way to favor counterfactual reasoning. More precisely, we add structure for:

- estimating physical properties from images,
- modelling interactions between objects through Graph Convolutional Network (GCN),
- estimating latent representations of the confounder variables,
- exploiting these representations for predictions of the output object positions.

At this point we would like to stress again, that the representation of the confounders U is latent and discovered from data without supervision.

6.4.1 Unsupervised estimation of the confounders

While our method is capable of handling raw RGB frames as input, its internal reasoning is done on estimated representations in object-centric viewpoints. We train a convolutional neural network to detect the K objects and their 3D position in the scene, denoted as $O{=}\{o^k\}, k{=}0\ldots\ K{-}1$ where o^k corresponds to the 3D position of object k. The de-rendering module is explained in Section 6.4.3.

Predicting the future of a given block k requires modelling its interactions (through friction and collisions) with the other blocks in the scene, which we do with GCN (Kipf et al. 2017; Battaglia et al. 2018). The set of K objects in the scene is represented as a graph $\mathcal{G}{=}(\mathcal{V},\mathcal{E})$ where the nodes V are associated to objects $\{o^k\}$, and the object interactions to edges $(o^k, o^j) \in \mathcal{E}$ in the fully-connected graph. Object embeddings $\{o^k\}$ are updated classically and as follows, resulting in new embeddings $\{\tilde{o}^k\}$:

$$e^k = \frac{1}{|\Omega_k|} \sum_{o^j \in \Omega_k} f(o^k, o^j) \tag{6.1}$$

$$e = \frac{1}{K} \sum_k e^k \tag{6.2}$$

$$\tilde{o}^k = g(o^k, e^k, e) \tag{6.3}$$

where Ω_k is the set of neighboring objects of $o^k_.$ $f(.)$ and $g(.)$ are non-linear mappings (Multi-Layer Perceptron (MLP)s), and their inputs are by default concatenated. For simplicity, in what follows, the update of an object o^k with a GCN will be denoted by $\tilde{o}^k = GCN(o^k, O)$.

As mentioned above, we want to infer a latent representation U of the *confounding* quantities for each object k given the input sequence $X_{0:\tau}$ (the original past **A** and the original outcome **B**), without any supervision. This latent representation U is trained end-to-end by optimizing the counterfactual prediction loss. To this end, we pass the updated object states $\tilde{o}^k$ through a recurrent network to model the temporal evolution of this representation. In particular, we run a dedicated RNN for each object, each object maintaining its own hidden state h^k:

$$h^k_t = \phi(\tilde{o}^k_t, h^k_{t-1}) \tag{6.4}$$

where we index objects and states with subscript t indicating time, and ϕ is a Gated Recurrent Unit (GRU) (gate equations have been omitted for simplicity). The recurrent network parameters are shared over objects k, which results in a model which is invariant to the number of objects present in the set. This allows to use do-operators which change the number of objects in the scene (removal). We set the latent representation of the confounders to be the set $U{=}\{u^k\}$, where $u^k \triangleq h^k_\tau$ is the temporally last hidden state of the recurrent network.

6.4.2 Trajectory prediction gated by stability

We predict the counterfactual outcome **D**, i.e. the 3D positions of all objects of the sequence $\bar{X}_{1:\tau}$, with a recurrent network, which takes into account the confounders U. We cast this problem as a sequential prediction task, at each time step t predicting the residual position Δ^k_t w.r.t. to position $t{-}1$, i.e. the velocity

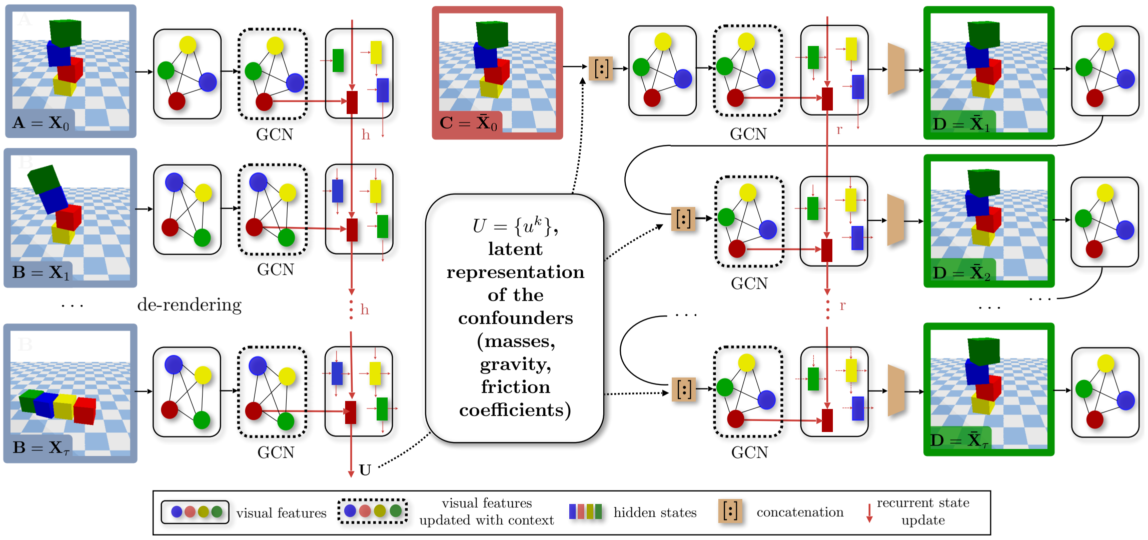


Figure 6.4 – **CoPhyNet.** Our model learns counterfactual reasoning in a *weakly supervised* way: while we supervise the do-operator, we do *not* supervise the confounder variables (masses, frictions, gravity). Input images of the original past (**A**) and the original outcome (**B**) are de-rendered into latent representations which are converted into fully-connected attributed graphs. A GCN updates node features to augment them with contextual information, which is integrated temporally with a set of Recurrent Neural Network (RNN), one for each object, running over time. The last hidden RNN state is taken as an estimate of the confounder U. A second set of GCN+RNN predicts residual object positions (**D**) using the modified past (**C**) and the confounder representation U. For clarity we draw arrows for the red object only. *Not shown: stability prediction and gating.*

vector. As in the rest of the model, this prediction is obtained object-wise, albeit with explicit modelling of the inter-object relationships through a graph network. More precisely,

$$\tilde{\bar{o}}_t^k = GCN(\bar{o}_t^k, \{[\bar{o}_t^k : u^k]\}) \tag{6.5}$$

$$r_t^k = \psi(\tilde{\bar{o}}_t^k, r_{t-1}^k) \tag{6.6}$$

$$\Delta_t^k = \boldsymbol{W} r_t^k, \tag{6.7}$$

where r_t^k is the hidden state of the GRU network denoted by ψ, and $\boldsymbol{W}$ is the weight matrix of a linear output layer. GCN is a GCN as described in Equation 6.3 and thereafter.

At each moment of time, each object can either remain stationary or move under the influence of external physical forces or by inertia. The first task for the model is therefore to detect which objects are moving (i.e. affected by the environment) and then estimate parameters of the motion if it occurs. This is aligned well with the concepts of **whether-causation** and **how-causation** defined in the field of *perceived causality* (Gerstenberg et al. 2015). In our work, the *whether-cause* is estimated in the form of a binary stability indicator s_t^k described below (for each object, updated at each time step) that is then leveraged to gate the object position predictor (*how-cause* estimator):

$$\bar{o}_{t+1}^k = \bar{o}_t^k + \sigma\left(\frac{1 - s_t^k}{\lambda}\right)\Delta_t^k, \tag{6.8}$$

where $\sigma(.)$ is the sigmoid function and λ is a sparsifying temperature term.

Counterfactual estimation of stability — Estimation of object stability s_t^k is a counterfactual problem, as stability depends on the physical properties, and therefore on the latent confounder representation u_k. We combine the confounders $U{=}\{u^k\}$ with the past after do-intervention (C), encoded in object states denoted as $\bar{O}_t{=}\{\bar{o}_t^k\}$ at time step t. In particular, for each node we concatenate its object features with its confounder representation and we update the resulting object state with a graph network to take into account inter-object relationships:

$$s_t'^k = GCN([\bar{o}_t^k : u^k], \{[\bar{o}_t^k : u^k]\}) \tag{6.9}$$

$$s_t^k = \boldsymbol{V} s_t'^k, \tag{6.10}$$

where s_t^k corresponds to the logits of stability of object k at time t and $\boldsymbol{V}$ is the weight matrix of a linear layer (for simplifying notations we omit bias here and in the rest of the chapter).

6.4.3 Neural de-rendering

We train a convolutional neural network to detect the K blocks and their 3D position in the scene, denoted as $O{=}\{o^k\}, k{=}0 \ldots K{-}1$ where o^k corresponds to the 3D position of object k. The Convolutional Neural Network (CNN) is inspired by recent methods for object detection (He et al. 2017) and takes as input an RGB image of resolution $224{\times}224$. All convolution kernels are of size $3{\times}3$. We restrict our object detector to detections of a single instance per object type. The extension to detecting multiple object of the same type would be straightforward using from example a Region Proposal Network (RPN) (He et al. 2017). We define a *double convolution* as a stack of a convolution layer, a batch norm layer and a Rectified Linear Unit (ReLU) layer, repeated two times, where the output number of channels corresponds to the number of channels in the hidden layers. We run a double convolution followed by a max pooling operator of kernel 2 two times where the number of output channels is respectively 64 and 128. And finally we run a double convolution with 256 output channels leading to a feature map of size 256×56×56. We have to detect objects of K categories (K=4 in our case) and their 3D position in the environment. We design K different heads with the architecture which corresponds to a double convolution with 512 output channels followed by a convolution of kernel size 28×28 and with 128 output channels, a batch norm layer and a ReLU. Each head outputs a feature map of size 128×1×1 which is resized into a vector of size 128. Finally we detect the presence of an object using a linear regression and we regress its 3D coordinates.

6.4.4 Training & Architectures

Training — We first pre-train the de-rendering module alone without the model parts responsible for reasoning in the graph space. In particular, we de-render images $\bar{X}$ randomly sampled from A, B, C and D into its object representations $O=\{o^k\}_{k=1...K}$ and train with the following supervised loss:

$$\mathcal{L}_{\text{derender}} = \sum_{k=1}^{K} \mathcal{L}_{mse}(o^k, o^{k*}) \tag{6.11}$$

where $\bar{o}^{k*}$ are the ground truth 3D positions and $\mathcal{L}_{mse}$ corresponds to the mean square error.

Then the full counterfactual prediction model is trained end-to-end in the graph space only (i.e. not over the derendering engine) with the following losses:

$$\mathcal{L}_{e2e} = \sum_{k=1}^{K} \mathcal{L}_{ce}(s^k, s^{k*}) + \sum_{t=0}^{\tau} \left[\sum_{k=1}^{K} \mathcal{L}_{mse}(\bar{o}_t^k, \bar{o}_t^{k*}) \right]$$

where $\mathcal{L}_{ce}$ is the binary cross entropy loss between the stability groundtruth of object k and its prediction.

Architectures — The de-rendering engine corresponds to a stack of 3×3 convolutions and max-pooling.

The mapping f and g are both MLP with respectively 4 and 2 layers. They both have hidden layer of size 32 and ReLU as activation function. ϕ is a GRU with 2 layers and a hidden state h of dimension 32. ψ is a GRU with 2 layers and a hidden state r of dimension 32.

Below are the descriptions of other networks used for our method:

- f, g — The mappings f and g are both MLP with 4 and 2 layers respectively. They both have hidden layers of size 32 and ReLU as activation function.
- ϕ — is a GRU with 2 layers and a hidden state h of dimension 32.
- ψ — is a GRU with 2 layers and a hidden state r of dimension 32.

	Scenario	Top	Middle	Bottom	Mean	σ
Human	Non-CF	108.89	53. 15	13.67	58.57	33.61
	CF	99.98	49.65	13.99	54.54	34.15
Copying	$C \rightarrow D$	65.41	25.51	7.36	32.76	N/A
	$B \rightarrow D$	92.60	35.85	18.54	49.00	N/A
CophyNet	Non-CF	77.97	33.10	2.39	37.81	N/A
	CF	**57.10**	**23.26**	**4.40**	**28.25**	N/A

Table 6.1 – **Comparison with human performance.** In the `BlockTowerCF` scenario obtained with AMT studies. We report 2D pixel error for each block, as well as global mean and variance σ (reference resolution 448 × 448) on the test set with K=3 blocks.

- Confounders — (mass and friction coefficients) are predicted with a single fully-connected layer on top of the "confounder" representation of each object denoted u^k.

6.5 Experiments

Training details — All models were implemented in PyTorch. We used the Adam optimizer (Kingma et al. 2015) and a learning rate of 0.001. For training the de-rendering pipeline, 200k frames were sampled for each of the three scenarios.

Human performance — We empirically measured human performance in the `BlockTowerCF` scenario with crowdsourcing (Amazon Mechanical Turk (AMT)). For this study, we have collected predictions from 100 participants, where each subject was given 20 assignments in both non-counterfactual (Figure 6.5) and counterfactual (Figure 6.6) settings. The human subjects were given 10 sec to click on the final positions of each block in the image **C** after the tower has fallen (or remained stable). The obtained quantitative results for both settings are reported in Table 6.1. We compare against copying baselines (i.e. predicting block positions in the frame **D** by either copying them from **C** or from **B**).

We observe that humans perform slightly better in the counterfactual setup after having observed the first dynamic sequence $(\mathbf{A}, \mathbf{B})$ together with **C** compared to the classical prediction where only **C** is shown. This behavior has also been previously observed in experiments on intuitive physics in cognitive psychology (Kubricht et al. 2017) that revealed poor human abilities to extrapolate physical dynamics from a single image. Similar human studies have also been conducted in (Battaglia et al. 2013) in a more simplistic setup of predicting the

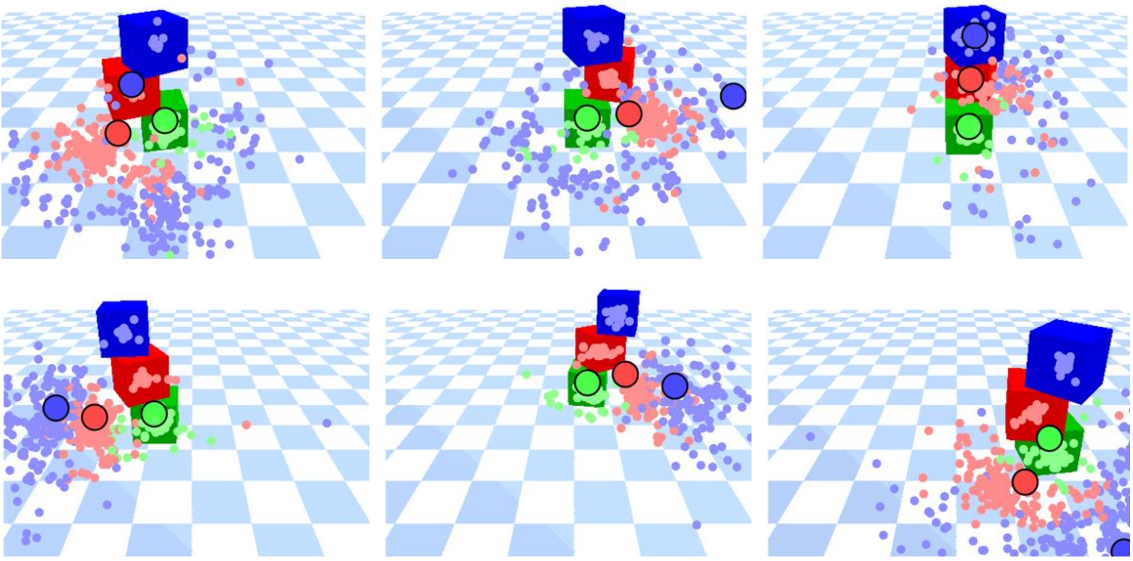


Figure 6.5 – **Examples for human performance 1.** On the task predicting the future (non counterfactual), as performed by mechanical turkers. Each turker has been confronted with the past only (a single image of block positions, shown). Dots correspond to human estimates of the objects' resting positions. Larger circles indicate ground truth final positions of each of the block.

direction of falling, where the authors also reported that the task appeared to be challenging for human subjects.

The empirical results indicate that the participants decisions may have been however driven by simple inductive biases, e.g. "observed (in)stability in $(\mathbf{A}, \mathbf{B})$"→"predict (in)stability in $(\mathbf{C}, \mathbf{D})$". The evidence for this approach is demonstrated qualitatively in Figure 6.6: the variance in predictions after having observed a stable sequence is decreased (first row), after having observed a falling case – increased (second row). In all cases, human performance remains inferior w.r.t. the copying baselines.

The last part of Table 6.1 shows results of our model (denoted by **CophyNet** in the rest of the discussion) after projecting the estimated 3D positions of all objects back into the 2D image space. CophyNet significantly outperforms both human subjects and copying baselines.

Performance and comparisons — We evaluate the counterfactual prediction performance of the proposed CophyNet model against various baselines (shown in Table 6.2-Table 6.3 separately for each of the three scenarios of the CoPhy benchmark). The evaluated Network Physics Engine (NPE) (Chang et al. 2017)

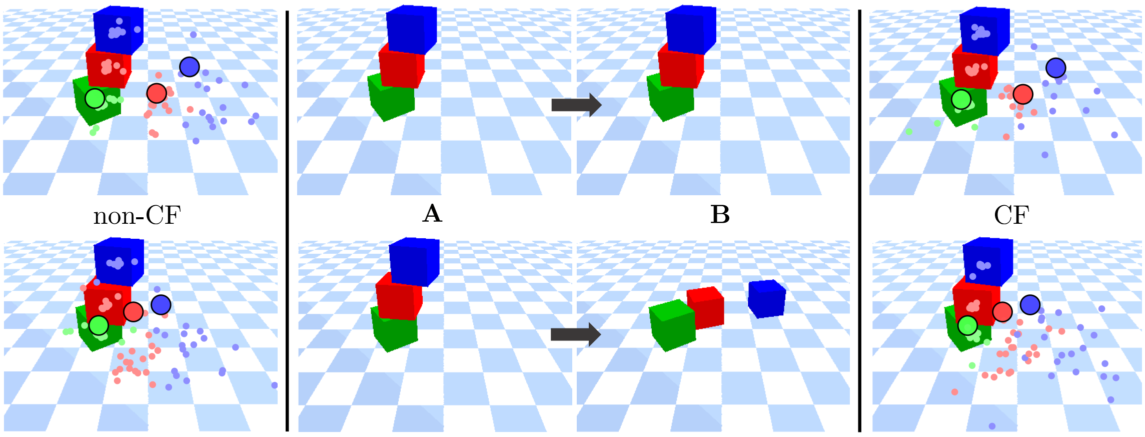


Figure 6.6 – **Examples for human performance 2.** On the task of counterfactual prediction, as performed by mechanical turkers. Each turker has been confronted with the data (A,B,C) — past, outcome, past after do-intervention. Dots correspond to human estimates of the objects' resting positions (outcome after do-intervention).

and Interaction Network (IN) (Battaglia et al. 2016), are both non-counterfactual baselines, that predict future block coordinates from past coordinates after do-intervention without taking the confounders into account. Both methods assume GT object positions are available as input at training and test time, so they directly work on GT positions. They both predict the next position of each object using a GCN. IN is modeling object pairwise interaction between all objects in the scene while NPE is taking into account only neighbours objects for estimating the object interactions.

Our method consistently outperforms NPE and IN by a large margin in all scenarios. The CophyNet model also usually (but not always) outperforms the augmented variants of these methods that include the GT confounder quantities as input (a not comparable setting).

Figure 6.7 illustrates several randomly sampled experimental setups and corresponding counterfactual predictions by the CophyNet model in the BlocktowerCF scenario.

Generalization — We evaluate the ability of the CophyNet model to generalize to new physical setups which were not observed in the training data. In Table 6.2 we show model performance on unseen confounder combinations and on unseen number of blocks in the BlocktowerCF scenario (lines marked with †). Our proposed solution generalizes well under unseen settings compared to other methods. In Table 6.4 we also demonstrate that our method outperforms the baselines by a large margin on unseen numbers of balls in the BallsCF setup. Finally, in the CollisionCF scenario (Table 6.3) we train on one type of moving

Train→Test	Copy C	Copy B	IN	NPE	CophyNet	*IN sup.*
3 → 3	0.470	0.601	0.318	0.331	**0.294**	*0.296*
3 → 3 †	0.365	0.592	0.298	0.319	**0.289**	*0.282*
3 → 4	0.754	0.846	0.524	0.523	**0.482**	*0.467*
4 → 4	0.735	0.852	0.521	0.528	**0.453**	*0.481*
4 → 4 †	0.597	0.861	0.480	0.476	**0.423**	*0.464*
4 → 3	0.480	0.618	0.342	0.350	**0.301**	*0.297*

Table 6.2 – **Results in** `BlocktowerCF`. MSE on 3D pose average over time. *IN sup.* methods in the last column exploit the ground truth confounder quantities as input and thus represent *a soft upper bound* (are not comparable). †Test confounder configurations not seen during training (50/50 split).

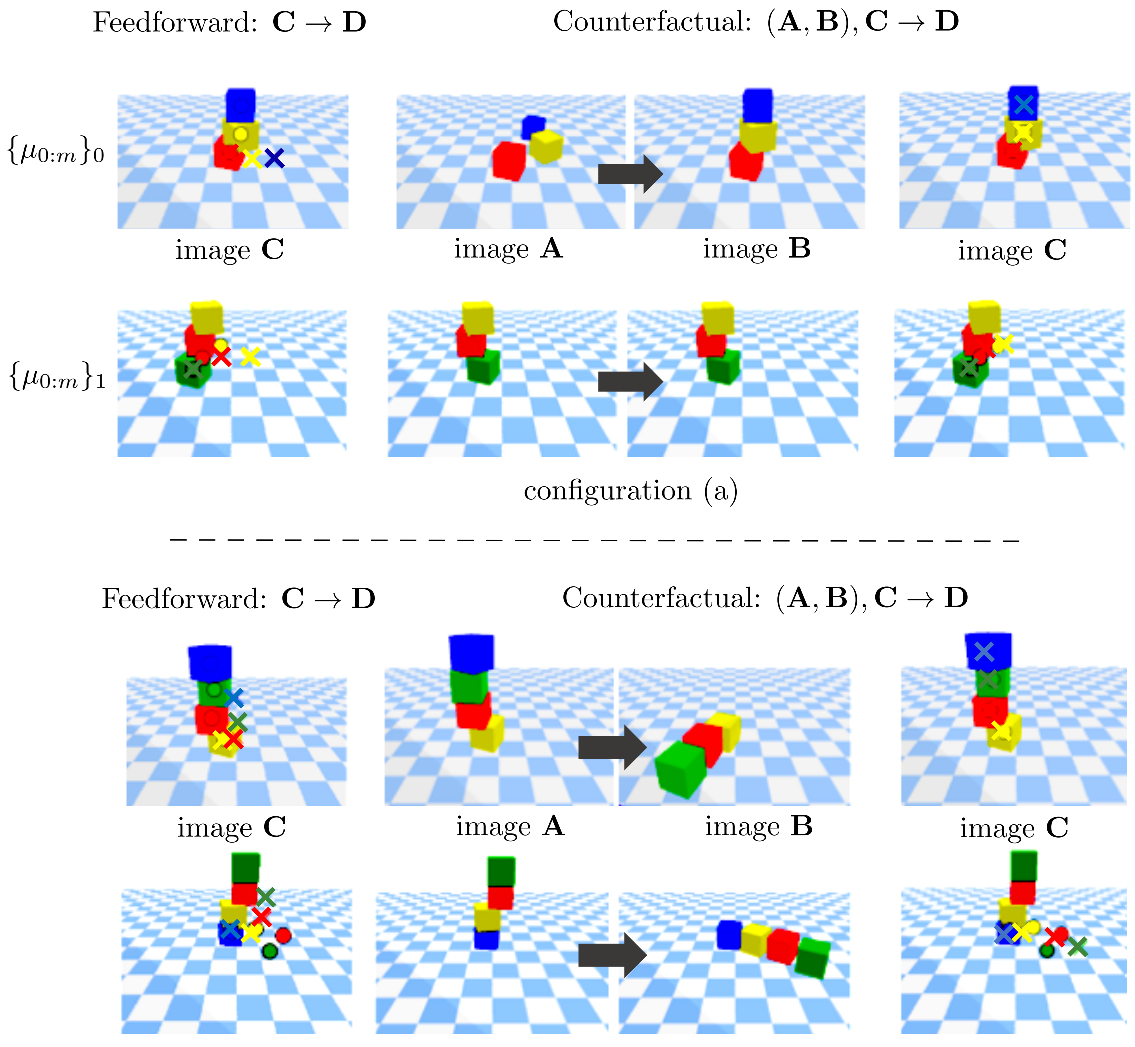
Train→Test	Copy C	Copy B	IN	NPE	CophyNet	*IN sup.*
all→all	4.370	0.665	0.701	0.697	**0.173**	*0.332*
sphere→cylinder	4.245	0.481	0.715	0.710	**0.220**	*0.435*
cylinder→sphere	4.571	0.932	0.720	0.699	**0.152**	*0.586*

Table 6.3 – **Results in** `CollisionCF`**.** MSE on 3D pose average over time. *IN sup.* methods in the last column exploit the ground truth confounder quantities as input and thus is not directly comparable (still showing inferior performance).

Train→Test	Copy C	Copy B	IN	NPE	CophyNet	*IN sup.*
4 → 2	7.271	3.267	5.060	4.989	**2.307**	*2.109*
4 → 3	6.820	2.865	4.895	4.901	**1.990**	*1.886*
4 → 4	6.538	2.688	4.785	4.821	**1.978**	*2.069*
4 → 5	6.221	2.568	4.732	4.817	**1.958**	*2.346*
4 → 6	6.045	2.488	4.661	4.668	**1.899**	*2.564*

Table 6.4 – **Results in** `BallsCF`**.** MSE on 3D pose average over time. *IN sup.* methods in the last column exploit the ground truth confounder quantities as input and thus is not directly comparable.

objects and test on another type (spheres vs cylinders). In this case we also show that our method is able to generalize to the new object types even when it has not seen such a combination of <moving-object, static-object >before. Our method is able to estimate the object properties when an object is moving or initially stable.


Feedforward: $\mathbf{C} \rightarrow \mathbf{D}$
Counterfactual: $(\mathbf{A}, \mathbf{B}), \mathbf{C} \rightarrow \mathbf{D}$
$\{\mu_{0:m}\}_0$
image $\mathbf{C}$
image $\mathbf{A}$
image $\mathbf{B}$
image $\mathbf{C}$
$\{\mu_{0:m}\}_1$
Feedforward: $\mathbf{C} \rightarrow \mathbf{D}$
Counterfactual: $(\mathbf{A}, \mathbf{B}), \mathbf{C} \rightarrow \mathbf{D}$
image $\mathbf{C}$
image $\mathbf{A}$
image $\mathbf{B}$
image $\mathbf{C}$

configuration (a)

configuration (b)

Figure 6.7 – **Visual examples of the counterfactual predictions.** Produced by CophyNet (in the BlocktowerCF scenario). Circles denote GT position and crosses correspond to predictions.

Impact of the confounder estimate — Our model does not rely on any supervision of the confounders; we do, however, explore what effect supervision could have on performance, as shown in Table 6.5. Adding the supervision increases the performance of the model for $K{=}3$ but the difference seems marginal (0.004 for $K{=}3$ and 0.020 for $K{=}4$). Directly feeding the confounder quantities as input leads to better performance, which is expected (but not comparable).

Model architecture — All design choices of CophyNet are ablated in Table 6.6 to fully illustrate the impact of each submodule. Estimating the stability once for the whole sequence **D** decreases the performance by 0.020 for $K{=}3$ and

Subset	Feedforward		Counterfactual	
	confounders: input	–	confounders: supervision	–
K=4	0.248	0.349	0.281	0.285
K=3	0.410	0.552	0.458	0.478

Table 6.5 – **Ablation study on** `BlockTowerCF`**.** Impact of the confounder estimate (MSE on 3D pose average over time, on the validation set). Feedforward prediction methods do not estimate the confounder, counterfactual methods do. We compare against soft upper bounds which use the ground truth confounder as input, or which supervise its estimation.

Method	$3 \rightarrow 3$	$3 \rightarrow 4$
Static gating	0.305	0.496
GCN replaced by MLP	0.289	0.764
Single-step prediction	0.295	0.492
CophyNet	0.285	0.478

Table 6.6 – **Ablation study on** `BlockTowerCF`**.** Impact of each component of our model (MSE on 3D pose average over time).

0.018 for K=4 compared to predicting the stability per object at each time step. Replacing the GCN by a MLP (i.e. concatenating the object representation) hurts the performance of the overall system by increasing the MSE by 0.286 when tested in the K=4 setting. Finally we compare our approach against a single-step counterfactual prediction. Non-surprisingly, predicting the future autoregressively in a step-by-step fashion turns out to be more effective than predicting the whole sequence at once.

Confounder estimation — After training for predicting the target CF sequences, we evaluate the quality of the learned latent representation. In this experiment, we predict the confounder quantities of each object (mass, friction coefficient) from their latent representation by training a simple linear classifier, freezing the weights of the whole network network. The obtained results are shown in Table 6.7. A prediction is correct if both the mass and the friction coefficient are correctly predicted. Our model outperforms the random baseline by a large margin suggesting that the confounder quantities are correctly encoded into the latent representation of each object during training.

Method	$3 \rightarrow 3$	$4 \rightarrow 4$
Random	25.0	25.0
CophyNet	65.7	68.9

Table 6.7 – **Ablations on BlockTowerCF: confounder prediction.** Masses, friction coefficients estimated from the joint latent representation U. Metric: 4-way classification accuracy: Random=random classification.

Method	$3 \rightarrow 3$	$3 \rightarrow 4$
Copy C	71.0	69.8
Copy B	69.9	68.5
GCN(C)	71.8	70.1
CophyNet	76.8	73.8

Table 6.8 – **Ablation study on BlockTowerCF.** Stability prediction (accuracy per block). With the ground truth values of the confounders provided as an input to the GCN, GCN(C) reaches performance of 85.4 and 77.3 in the $3 \rightarrow 3$ and $3 \rightarrow 4$ settings respectively (*a soft upper bound*, not comparable).

Stability prediction — We studied the performance of the stability estimation module in the BlockTowerCF scenario and compared it to several baselines, as shown in Table 6.8. Our method predicts stability of each block from the confounder estimate U and the frame $\mathbf{C}$. It outperforms the baselines estimating stability from a single input $\mathbf{C}$ or from the sequence $(\mathbf{A}, \mathbf{B})$ by a large margin, further indicating the efficiency of the confounder estimation and the complementarity of this non-visual information w.r.t. the visual observation $\mathbf{C}$.

Training from estimated positions — In Table 6.9 we report the impact of the de-rendering module on performance. In particular, we compare performance of our model (CophyNet w/o GT, as described in the main part of the paper) with a version where we use ground-truth positions (CophyNet GT) for training. During training time, GT positions are fed to the full model with the exception of the derendering module, not needed. For testing, we do, however, use positions estimated by the de-rendering module in both versions.

We can see, that training using the ground-truth positions gives slighly better performance than training from estimated positions, which is expected.

Train→Test	Copy C	Copy B	CophyNet GT	CophyNet w/o GT (=ours)
3 → 3	0.470	0.601	**0.294**	0.309
3 → 3 †	0.365	0.592	**0.289**	0.298
3 → 4	0.754	0.846	**0.482**	0.504

Table 6.9 – **Results in `BlocktowerCF`.** MSE on 3D pose average over time. We investigate different type of training procedures: from the ground-truth positions (GT) or from the estimated positions (w/o GT). †Test confounder configurations not seen during training (50/50 split).

6.6 Conclusion

We formulated a new task of counterfactual reasoning for learning intuitive physics from images, developed a large-scale benchmark suite and proposed a practical approach for this problem. The task requires to predict *alternative* outcomes of a physical problem (3D block positions) given the original past and outcome and an alternative past after do-intervention. Our suite of challenging benchmarks cannot be solved by classical methods predicting by extrapolation, as the alternative future depends on confounder variables, which are unobservable from a single image of the alternative past.

We train a neural model by supervising the do-operator, but not the confounders. Our experiments show that the CF setting outperforms conventional forecasting, and that the latent representation is related to the GT confounder quantities. We report human performance on this task, show its challenging nature, and corroborating the advantage of CF prediction also for humans.

We believe that counterfactual reasoning in high-dimensional spaces is a key component of AI and hope that our task will spawn new research in this area and thus contribute to bridging the gap between the causal reasoning and the Deep Learning (DL) literature. We also expect the proposed benchmark to become a testbed for perception modules in model based Reinforcement Learning (RL), which employs predictive models of an environment for learning agent behavior. Forward models are classically used in this context, but we conjecture that counterfactual reasoning will contribute to disentangling representations and inferring causal relationships between different factors of variation.

Chapter 7

CONCLUSION

Contents

7.1 Summary of Contributions

In this manuscript, we proposed several approaches for learning video representations. Ours contributions can be summarized into three categories.

Attention mechanisms — Finding features of interest in videos is a key component for recognizing fine-grained human actions. To address this problem we proposed different attention mechanisms both in the spatial and temporal domains with the underlying objective to collect relevant features in an automatic manner.

In Chapter 3, we exploited high-level information learned from articulated human pose data as context information to draw attention over pre-defined visual subparts of the video at each time instant. We defined a set of possible locations of interest, the human hands, and the contextual information extracted of pose features automatically weight visual features extracted around the hands. We have also introduced a temporal attention mechanism that automatically learns to pay more importance to timesteps which are more relevant to the task. We showed that such spatio-temporal attention mechanism improved the performance in term of human action recognition metrics and was also able to automatically discard noisy visual features such as for example provided by bad hand position estimation.

In Chapter 4, we went one step further by deploying an attention mechanism without using the external information of the articulated human pose at test time. Moreover, we did not restrain the attention mechanism and gave to the model the liberty to freely explore different points at each frame. This mechanism produced an unstructured "glimpse cloud", which corresponds to local spatio-temporal features and which are soft-assigned to a set of recognition workers based on the similarity with previously assigned features. Our method achieved State Of The

Art (SOTA) performance at time of publication of this work, and we showed that the key design choices were responsible for this performance improve: creating multiple glimpses per frame; sequential attention; and finally, distributed decision making.

Object interactions — While human behavior understanding is a necessary first step, we have also focused on more challenging tasks such human-object interactions in videos. In Chapter 5, we proposed a method able to predict the semantic content of a video by modeling the interactions between humans and objects detected in the scene. We represented the video as a fully-connected space-time graph where each node is an object and its attributes is described by its visual features, its spatial locations and its semantic information. We learned the human-objects interactions using Graph Convolutional Network (GCN) with the underlying goal to solve a problem of video classification. We showed that such a representation leads to improvements in the overall performance and, more importantly, it allows to highlight the importance of learned human-object interactions for each video label, which we illustrate in the form of new graphs.

Counterfactual prediction — Being able to associate a pre-defined label to a given visual content is a classical task in Computer Vision (CV), but training such systems does not necessarily lead to the emergence of high-level reasoning, as we discussed in Section 2.1. In Chapter 6, we went one step further by proposing to tackle more reasoning-like problems such as predicting counterfactuals. We used the same scene representation as presented in Chapter 5 by modeling a scene from its object interactions through space and time. We focused on videos representing physical interactions such as colliding objects or stacked cubes. Given a seen video we proposed to predict what would be the outcome of a minimal change in the initial geometrical configuration. To do so, we introduced an end-to-end method that, first, detects the objects present in the scene and regresses their positions. Second, the method estimates unobserved object properties such as mass or friction coefficients by examining the initial sequence. And third, we predict the outcome of the minimal change at the initial stage using the estimated object latent embeddings and the newly estimated geometrical object properties. We constructed a new large-scale benchmark composed of 3 challenging datasets for solving the counterfactual learning task in high-dimensional space.

7.2 Perspectives for Future Work

The work conducted during this thesis and the recent advances in the field open the door towards a wide range of possible perspectives and exciting research directions. In the following we highlight some possible future work directions related to the tasks tackled in this thesis as well as more general problems for the community.

Disentangling appearance and motion for human action recognition In this manuscript the task of human action recognition has been common thread during several consecutive chapters. We proposed spatio-temporal methods based on attention mechanisms and human-object relations for identifying human actions. However often a single image is sufficient for inferring a human action. We think that developing methods which bridge the gap between human-object interactions from static images (Gkioxari et al. 2018) and human action recognition in videos (Kay et al. 2017) could help to better understand how to model efficiently the motion information. It would be of high interest to automatically understand how motion features should be interpreted for being complementary to the information that can already be extracted from static images. This open the door toward a more sparse and disentangled representation of a human action. Such representation are nowadays widely used in generative modeling (Chen et al. 2017a) and could be a source of inspiration for discriminative systems.

Toward long and complex human activities Recognizing short human actions is a necessary first step towards the understanding of semantically meaningful complex activities. Such activities could be composed of multiple subjects in the scene whereas in this thesis most of the human actions that we have been working with where restricted to one or two subjects. Extracting semantic information at a human group level could be of high interest for better understanding the contextual information of a scene. Graph representations (Ibrahim et al. 2016; Ibrahim et al. 2018) have been proposed with success for representing human group activities where each node corresponds to a detected person. We think that understand human group information from a semantic level could be beneficial for better understanding human behavior since it allows to give us context information about the scene.

Counterfactual learning to more realistic scenarios In Chapter 6, we have presented a method for counterfactual learning operating from the pixel space. Our system was a two-stage procedure where we were first extracting object properties from the visual space and then predicting counterfactuals from this low-dimensional space. An interesting direction would be to simplify this system by predicting counterfactual from visual embeddings directly. We are at the moment working on this type of extension for a journal submission. While it seems mandatory at the moment to supervise the object properties such as the 3D pose for obtaining semantically meaningful embeddings, a long-term goal would be to predict counterfactuals from the visual space without this intermediate supervision. In some sense the model would need to find objects by itself and model the interactions without having access to the explicit modeling extracted from the object visual properties supervision. A final question concern the applications of counterfactual learning to real-word data. We have seen in Chapter 6 that they can be learned from simulated data but an interesting directions would be to extend them to real scenarios where we cannot supervise the *do*-operator and we do not have access to the intervention itself.

Self-supervised learning — While manual annotations are necessary for evaluating visual representations, constructing large-scale well-annotated datasets is costly and difficult in practice. The annotation practice becomes a real problem in itself when we want to annotate high-level concepts and/or very accurate labels. The mental representation of high-level concepts is not the same for every one and for example annotating the start and the end of long and complex human actions could be a ill-posed problem. One way to overcome this issue is to rely on self-supervised training procedure for constructing visual representations. While videos themselves provide signal supervisions such as the temporal dimension (Misra et al. 2016; Pickup et al. 2014), associated metadata such as audio, comments, title or descriptions is a potential source of supervision for learning video representations without using manual annotations. Instructional videos are an interesting type of videos where the audio content is more or less aligned with the visual content. While a lot of noise exists in such data, recent work (Miech et al. 2019a) shows that powerful video representations can be learned from a large-scale instructional video dataset (Miech et al. 2019b). This opens the door for improving video representations without the need for human annotations.

Efficient learning of semantic concepts — Current deep neural networks typically require hundreds or more training examples of the same semantic class for being able to learn a high-level representation. This is not efficient and not scalable to thousands of semantic concepts. Moreover, recent work (Tommasi et al. 2017; Torralba et al. 2011) shows that the learned representation is highly biased by the data collection process and more generally by the datasets biases. However, humans are capable of grabbing semantic concepts from a few examples only. This is mainly due to their ability to synthetize and to the use of existing knowledge of previously learned semantic concepts for learning new ones. Learning such robust representations is of interest and has been tackled so far by meta-learning methods (Finn et al. 2017) for solving classification task in few-shot learning scenarios where the number of training examples is limited. Recent works (Gidaris et al. 2019) show that self-supervised training objectives as auxiliary losses could be helpful for improving visual representation. We believe that generalizing such evaluation setup scenarios is a necessary step towards more robust and generalizable feature representations.

Unifying image and video representations — Classifying the video content in a pre-defined human actions class is a necessary first step similar to the object recognition task in image understanding. However, for real world applications we want to extract more precise information such as the spatio-temporal volume in the video containing the action of interest. Current action detection methods (Kalogeiton et al. 2017; Peng et al. 2016; Chéron et al. 2018) are multi-stage procedures and rely on a pose detector for detecting potential actors in the scene. Bridging the gap between human pose estimation, object detection and human-object interactions would be a good way to solve this problem in a end-to-end manner following a training strategy based on (Redmon et al. 2017).

Efficient video models — Training video models composed of 3D convolutions is extremely resource demanding. For example training I3D (Carreira et al. 2017) on the Kinetics dataset (Kay et al. 2017) requires 64 GPUs and the training time is approximately around 2 days. While being a real disaster in term of CO2 consumption, such large-scale experiments are reproducible only by big industrial labs. Recent works focus on training with a single GPU (Wu et al. 2019b), or speeding up the training procedure by restricting the convolution operations (Xie et al. 2017) or relying on 2D convolutions only using a shift procedure (Lin et al. 2019). While these methods show good performance at inference time using GPUs, they are really bad when it comes to inference time using CPUs only. Scaling video models to inference on CPUs or low-consumption embedded devices is an important step towards bringing perception module to robotics applications for examples.

Video analysis from compressed representation — Daily videos that we are watching on TV screens, laptops or smartphones are transferred through internet in a compressed representation (e.g. h264, vp9). Video compression algorithms are generally based on the removal of redundancy in the original signal. In short, I-frames are selected with a certain frequency and the entire image is compressed using an image compression technique while the remaining frames are encoded by computing the residuals and motion vector from the previous and consecutive I-frames (called P and B frames). Employing neural networks to reduce the size of the compressed video has shown great success over the last year (Ma et al. 2019) and improving these methods is an interesting research topic since 80% of the internet bandwidth is due to video traffic. On the other side, solving CV task (Wu et al. 2018) such as action recognition has shown to be possible from the compressed representation while speeding up the training and inference time. This opens new perspectives for building more efficient video representation from compressed signals.

Visual reasoning with learnable causal structure — Visual Question Answering (VQA) is a task of interest in the subfield of visual reasoning which consists at learning a joint vision/text embedding. Current methods are showing a clear overfitting to concepts and examples from the training set (Shrestha et al. 2019; Marino et al. 2019; Manjunatha et al. 2019). We think that such behaviors are observed because most recent approaches are tackling the problem by training a system to *associate* each visual example to its textual information. In Section 2.1 and Chapter 6, we have stressed out that causal relationships cannot be learned by employing the *association* level only. Hence the reasoning ability of such system are quite limited since no external knowledge is built during the learning of the system and they cannot solve typical activities such as *imagining* or *retrospection* mandatory for moving towards causal reasoning. Reasoning is about combining concepts and we think that this issue can be solved by introducing an external causal structure that the model should exploit and extend through its training procedure. This can be seen as constructing *ontologies* that correspond to a hierarchical structure of semantically meaningful concepts with their associated relationships. Hence new concepts should be constructed from existing basic concepts. Such causal structure can be seen as step towards data integration where ontologies are to data what grammar is to language. The construction of such external structure should enable reusable knowledge representation and the learning of such relations between concepts should enable automated reasoning about data.

Computer vision & Robotics — Finally, for the moment, we have highlighted different future work perspectives where the representation is learned in a purely passive way. Data are collected and a system is trained in a supervised or unsupervised manner depending on the data type. While it has been shown that babies learn largely by observation at the first stage of their lives (Gopnik et al. 2000), the role of interactions with the environment plays an important role in the development of perception, navigation, planing and more generally reasoning capacities. This is somehow related to the area of *active vision* initially suggested for improving the perceptual quality of tracking algorithm (Aloimonos et al. 1988). Tackling vision tasks in an active way could help at producing better predictions to cope with problems related to limited view points or occlusions. An active vision system also involves the introduction of visual attention mechanism which is an essential part of the human visual system. We have seen in Chapter 3 and Chapter 4 that visual attention mechanisms are particularly helpful for gathering important local information from a fix camera viewpoint and extending their behavior with systems that allow active camera view point would be of great interest. We think that bridging the gap between computer vision and robotics is a necessary step towards the construction of more robust visual representations (Pinto et al. 2016) and intelligent systems.

BIBLIOGRAPHY

Agarwal, Ankur and Bill Triggs (2005). "Recovering 3D Human Pose from Monocular Images". In: *IEE TPAMI* (cit. on p. 38).

Alexe, Bogdan, Thomas Deselaers, and Vittorio Ferrari (2010). "What is an object?" In: *CVPR* (cit. on p. 35).

Ali, S., A. Basharat, and M. Shah (2007). "Chaotic invariants for human action recognition". In: *ICCV* (cit. on p. 40).

Aloimonos, John, Isaac Weiss, and Amit Bandyopadhyay (1988). "Active vision". In: *International journal of computer vision* 1.4, pp. 333–356 (cit. on p. 144).

Ba, J., V. Mnih, and K. Kavukcuoglu (2015). "Multiple Object Recognition with Visual Attention". In: *ICLR* (cit. on pp. 4, 50).

Baccouche, M., F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt (2011). "Sequential Deep Learning for Human Action Recognition". In: *HBU* (cit. on pp. 43, 44, 65, 102).

Baccouche, Moez, Franck Mamalet, Christian Wolf, Christophe Garcia, and Atilla Baskurt (2012). "Spatio-Temporal Convolutional Sparse Auto-Encoder for Sequence Classification." In: *BMVC*, pp. 1–12 (cit. on p. 43).

Bahdanau, D., K. Cho, and Y. Bengio (2014). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *CoRR* abs/1409.0473. URL: `http://arxiv.org/abs/1409.0473` (cit. on pp. 47, 48, 70).

Bakhtin, Anton, Laurens van der Maaten, Justin Johnson, Laura Gustafson, and Ross Girshick (2019). "PHYRE: A New Benchmark for Physical Reasoning". In: *NeurIPS* (cit. on p. 122).

Balke, Alexander and Judea Pearl (1994). "Counterfactual Probabilities: Computational Methods, Bounds and Applications". In: *UAI* (cit. on pp. 6, 26, 118).

Baradel, F., C. Wolf, and J. Mille (2017a). "Pose-conditioned Spatio-Temporal Attention for Human Action Recognition". In: *Pre-print: arxiv:1703.10106* (cit. on p. 95).

Baradel, Fabien, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf (2020a). "CoPhy: Counterfactual Learning of Physical Dynamics". In: *Proceedings of the International Conference on Learning Representations (ICLR)* (cit. on pp. 7, 117).

Baradel, Fabien, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf (2020b). "CoPhy++: Counterfactual Learning of Physical Dynamics from Visual Input". In: *to be submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (cit. on p. 7).

Baradel, Fabien, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori (2018a). "Object Level Visual Reasoning in Videos". In: *Proceedings of the IEEE European Conference on Computer Vision (ECCV)* (cit. on pp. 7, 99).

Baradel, Fabien, Christian Wolf, and Julien Mille (2017b). "Human Action Recognition: Pose-based Attention draws focus to Hands". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV) - Workshop "Hands in Action"* (cit. on pp. 7, 63, 95).

Baradel, Fabien, Christian Wolf, and Julien Mille (2018b). "Human Activity Recognition with Pose-driven Attention to RGB". In: *Proceedings of the British Machine Vision Conference (BMVC)* (cit. on pp. 7, 63).

Baradel, Fabien, Christian Wolf, Julien Mille, and Graham Taylor (2018c). "Glimpse Clouds: Human Activity Recognition from Unstructured Feature Points". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 7, 79, 103).

Battaglia, Peter W., Jessica B. Hamrick, and Joshua B. Tenenbaum (2013). "Simulation as an engine of physical scene understanding". In: *PNAS*. Vol. 110. 45, 18327–18332 (cit. on pp. 122, 130).

Battaglia, Peter W., Razvan Pascanu, Matthew Lai, Danilo Jimenez Rezende, and Koray Kavukcuoglu (2016). "Interaction Networks for Learning about Objects, Relations and Physics". In: *NIPS* (cit. on pp. 58, 103, 132).

Battaglia, P.W., J.B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V.F. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, Ç. Gülçehre, F. Song, A.J. Ballard, J. Gilmer, G.E. Dahl, A. Vaswani, K. Allen, C. Nash, V/ Langston, C. Dyer, N. Heess, D. Wierstra, P. Kohli, M. Botvinick, O. Vinyals, Y. Li, and R. Pascanu (2018). "Relational inductive biases, deep learning, and graph networks". In: *arXiv preprint arXiv:1807.09244* abs/1806.01261 (cit. on pp. 57, 58, 126).

Bay, Herbert, Tinne Tuytelaars, and Luc Van Gool (2006). "SURF: Speeded Up Robust Features". In: *ECCV* (cit. on p. 30).

Beaudet, P. R. (1978). "Rotationally invariant image operators". In: *International Joint Conference on Pattern Recognition* (cit. on p. 42).

Ben-Younes*, Hedi, Rémi Cadène*, Nicolas Thome, and Matthieu Cord (2017). "MUTAN: Multimodal Tucker Fusion for Visual Question Answering". In: *iccv* (cit. on p. 59).

Ben-Younes*, Hedi, Rémi Cadène*, Nicolas Thome, and Matthieu Cord (2019). "MUREL: Multimodal Relational Reasoning for Visual Question Answering". In: *cvpr* (cit. on pp. 5, 60).

Bengio, Yoshua, Patrice Simard, and Paolo Frasconi (1994). "Learning long-term dependencies with gradient descent is difficult". In: *IEEE transactions on neural networks* 5.2, pp. 157–166 (cit. on p. 16).

Blank, M., L. Gorelick, E. Shechtman, M. Irani, and R. Basri (2005). "Actions as space-time shapes". In: *ICCV* (cit. on p. 41).

Bobick, A.F. and J.W. Davis (2001). "The recognition of human movement using temporal templates". In: *IEEE T-PAMI* (cit. on p. 41).

Bolei, Zhou, Andonian Alex Zhang, and Antonio Torralba (2017). “Temporal Relational Reasoning in Videos”. In: *arXiv preprint arXiv:1711.08496v1* (cit. on pp. 60, 61, 103, 111).

Boser, B., I. Guyon, and V. Vapnik. (1992). “A Training Algorithm for Optimal Margin Classifiers”. In: *COLT* (cit. on p. 32).

Bottou, L., J. Peters, J. Quiñonero-Candela, D.X. Charles, D.M. Chickering, E. Portugualy, D. Ray, P. Simard, and E. Snelson (2013). “Counterfactual Reasoning and Learning Systems: The Example of Computational Advertising”. In: *JMLR* (cit. on p. 6).

Bottou, Leon (1991). “Stochastic gradient learning in neural networks”. In: *Neuro-Nîmes* (cit. on p. 12).

Bottou, Leon, Frank E. Curtis, and Jorge Nocedal (June 2016). “Optimization Methods for Large-Scale Machine Learning”. In: *SIAM Review* 60 (cit. on p. 35).

Breiman, Leo, Jerome Friedman, Charles J Stone, and Richard A Olshen (1984). “Classification and regression trees”. In: *CRC press* (cit. on p. 32).

Brox, T. and J. Malik (2010). “Object segmentation by long term analysis of point trajectories”. In: *ECCV* (cit. on p. 43).

Burns, J Brian, Richard S. Weiss, and Edward M Riseman (1993). “View variation of point-set and line-segment features”. In: *IEEE PAMI* (cit. on p. 28).

C. Feichtenhofer A. Pinz, R. P Wildes (2017). “Spatiotemporal Multiplier Networks for Video Action Recognition”. In: *CVPR* (cit. on p. 44).

Caesar, Holger, Jasper Uijlings, and Vittorio Ferrari (2018). “Coco-stuff: Thing and stuff classes in context”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1209–1218 (cit. on p. 17).

Canny, John (1986). “A computational approach to edge detection.” In: *IEE PAMI* (cit. on p. 28).

Cao, Zhe, Tomas Simon, Shih-En Wei, and Yaser Sheikh (2017). “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields”. In: *CVPR* (cit. on p. 39).

Caron, Mathilde, Piotr Bojanowski, Armand Joulin, and Matthijs Douze (2018). “Deep clustering for unsupervised learning of visual features”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 132–149 (cit. on pp. 20, 21).

Carreira, Joao, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik (2016). “Human pose estimation with iterative error feedback”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4733–4742 (cit. on p. 38).

Carreira, Joao and Andrew Zisserman (2017). “Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset”. In: *CVPR* (cit. on pp. 40, 44, 91, 102, 110–112, 143).

Chang, Michael B, Tomer Ullman, Antonio Torralba, and Joshua B Tenenbaum (2017). “A Compositional Object-Based Approach to Learning Physical Dynamics”. In: *ICLR* (cit. on pp. 58, 123, 131).

Chen, Mickaël, Ludovic Denoyer, and Thierry Artières (2017a). "Multi-view data generation without view supervision". In: *arXiv preprint arXiv:1711.00305* (cit. on p. 141).

Chen, Xianjie and Alan L Yuille (2014). "Articulated pose estimation by a graphical model with image dependent pairwise relations". In: *NIPS* (cit. on p. 38).

Chen, Xinlei, Ross Girshick, Kaiming He, and Piotr Dollár (2019). "Tensormask: A Foundation for Dense Object Segmentation". In: *ICCV* (cit. on p. 36).

Chen, Zhu, Zhao Yanpeng, Huang Shuaiyi, Tu Kewei, and Ma Yi (2017b). "Structured Attentions for Visual Question Answering". In: *IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 59).

Chéron, Guilhem, Jean-Baptiste Alayrac, Ivan Laptev, and Cordelia Schmid (2018). "A flexible model for training action localization with varying levels of supervision". In: *Advances in Neural Information Processing Systems*, pp. 942–953 (cit. on p. 143).

Chéron, Guilhem, Ivan Laptev, and Cordelia Schmid (2015). "P-cnn: Pose-based cnn features for action recognition". In: *Proceedings of the IEEE international conference on computer vision*, pp. 3218–3226 (cit. on p. 47).

Cho, K., A. Courville, and Y. Bengio (2015). "Describing Multimedia Content using Attention-based Encoder-Decoder Networks". In: *IEEE-T-Multimedia* 17, pp. 1875–1886 (cit. on pp. 50, 64).

Cho, Kyunghyun, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio (2014). "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches". In: *arXiv preprint arXiv:1507.05738*, pp. 103–111. arXiv: `1409.1259 [cs.CL]`. URL: `http://aclweb.org/anthology/W/W14/W14-4012.pdf` (cit. on p. 84).

Chung, Junyoung, Çağlar Gülçehre, Kyunghyun Cho, and Yoshua Bengio (2014). *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. Tech. rep. Arxiv report 1412.3555. Presented at the Deep Learning workshop at NIPS2014. Université de Montréal (cit. on p. 102).

Clowes, M. B. (1971). "On seeing things". In: *Artificial Intelligence* (cit. on p. 28).

Csurka, G., C. Dance, L. Fan, J. Willamowski, and C. Bray (2004). "Visual categorization with bags of keypoints". In: *ECCV Workshop* (cit. on p. 31).

Dai, Bo, Yuqi Zhang, and Dahua Lin (2017). "Detecting Visual Relationships with Deep Relational Networks". In: *cvpr* (cit. on p. 56).

Dai*, Zihang, Zhilin Yang*, Yiming Yang, William W. Cohen, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov (2019). *Transformer-XL: Language Modeling with Longer-Term Dependency*. URL: `https://openreview.net/forum?id=HJePno0cYm` (cit. on p. 54).

Dalal, N. and B. Triggs (2005). "Histograms of oriented gradients for human detection". In: *CVPR* (cit. on pp. 34, 35, 41).

Dalal, N., B. Triggs, and C. Schmid (2006). "Human detection using oriented histograms of flow and appearance". In: *ECCV* (cit. on p. 42).

Damen, Dima, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray (2018). "Scaling Egocentric Vision: The EPIC-KITCHENS Dataset". In: *arXiv preprint arXiv:1804.02748* (cit. on pp. 8, 110).

Davis, L. (1975). "A survey of edge detection techniques". In: *Computer Graphics and Image Processing* (cit. on p. 28).

Dawid, A Philip (2000). "Causal inference without counterfactuals". In: *Journal of the American statistical Association* 95.450, pp. 407–424 (cit. on p. 26).

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (June 2018). "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (cit. on pp. 20, 54).

Dias, Philipe Ambrozio, Zhou Shen, Amy Tabb, and Henry Medeiros (2019). "Freelabel: A publicly available annotation tool based on freehand traces". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, pp. 21–30 (cit. on p. 17).

Dollar, P., V. Rabaud, G. Cottrell, and S. Belongie (2005). "Behavior recognition via sparse spatiotemporal features". In: *VS-PETS* (cit. on p. 42).

Donahue, Jeff, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Trevor Darrell, and Kate Saenko (2015). "Long-term recurrent convolutional networks for visual recognition and description." In: *CVPR* (cit. on p. 44).

Donahue, Jeff, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell (2014). "Decaf: A deep convolutional activation feature for generic visual recognition". In: *ICML* (cit. on p. 33).

Du, Yong, Yun Fu, and Liang Wang (2015a). "Skeleton based action recognition with convolutional neural network". In: *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*. IEEE, pp. 579–583 (cit. on p. 46).

Du, Yong, Wei Wang, and Liang Wang (June 2015b). "Hierarchical recurrent neural network for skeleton based action recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1110–1118 (cit. on pp. 46, 67, 82, 94, 95).

Duda, Richard O and Peter E Hart (1972). "Use of the hough transformation to detect lines and curves in pictures". In: *Communications of the ACM* (cit. on p. 28).

Duke, Brendan (2018). *Lintel: Python Video Decoding*. https://github.com/dukebw/lintel (cit. on p. 111).

Efros, A. A., A. C. Berg, G. Mori, and J. Malik (2003). "Recognizing action at a distance". In: *ICCV* (cit. on pp. 40, 41).

Eslami, S. M. Ali, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, koray kavukcuoglu, and Geoffrey E Hinton (2016). "Attend, Infer, Repeat: Fast Scene Understanding with Generative Models". In: *Advances in Neural Information Processing Systems* (cit. on p. 50).

Evangelidis, G., G. Singh, and R. Horaud (2014). "Skeletal Quads:Human action recognition using joint quadruples". In: *ICPR*, pp. 4513–4518 (cit. on p. 95).

Everingham, M., L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman (2010). "The pascal visual object classes (VOC) challenge". In: *IJCV* (cit. on pp. 32, 34).

Eweiwi, A., M.S. Cheema, C. Bauckhage, and J. Gall (2014). "Efficient Pose-Based Action Recognition". In: *ACCV*, pp. 428–443 (cit. on p. 75).

Fathi, A. and G. Mori (2008). "Action recognition by learning mid-level motion features". In: *CVPR* (cit. on p. 41).

Feichtenhofer, Christoph, Haoqi Fan, Jitendra Malik, and Kaiming He (2019). "SlowFast Networks for Video Recognition". In: *ICCV* (cit. on pp. 45, 46).

Feichtenhofer, Christoph, Axel Pinz, and AP Zisserman (2016). "Convolutional two-stream network fusion for video action recognition". In: *CVRP* (cit. on p. 44).

Felzenszwalb, P., D. McAllester, and D. Ramanan (2008). "A discriminatively trained and multiscale and deformable part model". In: *CVPR* (cit. on pp. 34, 35, 38).

Fernando, B., E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars (2015). "Modeling video evolution for action recognition". In: *CVPR* (cit. on p. 43).

Ferrari, V., M. Marin-Jimenez, and A. Zisserman (2008). "Progressive search space reduction for human pose estimation". In: *CVPR* (cit. on p. 41).

Finn, Chelsea, Pieter Abbeel, and Sergey Levine (2017). "Model-agnostic meta-learning for fast adaptation of deep networks". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, pp. 1126–1135 (cit. on p. 142).

Fischler, M. and R. Elschlager (1973). "The Representation and Matching of Pictorial Structures". In: *IEEE TRANSACTIONS ON COMPUTERS* (cit. on p. 38).

Fleuret, François, Ting Li, Charles Dubout, Emma K. Wampler, Steven Yantis, and Donald Geman (2011). "Comparing machines and humans on a visual categorization test." In: *Proceedings of the National Academy of Sciences of the United States of America* 108 43, pp. 17621–5 (cit. on pp. 5, 100).

Forsyth, D. A. and M. Fleck (1997). "Body plans". In: *CVPR* (cit. on p. 38).

Fouhey, David F., Weicheng Kuo, Alexei A. Efros, and Jitendra Malik (2018). "From Lifestyle VLOGs to Everyday Interactions". In: *CVPR* (cit. on pp. 8, 103, 110, 112).

Fukui, Akira, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach (2016). "Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding." In: *emnlp*. The Association for Computational Linguistics (cit. on p. 51).

Fukushima, K. (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological cybernetics* 36.4, pp. 193–202 (cit. on pp. 2, 14).

Gaidon, A., Z. Harchaoui, and C. Schmid (2013). “Temporal Localization of Actions with Actoms”. In: *IEEE TPAMI* (cit. on p. 42).

Gerstenberg, Tobias, Noah D. Goodman, David A. Lagnado, and Joshua A. Tenenbaum (2015). “How, whether, why: Causal judgments as counterfactual contrasts”. In: *Annual Conference of the Cognitive Science Society* (cit. on pp. 122, 128).

Gibson, J.J. (1950). “The Perception of the Visual World”. In: *Houghton Mifflin* (cit. on p. 41).

Gidaris, Spyros, Andrei Bursuc, Nikos Komodakis, Patrick Pérez, and Matthieu Cord (2019). “Boosting Few-Shot Visual Learning with Self-Supervision”. In: *Proceedings of the IEEE International Conference on Computer Vision* (cit. on p. 142).

Gidaris, Spyros, Praveer Singh, and Nikos Komodakis (2018). “Unsupervised representation learning by predicting image rotations”. In: *arXiv preprint arXiv:1803.07728* (cit. on pp. 20, 21).

Girdhar, Rohit and Deva Ramanan (2017). “Attentional Pooling for Action Recognition”. In: *NIPS*, pp. 34–45 (cit. on pp. 4, 47).

Girshick, Ross (2015). “Fast R-CNN”. In: *ICCV* (cit. on pp. 36, 56).

Girshick, Ross, J. Donahue, T. Darrell, and J. Malik (2014). “Rich feature hierarchies for accurate object detection and semantic segmentation”. In: *CVPR* (cit. on p. 36).

Gkioxari, G., B. Hariharana, R. Girshick, and J. Malik (2014). “Using k-poselets for detecting people and localizing their keypoints”. In: *CVRP* (cit. on p. 38).

Gkioxari, Georgia, Ross Girshick, Piotr Dollár, and Kaiming He (2018). “Detecting and recognizing human-object interactions”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8359–8367 (cit. on p. 141).

Gopnik, Alison, Andrew N Meltzoff, and Patricia K Kuhl (2000). *The scientist in the crib: What early learning tells us about the mind*. William Morrow Paperbacks (cit. on pp. 17, 144).

Goyal, Raghav, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic (Oct. 2017). “The "Something Something" Video Database for Learning and Evaluating Visual Common Sense”. In: *ICCV* (cit. on pp. 103, 110, 111).

Grauman, Kristen and Trevor Darrell (2005). “The pyramid match kernel: Discriminative classification with sets of image features”. In: *ICCV* (cit. on p. 32).

Graves, A. and J. Schmidhuber (2009). “Offline handwriting recognition with multidimensional recurrent neural networks”. In: *NIPS* (cit. on p. 67).

Graves, Alex, Greg Wayne, and Ivo Danihelka (Oct. 2014). “Neural Turing Machines”. In: arXiv: `1410.5401 [cs.NE]` (cit. on pp. 51–53).

Groth, Oliver, Fabian B. Fuchs, Ingmar Posner, and Andrea Vedaldi (2018). “ShapeStacks: Learning Vision-Based Physical Intuition for Generalised Object Stacking”. In: *ECCV* (cit. on pp. 55, 123).

Gu, Chunhui, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik (2017). “AVA: A Video Dataset of Spatio-temporally Localized Atomic Visual Actions”. In: *Arxiv*. URL: https://arxiv.org/abs/1705.08421 (cit. on p. 103).

Gupta, A, J Martinez, Little J, and Woodham R (2014). “3d pose from motion for cross-view action recognition via non- linear circulant temporal encoding”. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 94).

Hale, J (2019). “More than 500 hours of content are now being uploaded to YouTube every minute”. In: *Santa Monica, CA: Tubefilter* (cit. on p. 1).

Harris, C. and M. Stephens (1988). “A combined corner and edge detector”. In: *Alvey vision conference* (cit. on p. 41).

Hartigan, J. A. and M. A. Wong (1979). “A k-means clustering algorithm”. In: *JSTOR: Applied Statistics* (cit. on p. 17).

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (2017). “Mask R-CNN”. In: *ICCV* (cit. on pp. 8, 34, 36, 37, 107, 128).

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2016). “Deep residual learning for image recognition”. In: *CVPR* (cit. on pp. 33, 91, 110–112).

Hénaff, Olivier J, Ali Razavi, Carl Doersch, SM Eslami, and Aaron van den Oord (2019). “Data-efficient image recognition with contrastive predictive coding”. In: *arXiv preprint arXiv:1905.09272* (cit. on p. 20).

Hinton, Geoffrey E and Richard S Zemel (1994). “Autoencoders, minimum description length and Helmholtz free energy”. In: *Advances in neural information processing systems*, pp. 3–10 (cit. on p. 19).

Hjelm, R Devon, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio (2018). “Learning deep representations by mutual information estimation and maximization”. In: *arXiv preprint arXiv:1808.06670* (cit. on p. 21).

Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long short-term memory”. In: *Neural computation* 9.8, pp. 1735–1780 (cit. on pp. 16, 66, 82, 102).

Hogg, David C. (1983). “Model-based vision: a program to see a walking person”. In: *Image Vision Computational* (cit. on p. 38).

Hou, Y., Z. Li, P. Wang, and W. Li (2016). “Skeleton Optical Spectra Based Action Recognition Using Convolutional Neural Networks”. In: *IEEE Transactions on Circuits and Systems for Video Technology* (cit. on p. 67).

Hu, Dichao (2019). “An introductory survey on attention mechanisms in NLP problems”. In: *Proceedings of SAI Intelligent Systems Conference*. Springer, pp. 432–448 (cit. on p. 49).

Hu, J., S. Lu, and S. Gang (2018). "Squeeze-and-excitation networks". In: *CVPR* (cit. on p. 54).

Hu, J., W.-S. Zheng, J.-H. Lai, and J. Zhang (2015). "Jointly Learning Heterogeneous Features for RGB-D Activity Recognition". In: *CVPR*, pp. 5344–5352 (cit. on p. 95).

Hu, R., J. Andreas, Rohrbach R., T. Darrell, and K. Saenko (2017). "Learning to reason: End-to-end module networks for visual question answering". In: *ICCV* (cit. on p. 58).

Hubel, D. H. and T. N. Wiesel (1959). "Receptive fields of single neurones in the cat's striate cortex". In: *The Journal of Physiology* (cit. on pp. 2, 28).

Hudson, D. A. and C. D. Manning (2018a). "Compositional attention networks for machine reasooning". In: *ICLR* (cit. on p. 58).

Hudson, D.A. and C.D. Manning (2018b). "Compositional Attention Networks for Machine Reasoning". In: *ICLR* (cit. on pp. 59, 103).

Hudson, Drew and Christopher D Manning (2019). "Learning by Abstraction: The Neural State Machine". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., pp. 5903–5916. URL: http://papers.nips.cc/paper/8825-learning-by-abstraction-the-neural-state-machine.pdf (cit. on p. 60).

Huffman, D. A. (1971). "Impossible objects and nonsense sentences". In: *Machine Intelligence* (cit. on p. 28).

Hume, David (1748). *An Enquiry Concerning Human Understanding* (cit. on p. 26).

Huttenlocher, Daniel P (1987). "Object recognition using alignment". In: *ICCV* (cit. on p. 28).

Ibrahim, Mostafa S and Greg Mori (2018). "Hierarchical relational networks for group activity recognition and retrieval". In: *Proceedings of the European conference on computer vision (ECCV)*, pp. 721–736 (cit. on p. 141).

Ibrahim, Mostafa S, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori (2016). "A hierarchical deep temporal model for group activity recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1971–1980 (cit. on p. 141).

Ioffe, Sergey and Christian Szegedy (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift". In: *ICML* (cit. on p. 33).

Itti, L., C. Koch, and E. Niebur (1998). "A Model of Saliency-Based Visual Attention for Rapid Scene Analysis". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 20.11, pp. 1254–1259 (cit. on pp. 4, 47).

Jaderberg, Max, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu (2015). "Spatial Transformer Networks". In: *NIPS*, pp. 2017–2025 (cit. on p. 85).

Jain, A., A. R. Zamir, S. Savarese, and A. Saxena (2016). "Structural-RNN: Deep Learning on Spatio-Temporal Graphs". In: *CVPR* (cit. on p. 82).

Janner, Michael, Sergey Levine, William T. Freeman, Joshua B. Tenenbaum, Chelsea Finn, and Jiajun Wu (2019). "Reasoning About Physical Interactions with Object-Centric Models". In: (cit. on p. 58).

Jegou, Herve, Matthijs Douze, Cordelia Schmid, and Patrick Perez (2010). "Aggregating local descriptors into a compact image representation". In: *CVPR* (cit. on p. 32).

Ji, S., W. Xu, M. Yang, and K. Yu (2013). "3D Convolutional Neural Networks for Human Action Recognition". In: *IEEE TPAMI* 35.1, pp. 221–231 (cit. on pp. 44, 65).

Ji, Yanli, Feixiang Xu, Yang Yang, Fumin Shen, Heng Tao Shen, and Wei-Shi Zheng (2018). "A large-scale rgb-d database for arbitrary-view human action recognition". In: *Proceedings of the 26th ACM international conference on Multimedia*, pp. 1510–1518 (cit. on p. 46).

Johansson, Fredrik D., Uri Shalit, and David Sontag (2016). "Learning Representations for Counterfactual Inference". In: *ICML* (cit. on p. 6).

Johansson, G. (1973). "Visual perception of biological motion and a model for its analysis". In: *Perception and Psychophysics* (cit. on pp. 3, 4, 40).

Johnson, J., B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. Zitnick, and Girshick R. (2017). "Inferring and executing programs for visual reasoning". In: *ICCV* (cit. on p. 58).

Jolliffe, I. T. (2005). "Principal Component Analysis and Factor Analysis". In: *Springer Series in Statistics* (cit. on p. 17).

Jonides, John (1983). "Further toward a model of the mind's eye's movement". In: *Bulletin of the Psychonomic Society* 21.4, pp. 247–250 (cit. on p. 4).

Jordan, Michael I. (1986). "Attractor dynamics and parallelism in a connectionist sequential machine". In: *Cogitive Science Conference* (cit. on p. 16).

Kahneman, D. (2011). "Thinking, Fast and Slow". In: *Farra, Straus and Giroux* (cit. on p. 21).

Kalogeiton, Vicky, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid (2017). "Action tubelet detector for spatio-temporal action localization". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4405–4413 (cit. on p. 143).

Kanazawa, Angjoo, Michael J. Black, David W. Jacobs, and Jitendra Malik (June 2018). "End-to-End Recovery of Human Shape and Pose". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 40).

Karpathy, A., G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei (2014). "Large-scale Video Classification with Convolutional Neural Networks". In: (cit. on pp. 44, 102).

Kay, W., J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman (2017). "The Kinetics human action video dataset". In: *arXiv abs/1705.06950* (cit. on pp. 3, 44, 141, 143).

Ke, Qiuhong, Mohammed Bennamoun, Senjian An, Ferdous Sohel, and Farid Boussaid (July 2017). "A new representation of skeleton sequences for 3d action recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3288–3297 (cit. on pp. 46, 67, 74, 95).

Kim, Jin-Hwa, Kyoung Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang (2017a). "Hadamard Product for Low-rank Bilinear Pooling". In: *iclr* (cit. on p. 51).

Kim, Junkyung, Matthew Ricci, and Thomas Serre (2018). *Not-So-CLEVR: Visual Relations Strain Feedforward Neural Networks*. URL: https://openreview.net/forum?id=HymuJz-A- (cit. on p. 103).

Kim, Y., C. Denton, L. Hoang, and A.M. Rush (2017b). "Structured attention networks". In: *ICLR* (cit. on p. 47).

Kingma, Diederik and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *ICML*. Vol. abs/1412.6980. URL: http://arxiv.org/abs/1412.6980 (cit. on pp. 73, 93, 109, 130).

Kingma, Dirk and Max Welling (2014). "Auto-Encoding Variational Bayes". In: *ICLR* (cit. on pp. 19, 20).

Kipf, Thomas N. and Max Welling (2017). "Semi-Supervised Classification with Graph Convolutional Networks". In: *ICLR* (cit. on pp. 57, 58, 126).

Kirillov, Alexander, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar (2018). "Panoptic Segmentation". In: *CVPR* (cit. on p. 36).

Kläser, A., M. Marszalek, and C. Schmid (2008). "A spatio-temporal descriptor based on 3D-gradients". In: *BMVC* (cit. on p. 41).

Kocaoglu, Murat, Christopher Snyder, Alexandros G. Dimakis, and Sriram Vishwanath (2018). "CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training". In: *ICLR* (cit. on pp. 22, 122).

Koenderink, Jan J and Andrea J van Doorn (1987). "Representation of local geometry in the visual system". In: *Biological cybernetics* (cit. on p. 30).

Krishna, Ranjay, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalanditis, Li-Jia Li, David A. Shamma, Michael Bernstein, and Li Fei-Fei (2017). "Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations". In: *International Journal of Computer Vision (IJCV)* 123.1 (1), 32—73 (cit. on pp. 56, 103).

Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012). "Imagenet classification with deep convolutional neural networks". In: *NIPS*, pp. 1097–1105 (cit. on pp. 21, 32).

Kubricht, James R., Keith J. Holyoak, and Hongjing Lu (2017). "Intuitive Physics: Current Research and Controversies". In: *Trends in Cognitive Sciences* 21.10, pp. 749–759 (cit. on pp. 55, 121, 122, 130).

Kuehne, H., H. Jhuang, E. Garrote, T. Poggio, and T. Serre (2011). "HMDB: a large video database for human motion recognition". In: *Proceedings of the International Conference on Computer Vision (ICCV)* (cit. on p. 44).

Kuen, J., Z. Wang, and G. Wang (2015). "Recurrent Attentional Networks for Saliency Detection". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3668–3677 (cit. on p. 50).

Lake, Brenden M., Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman (2017). "Building Machines That Learn and Think Like People". In: *Behavioral and Brain Sciences*, 1–101 (cit. on pp. 22, 122).

Lan, Zhenzhong, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut (2020). "ALBERT: A Lite BERT for Self-supervised Learning of Language Representations". In: *International Conference on Learning Representations*. URL: https://openreview.net/forum?id=H1eA7AEtvS (cit. on p. 54).

Lang, K.J. and G.E. Hinton (1988). "A time delay neural network architecture for speech recognition". In: *Technical Report CMU* (cit. on p. 14).

Laptev, I. and T. Lindeberg (2003). "On space-time interest points". In: *ICCV* (cit. on pp. 40, 41).

Laptev, I., M. Marszalek, C. Schmid, and B. Rozenfeld (2005). "Learning realistic human actions from movies". In: *IJCV* (cit. on p. 42).

Laptev, I. and P. Perez (2007). "Retrieving actions in movies". In: *ICCV* (cit. on p. 41).

Laptev, Ivan (2005). "On space-time interest points". In: *International journal of computer vision* 64.2-3, pp. 107–123 (cit. on pp. 41, 42).

Laptev, Ivan (2013). "Modeling and visual recognition of human actions and interactions". In: *Habilitation à diriger des recherches Ecole Normale Suprieure* (cit. on p. 4).

Larochelle, H. and G. Hinton (2010). "Learning to combine foveal glimpses with a third-order Boltzmann machine". In: *NIPS*, pp. 1243–1251 (cit. on pp. 4, 47, 87).

Lazebnik, S., C. Schmid, and J. Ponce (2006). "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories". In: *CVPR* (cit. on pp. 32, 33, 42).

LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel (1989). "Backpropagation applied to handwritten zip code recognition". In: *Neural Computation* 1.4, pp. 541–551 (cit. on p. 15).

LeCun, Y., L. Bottou, and Y. Bengio (1997). "Reading checks with graph transformer networks". In: *ICASSP* (cit. on pp. 2, 12, 14, 15).

LeCun, Y., L. Bottou, G. Orr, and K. Muller (1998). "Efficient backprop". In: *Neural Networks: Tricks of the trade* (cit. on p. 15).

Lee, H. and Z. Chen (1985). "Determination of 3D Human Body Postures from a Single View". In: *COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING* (cit. on p. 38).

Lee, I., D. Kim, S. Kang, and S. Lee (Oct. 2017). "Ensemble Deep Learning for Skeleton-Based Action Recognition Using Temporal Sliding LSTM Networks". In: *ICCV* (cit. on pp. 94, 95).

Lerer, Adam, Sam Gross, and Rob Fergus (2016). "Learning Physical Intuition of Block Towers by Example". In: *ICML* (cit. on pp. 55, 56).

Leung, Thomas and Jitendra Malik (2001). "Representing and recognizing the visual appearance of materials using three-dimensional textons". In: *IJCV* (cit. on p. 29).

Lezama, J., K. Alahari, J. Sivic, and I. Laptev (2011). "Track to the future: Spatiotemporal video segmentation with long-range motion cues". In: *CVPR* (cit. on p. 43).

Li, B, O Camps, and M Sznaier (2012a). "Cross-view activity recognition using hankelets". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 94).

Li, Linjie, Zhe Gan, Yu Cheng, and Jingjing Liu (Oct. 2019). "Relation-Aware Graph Attention Network for Visual Question Answering". In: *The IEEE International Conference on Computer Vision (ICCV)* (cit. on p. 60).

Li, R and T Zickler (2012b). "Discriminative virtual views for cross- view action recognition". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 94).

Li, Z., K. Gavrilyuk, E. Gavves, M. Jain, and C.G.M. Snoek (2017). "VideoLSTM Convolves, Attends and Flows for Action Recognition". In: *CVIU* (cit. on p. 83).

Likas, Aristidis, Nikos Vlassis, and Jakob J Verbeek (2003). "The global k-means clustering algorithm". In: *Pattern recognition* 36.2, pp. 451–461 (cit. on p. 18).

Lin, Ji, Chuang Gan, and Song Han (2019). "TSM: Temporal Shift Module for Efficient Video Understanding". In: *Proceedings of the IEEE International Conference on Computer Vision* (cit. on pp. 45, 143).

Lin, Liang, Keze Wang, Wangmeng Zuo, Meng Wang, Jiebo Luo, and Lei Zhang (2015). "A Deep Structured Model with Radius-Margin Bound for 3D Human Activity Recognition". In: *IJCV*. URL: http://arxiv.org/abs/1512.01642 (cit. on p. 75).

Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár (2017). "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988 (cit. on p. 36).

Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick (2014). "Microsoft COCO: Common Objects in Context". In: *eccv*. URL: /se3/wp-content/uploads/2014/09/coco_eccv.pdf,http://mscoco.org (cit. on pp. 36, 56).

Liu, J., A. Shahroudy, D. Xu, and G. Wang (2016a). "Spatio-Temporal LSTM with Trust Gates for 3D Human Action Recognition". In: *ECCV*, pp. 816–833 (cit. on pp. 67, 71, 74, 75, 95).

Liu, J., G. Wang, P. Hu, L-Y. Duan, and A. Kot (July 2017a). "Global Context-Aware Attention LSTM Networks for 3D Action Recognition". In: *CVPR* (cit. on pp. 74, 95).

Liu, Jun, Amir Shahroudy, Dong Xu, Alex C Kot, and Gang Wang (2017b). "Skeleton-based action recognition using spatio-temporal lstm network with trust gates". In: *IEEE transactions on pattern analysis and machine intelligence* 40.12, pp. 3007–3021 (cit. on pp. 4, 46, 64).

Liu, Li, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen (1809). "Deep learning for generic object detection: A survey". In: *International Journal of Computer Vision*, pp. 1–58 (cit. on p. 35).

Liu, M., H. Liu, and C. Chen (2017c). "Enhanced skeleton visualization for view invariant human action recognition". In: *Pattern Recognition* 68.Supplement C, pp. 346–362 (cit. on pp. 74, 94, 95).

Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg (2016b). "SSD: Single Shot MultiBox Detector". In: *ECCV* (cit. on p. 36).

Lopez-Paz, David (2016). "From dependence to causation". In: *arXiv preprint arXiv:1607.03300* (cit. on pp. 24, 26).

Lopez-Paz, David, Robert Nishihara, Soumith Chintala, Bernhard Scholkopf, and Leon Bottou (2017a). "Discovering Causal Signals in Images". In: *CVPR* (cit. on pp. 22, 27, 122).

Lopez-Paz, David and Maxime Oquab (2017b). "Revisiting Classifier Two-Sample Tests for GAN Evaluation and Causal Discovery". In: *ICLR* (cit. on pp. 22, 122).

Lowe, David G (1999). "Object recognition from local scale-invariant features". In: *ICCV* (cit. on p. 30).

Lowe, David G (2004). "Distinctive Image Features from Scale-Invariant Keypoints". In: *IJCV* (cit. on pp. 30, 31).

Lu, Cewu, Ranjay Krishna, Michael Bernstein, and Li Fei-Fei (2016). "Visual Relationship Detection with Language Priors". In: *European Conference on Computer Vision* (cit. on pp. 56, 57).

Luc, Pauline, Natalia Neverova, Camille Couprie, Jakob Verbeek, and Yann LeCun (2017). "Predicting Deeper Into the Future of Semantic Segmentation". In: *ICCV* (cit. on pp. 60, 103).

Luo, J., W. Wang, and H. Qi (2013). "Group sparsity and geometry constrained dictionary learning for action recognition from depth maps." In: *ICCV* (cit. on p. 75).

Luong, Thang, Hieu Pham, and Christopher D. Manning (Sept. 2015). "Effective Approaches to Attention-based Neural Machine Translation". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1412–1421. URL: https://www.aclweb.org/anthology/D15-1166 (cit. on pp. 49, 50).

Luvizon, D., D. Picard, and H. Tabia (June 2018). "2D/3D Pose Estimation and Action Recognition using Multitask Deep Learning". In: *CVPR* (cit. on pp. 47, 102).

Luvizon, Diogo, David Picard, and Hedi Tabia (2020). "Multi-task Deep Learning for Real-Time 3D Human Pose Estimation and Action Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (cit. on p. 46).

Luvizon, Diogo C, Hedi Tabia, and David Picard (2019). "Human pose regression by combining indirect part detection and contextual information". In: *Computers & Graphics* 85, pp. 15–22 (cit. on p. 38).

Ma, Siwei, Xinfeng Zhang, Chuanmin Jia, Zhenghui Zhao, Shiqi Wang, and Shanshe Wanga (2019). "Image and video compression with neural networks: A review". In: *IEEE Transactions on Circuits and Systems for Video Technology* (cit. on p. 143).

Ma, W. and B. Manjunath (1999). "NeTra: A toolbox for navigating large image databases". In: *Multimedia Systems* (cit. on p. 31).

Malinowski, Mateusz, Carl Doersch, Adam Santoro, and Peter Battaglia (Sept. 2018). "Learning Visual Question Answering by Bootstrapping Hard Attention". In: *eccv* (cit. on p. 50).

Malinowski, Mateusz, Marcus Rohrbach, and Mario Fritz (Jan. 1, 2016). "Ask Your Neurons: A Deep Learning Approach to Visual Question Answering". In: *arXiv:1605.02697*. published (cit. on p. 5).

Manjunatha, Varun, Nirat Saini, and Larry S Davis (2019). "Explicit bias discovery in visual question answering models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9562–9571 (cit. on p. 144).

Mao, J., C. Gan, P. Kohli, J. Tenenbaum, and J. Wu (2019). "The neuro-symbolic concept learner: Interpreting scenes, words, and sentences from natural supervision". In: *ICLR* (cit. on p. 58).

Marino, Kenneth, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi (2019). "Ok-vqa: A visual question answering benchmark requiring external knowledge". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3195–3204 (cit. on p. 144).

Marr, David (1982). "Vision: A Computational Investigation into the Human Representation and Processing of Visual Information". In: *MIT press* (cit. on p. 28).

Martin-Ordas, G., J. Call, and F. Colmenares (2008). "Tubes, tables and traps: great apes solve twofunctionally equivalent trap tasks but show no evidence of transfer across tasks". In: *Animal cognition* 11(3), pp. 432–430 (cit. on pp. 6, 118).

Mathieu, Michaël, Camille Couprie, and Yann LeCun (2016). "Deep multi-scale video prediction beyond mean square error". In: *ICLR* (cit. on p. 120).

Matikainen, P., M. Hebert, and R. Sukthankar (2009). "Trajectons: Action recognition through the motion analysis of tracked features". In: *ICCV workshop on Video-oriented Object and Event Classification* (cit. on p. 42).

McClooskey, M., A. Caramazza, and B. Green (1980). "Curvilinear motion in the absence of external forces: Naive beliefs about the motion of objects". In:

Journal of Experimental Psychoology: Learning, Memory, and Cognition 210.4474, pp. 1139–1141 (cit. on pp. 55, 121).

McClooskey, M., A. Washburn, and L. Felch (1983). “Intuitive physics: the straight-down belief and its origin”. In: *Journal of Experimental Psychoology: Learning, Memory, and Cognition* 9.4, p. 636 (cit. on pp. 55, 121).

McCloskey, M. and D. Kohl (1983). “Naive physics: The curvilinear impetus principle and its role in interactionos with moving objects”. In: *Journal of Experimental Psychoology: Learning, Memory, and Cognition* 9.1, p. 146 (cit. on pp. 55, 121).

McLachlan, Geoffrey J. and Thriyambakam Krishnan (2008). *The EM algorithm and extensions*. Wiley (cit. on p. 17).

Mehta, Dushyant, Helge Rhodin, Dan Casas, Pascal Fua, Oleksandr Sotnychenko, Weipeng Xu, and Christian Theobalt (2017). “Monocular 3d human pose estimation in the wild using improved cnn supervision”. In: *2017 International Conference on 3D Vision (3DV)*. IEEE, pp. 506–516 (cit. on p. 40).

Michotte, A. (1963). “The Perception of Causality”. In: *Basic Books* (cit. on p. 122).

Miech, Antoine, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman (2019a). “End-to-End Learning of Visual Representations from Uncurated Instructional Videos”. In: *arXiv preprint arXiv:1912.06430* (cit. on p. 142).

Miech, Antoine, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic (2019b). “Howto100M: Learning a text-video embedding by watching hundred million narrated video clips”. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2630–2640 (cit. on p. 142).

Mikolajczyk, Krystian and Cordelia Schmid (2002). “An affine invariant interest point detector”. In: *ECCV* (cit. on p. 42).

Mikolov, T. and G. Zweig (2016). “Context dependent recurrent neural network language model”. In: *Spoken Language Technology Workshop* (cit. on p. 69).

Misra, Ishan, C. Lawrence Zitnick, and Martial Hebert (2016). “Shuffle and Learn: Unsupervised Learning using Temporal Order Verification”. In: *ECCV* (cit. on p. 142).

Mnih, V., N. Heess, A. Graves, and K. Kavukcuoglu (2014). “Recurrent Models of Visual Attention”. In: *Neural Information Processing Systems (NIPS)*, pp. 2204–2212 (cit. on pp. 50, 51, 64, 68).

Molchanov, P., X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz (2016). “Online Detection and Classification of Dynamic Hand Gestures With Recurrent 3D Convolutional Neural Network”. In: *CVPR*, pp. 4207–4215 (cit. on pp. 65, 66).

Mooij, Joris M., Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf (2016). “Distinguishing Cause from Effect Using Observational Data: Methods and Benchmarks”. In: *JMLR* (cit. on pp. 22, 122).

Moore, Andrew (2001). *K-means and Hierarchical Clustering* (cit. on p. 18).

Mori, Greg, Xiaofeng Ren, Alexei A. Efros, and Jitendra Malik (2004). “Recovering Human Body Configurations: Combining Segmentation and Recognition”. In: *CVPR* (cit. on p. 38).

Neverova, N., C. Wolf, G.W. Taylor, and F. Nebout (2016). “ModDrop: adaptive multi-modal gesture recognition”. In: *IEEE TPAMI* 38.8, pp. 1692–1706 (cit. on pp. 47, 65, 66, 71).

Neverova, Natalia, Christian Wolf, Florian Nebout, and Graham W Taylor (2017). “Hand pose estimation through semi-supervised and weakly-supervised learning”. In: *Computer Vision and Image Understanding* 164, pp. 56–67 (cit. on p. 39).

Newell, Alejandro, Kaiyu Yang, and Jia Deng (2016). “Stacked hourglass networks for human pose estimation”. In: *ECCV* (cit. on p. 38).

Ng, Andrew et al. (2011). “Sparse autoencoder”. In: *CS294A Lecture notes* 72.2011, pp. 1–19 (cit. on pp. 19, 20).

Ng, Joe Yue-Hei, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici (2015). “Beyond Short Snippets: Deep Networks for Video Classification”. In: *Computer Vision and Pattern Recognition* (cit. on p. 44).

Novotny, David, Samuel Albanie, Diane Larlus, and Andrea Vedaldi (2018a). “Self-supervised learning of geometrically stable features through probabilistic introspection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3637–3645 (cit. on p. 20).

Novotny, David, Diane Larlus, and Andrea Vedaldi (2018b). “Capturing the geometry of object categories from video supervision”. In: *IEEE transactions on pattern analysis and machine intelligence* (cit. on p. 20).

Noyes, Dan (2015). “The top 20 valuable Facebook statistics”. In: *Zephoria, Florida, Available from: at https://zephoria. Com/social-media/top-15-valuable-facebookstatistics/[Accessed 10 February 2015]* (cit. on p. 1).

Oord, Aaron van den, Yazhe Li, and Oriol Vinyals (2018). “Representation learning with contrastive predictive coding”. In: *arXiv preprint arXiv:1807.03748* (cit. on p. 21).

O’Rourke, Joseph and N. I. Badler (1979). “Model-based image analysis of human motion using constraint propagation”. In: *IEE TPAMI* (cit. on p. 38).

Parameswaran, V. and R. Chellappa (2006). “View invariance for human action recognition”. In: *IJCV* (cit. on p. 40).

Pearl, Judea (2009). *Causality: Models, Reasoning and Inference*. 2nd. New York, NY, USA: Cambridge University Press (cit. on p. 120).

Pearl, Judea (2012). “The do-calculus revisited”. In: *arXiv preprint arXiv:1210.4852* (cit. on p. 25).

Pearl, Judea (2018). “Theoretical impediments to machine learning with seven sparks from the causal revolution”. In: *arXiv preprint arXiv:1801.04016* (cit. on pp. 22, 26, 27, 122).

Pearl, Judea and Dana McKenzie (2018). *The Book of Why: The New Science of Cause and Effect*. Basic Books (cit. on pp. 26, 120).

Peng, Xiaojiang and Cordelia Schmid (2016). "Multi-region two-stream R-CNN for action detection". In: *European conference on computer vision*. Springer, pp. 744–759 (cit. on p. 143).

Perez, E., F. Strub, H. de Vries, V. Dumoulin, and A. Courville (2018). "Film: Visual reasoning with a general conditioning layer". In: *AAAI* (cit. on p. 58).

Perez, E., H. De Vries, F. Strub, V. Dumoulin, and A. Courville (2017). "Learning Visual Reasoning Without Strong Priors". In: *ICML 2017's Machine Learning in Speech and Language Processing Workshop* (cit. on pp. 59, 103).

Perronnin, F., J. Sánchez, and T. Mensink (2010). "Improving the fisher kernel for large-scale image classification". In: *ECCV* (cit. on p. 32).

Peyre, Julia, Ivan Laptev, Cordelia Schmid, and Josef Sivic (2019). "Detecting unseen visual relations using analogies". In: *International Conference on Computer Vision (ICCV)* (cit. on p. 56).

Pickup, Lyndsey C., Zheng Pan, Donglai Wei, YiChang Shih, Changshui Zhang, Andrew Zisserman, Bernhard Scholkopf, and William T. Freeman (2014). "Seeing the Arrow of Time". In: *CVPR* (cit. on pp. 6, 101, 104, 142).

Pinto, Lerrel, Dhiraj Gandhi, Yuanfeng Han, Yong-Lae Park, and Abhinav Gupta (2016). "The curious robot: Learning visual representations via physical interactions". In: *European Conference on Computer Vision*. Springer, pp. 3–18 (cit. on p. 144).

Pishchulin, Leonid, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V Gehler, and Bernt Schiele (2016). "Deepcut: Joint subset partition and labeling for multi person pose estimation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4929–4937 (cit. on p. 39).

Polana, R. and R. Nelson (1994). "Low level recognition of human motion". In: *In IEEE Workshop on Nonrigid and Articulate Motion* (cit. on p. 41).

Rahmani, H and A Mian (2015). "Learning a non-linear knowledge transfer model for cross-view action recognition". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 94).

Rahmani, H and A Mian (2016). "3d action recognition from novel viewpoints". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 94).

Ramachandran, P., N. Parma, A. Vaswani, I. Bello, A Levskaya, and J. Shlens (2019). "Stand-Alone Self-Attention in Vision Models". In: *NIPS* (cit. on p. 54).

Ramanan, D., D.A. Forsyth, and A. Zisserman (2007). "Tracking people by learning their appearance". In: *IEEE T-PAMI* (cit. on pp. 40, 41).

Ramanan, Deva (2006). "Learning to parse images of articulated bodies". In: *NIPS* (cit. on p. 38).

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi (2016). "You Only Look Once: Unified, Real-Time Object Detection". In: *CVPR* (cit. on p. 36).

Redmon, Joseph and Ali Farhadi (2017). "YOLO9000: better, faster, stronger". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271 (cit. on pp. 37, 143).

Redmon, Joseph and Ali Farhadi (2018). "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.02767* (cit. on p. 37).

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun (2015). "Faster R-CNN: Towards real-time object detection with region proposal networks." In: *NIPS*, pp. 91–99 (cit. on pp. 34, 36, 56, 107).

Reynolds, Douglas A (2009). "Gaussian Mixture Models." In: *Encyclopedia of biometrics* 741 (cit. on p. 18).

Rifai, Salah, Grégoire Mesnil, Pascal Vincent, Xavier Muller, Yoshua Bengio, Yann Dauphin, and Xavier Glorot (2011). "Higher order contractive auto-encoder". In: *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 645–660 (cit. on pp. 19, 20).

Riochet, Ronan, Mario Ynocente Castro, Mathieu Bernard, Adam Lerer, Rob Fergus, Véronique Izard, and Emmanuel Dupoux (2018). "IntPhys: A Framework and Benchmark for Visual Intuitive Physics Reasoning". In: *CoRR* abs/1803.07616 (cit. on p. 122).

Roberts, Lawrence G. (1963). "Machine Perception of Three-Dimensional Solids". In: *PhD thesis, Massachusetts Institute of Technology* (cit. on p. 28).

Rocktäschel, Tim, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom (2015). "Reasoning about entailment with neural attention". In: *arXiv preprint arXiv:1509.06664* (cit. on p. 49).

Roger, Johansson, Holsanova Jana, Dewhurst Richard, and Holmqvist Kenneth (2012). "Eye movements during scene recollection have a functional role, but they are not reinstatements of those produced during encoding". In: *Journal of experimental psychology. Human perception and performance* (cit. on p. 48).

Rogez, Gregory, Philippe Weinzaepfel, and Cordelia Schmid (2019). "LCR-Net++: Multi-person 2D and 3D Pose Detection in Natural Images". In: *IEEE TPAMI* (cit. on p. 40).

Rojas-Carulla, M., M. Baroni, and D. Lopez-Paz (2018). "Causal Discovery Using Proxy Variables". In: *ICLR Workshop* (cit. on pp. 22, 122).

Rosenblatt, F. (1957). "The perceptron: A perceiving and recognizing automaton". In: *Project PARA, Cornell Aeronautical Lab* (cit. on pp. 14, 16).

Rubin, Donald B (1986). "Statistics and causal inference: Comment: Which ifs have causal answers". In: *Journal of the American Statistical Association* 81.396, pp. 961–962 (cit. on p. 27).

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). "Learning representations by back-propagating errors". In: *Nature* (cit. on pp. 12, 14).

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei (2015). "ImageNet large scale visual recognition challenge". In: *IJCV* 115.3, pp. 211–252 (cit. on pp. 3, 32, 72, 102, 108).

Rza Alp Guler Natalia Neverova, Iasonas Kokkinos (2018). "DensePose: Dense Human Pose Estimation In The Wild". In: (cit. on p. 40).

Sadeghi, Mohammad Amin and Ali Farhadi (2011). "Recognition using visual phrases". In: *CVPR* (cit. on p. 56).

Salton, Gerard and Michael J. McGill (1986). "Introduction to Modern Information Retrieval". In: *McGraw-Hill and Inc* (cit. on p. 31).

Sand, P. and S. Teller (2008). "Particle video: Long-range motion estimation using point trajectories". In: *IJCV* (cit. on p. 43).

Sande, K. van de, J.R.R. Uijlings, T. Gevers, and A.W.M. Smeulders (2011). "Segmentation as Selective Search for Object Recognition". In: *ICCV* (cit. on pp. 35, 36).

Santoro, Adam, Felix Hill, David G. T. Barrett, Ari S. Morcos, and Timothy P. Lillicrap (2018). "Measuring abstract reasoning in neural networks". In: *ICML* (cit. on p. 59).

Santoro, Adam, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Tim Lillicrap (2017). "A simple neural network module for relational reasoning". In: *NIPS*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Curran Associates, Inc., pp. 4967–4976. URL: http://papers.nips.cc/paper/7082-a-simple-neural-network-module-for-relational-reasoning.pdf (cit. on pp. 58, 59, 103, 105, 106, 112).

Schiele, Bernt and James L Crowley (1996). "Object recognition using multidimensional receptive field histograms". In: *ECCV* (cit. on p. 29).

Schmid, Cordelia and Roger Mohr (1997). "Local grayvalue invariants for image retrieval". In: *IEEE PAMI* (cit. on p. 30).

Schölkopf, B., D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. Mooij (2012). "On Causal and Anticausal Learning". In: *ICML* (cit. on pp. 22, 122).

Scholkopf, Bernhard (2019). "Causality for Machine Learning". In: *arXiv preprint arXiv:1911.10500* (cit. on p. 22).

Scovanner, P., S. Ali, and M. Shah (2007). "A 3-dimensional SIFT descriptor and its application to action recognition". In: *ACM* (cit. on p. 41).

Seymour, A. (1966). "The Summer Vision Project". In: *MIT Library* (cit. on p. 28).

Shahroudy, A., J Liu, T.-T. Ng, and G. Wang (2016a). "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis". In: *CVPR*, pp. 1010–1019 (cit. on pp. 46, 64, 66, 73, 74, 82, 92, 93, 95, 102).

Shahroudy, A., T.-T. Ng, Y. Gong, and G. Wang (2016b). "Deep Multimodal Feature Analysis for Action Recognition in RGB+D Videos." In: *arXiv* (cit. on pp. 74, 75, 95).

Sharma, S., R. Kiros, and R. Salakhutdinov (2016). "Action Recognition Using Visual Attention". In: *ICLR Workshop* (cit. on pp. 4, 50–52, 64, 65, 69, 83, 103).

Shen, Tao, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and Chengqi Zhang (2018). "Disan: Directional self-attention network for rnn/cnn-free language understanding". In: *Thirty-Second AAAI Conference on Artificial Intelligence* (cit. on p. 49).

Shi, Lei, Yifan Zhang, Jian Cheng, and Hanqing Lu (2019). "Two-stream adaptive graph convolutional networks for skeleton-based action recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 12026–12035 (cit. on p. 61).

Shpitser, I. and J Pearl (2009). "Effects of Treatment on the Treated: Identification and Generalization". In: *UAI* (cit. on p. 26).

Shrestha, Robik, Kushal Kafle, and Christopher Kanan (2019). "Answer them all! toward universal visual question answering models". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10472–10481 (cit. on p. 144).

Simonyan, Karen and Andrew Zisserman (2014). "Two-Stream Convolutional Networks for Action Recognition in Videos". In: *NIPS*, pp. 568–576 (cit. on pp. 44, 45, 102).

Simonyan, Karen and Andrew Zisserman (2015). "Very deep convolutional networks for largescale image recognition". In: *ICLR* (cit. on p. 33).

Sirovich, Lawrence and Michael Kirby (1987). "Low-dimensional procedure for the characterization of human faces". In: *Journal of the Optical Society of America* (cit. on p. 28).

Sivic, Josef, and Andrew Zisserman (2003). "Video Google: A Text Retrieval Approach to Object Matching in Videos". In: *ICCV* (cit. on p. 32).

Smeulders, Arnold WM, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain (2000). "Content-based image retrieval at the end of the early years". In: *IEEE Transactions on pattern analysis and machine intelligence* 22.12, pp. 1349–1380 (cit. on p. 2).

Song, S., C. Lan, J. Xing, W. Zeng, and J. Liu (2016). "An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data". In: *AAAI Conf. on AI* (cit. on pp. 4, 50, 51, 71, 74, 75, 95, 103).

Soomro, Khurram, Amir Roshan Zamir, Mubarak Shah, Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah (2012). "UCF101: A dataset of 101 human actions classes from videos in the wild". In: *CoRR* (cit. on p. 44).

Spirtes, Peter (2010). "Introduction to Causal Inference". In: *JMLR* (cit. on p. 120).

Stabinger, Sebastian, Antonio Rodríguez-Sánchez, and Justus Piater (2016). "25 years of CNNs: Can we compare to human abstraction capabilities?" In: *ICANN* (cit. on p. 100).

Steenkiste, Sjoerd van, Michael Chang, Klaus Greff, and Jürgen Schmidhuber (2018). "Relational Neural Expectation Maximization: Unsupervised Discovery of Objects and their Interactions". In: *ICLR* (cit. on pp. 58, 103).

Sukhbaatar, S., A. Szlam, J. Weston, and R. Fergus (2015). "End-To-End Memory Networks". In: *NIPS*, pp. 2440–2448 (cit. on pp. 51, 52).

Sun, Chen, Fabien Baradel, Kevin Murphy, and Cordelia Schmid (2019). "Learning Video Representations using Contrastive Bidirectional Transformer". In: *arXiv preprint arXiv:1906.05743* (cit. on p. 8).

Sun, Chen, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid (Sept. 2018). "Actor-centric Relation Network". In: *The European Conference on Computer Vision (ECCV)* (cit. on pp. 6, 47, 61).

Sun, L., K. Jia, K. Chen, D.Y. Yeung, B.E. Shi, and S. Savarese (2017). "Lattice Long Short-Term Memory for Human Action Recognition". In: *ICCV* (cit. on p. 82).

Swain, Michael J and Dana H Ballard (1991). "Color indexing". In: *IJCV* (cit. on p. 29).

Szegedy, C., V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna (2016). "Rethinking the Inception Architecture for Computer Vision". In: *CVPR*, pp. 2818–2826 (cit. on p. 72).

Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich (2015). "Going deeper with convolutions". In: *CVPR* (cit. on p. 33).

Szeliski, Richard (2010). "Computer Vision: Algorithms and Applications". In: *Springer* (cit. on p. 28).

Tao, Lingling and Rene Vidal (2015). "Moving Poselets: A Discriminative and Interpretable Skeletal Motion Representation for Action Recognition." In: *ICCV Workshops*, pp. 303–311 (cit. on p. 75).

Taylor, G., W. Fergus, Y. LeCun, and C. Bergler (2010). "Convolutional learning of spatio-temporal features". In: *ECCV* (cit. on p. 44).

T.Cover and P. Hart (1967). "Nearest Neighbor Pattern Classification". In: *IEEE Transactions on Information Theory* (cit. on p. 32).

Teney, Damien, Lingqiao Liu, and Anton van den Hengel (July 2017). "Graph-Structured Representations for Visual Question Answering". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 5, 59).

Tian, J. and Judea Pearl (2002). "A General identification condition for causal effects". In: *AAAI* (cit. on pp. 6, 26).

Tommasi, Tatiana, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars (2017). "A deeper look at dataset bias". In: *Domain adaptation in computer vision applications*. Springer, pp. 37–55 (cit. on p. 142).

Torralba, Antonio and Alexei A Efros (2011). "Unbiased look at dataset bias". In: *CVPR 2011*. IEEE, pp. 1521–1528 (cit. on p. 142).

Toshev, A. and C. Szegedy. (2014). "Deeppose: Human pose estimation via deep neural networks". In: *CVPR* (cit. on p. 38).

Tran, Du, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri (2015). "Learning Spatiotemporal Features With 3D Convolutional Networks". In: *ICCV* (cit. on pp. 44, 74, 102).

Tran, Du, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri (2018). "A Closer Look at Spatiotemporal Convolutions for Action Recognition". In: *CVPR* (cit. on pp. 44, 45).

Tsai, Yao-Hung Hubert, Santosh Divvala, Louis-Philippe Morency, Ruslan Salakhutdinov, and Ali Farhadi (June 2019). "Video Relationship Reasoning Using Gated Spatio-Temporal Energy Graph". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 61).

Turk, Matthew and Alex Pentland (1991). "Eigenfaces for recognition". In: *Journal of cognitive neuroscience* (cit. on p. 29).

Varol, Gül, Ivan Laptev, and Cordelia Schmid (2018). "Long-term Temporal Convolutions for Action Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (cit. on p. 44).

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2017). "Attention is All You Need". In: URL: `https://arxiv.org/pdf/1706.03762.pdf` (cit. on pp. 20, 52, 54).

Veličković, Petar, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio (2018). "Graph Attention Networks". In: *ICLR* (cit. on pp. 58, 103).

Vemulapalli, R., F. Arrate, and R. Chellappa (2014). "Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group". In: *CVPR*, pp. 588–595 (cit. on pp. 94, 95).

Vincent, P., H. Larochelle, Y. Bengio, and P.-A. Manzagol (2008). "Extracting and composing robust features with denoising autoencoders". In: *International Conference on Machine Learning* (cit. on pp. 19, 20).

Vincent, Pascal, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, and Pierre-Antoine Manzagol (2010). "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion". In: *Journal of machine learning research* 11.Dec, pp. 3371–3408 (cit. on p. 20).

Von Luxburg, Ulrike (2007). "A tutorial on spectral clustering". In: *Statistics and computing* 17.4, pp. 395–416 (cit. on p. 18).

Wang, H., A. Kläser, C. Schmid, and C.-L. Liu (2011). "Action Recognition by Dense Trajectories". In: *CVPR* (cit. on p. 102).

Wang, H., A. Kläser, and C. Schmid (2013a). "Dense trajectories and motion boundary descriptors for action recognition". In: *IJCV* (cit. on pp. 3, 40, 42–44).

Wang, H., D. Oneata, J. Verbeek, and C. Schmid (2015). "A robust and efficient video representation for action recognition." In: *IJCV* (cit. on p. 43).

Wang, H. and C. Schmid (2013b). "Action recognition with improved trajectories". In: *ICCV* (cit. on p. 42).

Wang, Jiang, Zicheng Liu, Ying Wu, and Junsong Yuan (2012). "Mining actionlet ensemble for action recognition with depth cameras". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1290–1297 (cit. on pp. 74, 75).

Wang, Jiang, Nie Xiaohan, Xia Yin, Wu Ying, and Zhu Song-Chun (2014). "Cross-View Action Modeling, Learning, and Recognition". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on pp. 92, 94).

Wang, Limin, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool (2016a). "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition". In: *ECCV* (cit. on p. 44).

Wang, Pichao, Wanqing Li, Chuankun Li, and Yonghong Hou (2016b). "Action Recognition Based on Joint Trajectory Maps with Convolutional Neural Networks". In: *ACM Conference on Multimedia* (cit. on pp. 67, 74, 95).

Wang, Xiaolong, Ross Girshick, Abhinav Gupta, and Kaiming He (2018a). "Non-local Neural Networks". In: *CVPR* (cit. on pp. 4, 45, 54).

Wang, Xiaolong and Abhinav Gupta (2018b). "Videos as Space-Time Region Graphs". In: *ECCV* (cit. on pp. 5, 47, 61).

Wei, Shih-En, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh (2016). "Convolutional pose machines". In: *CVPR* (cit. on pp. 38, 39).

Weiss, Isaac (1988). "Projective invariants of shapes". In: *CVPR* (cit. on p. 28).

Weston, J., S. Chopra, and A. Bordes (2015). "Memory Networks". In: *ICLR* (cit. on pp. 51, 52).

Willems, G., T. Tuytelaars, and L. Van Gool (2008). "An efficient dense and scale invariant spatio-temporal interest point detector." In: *ECCV* (cit. on p. 42).

Williams, R.J. (2012). "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine Learning* 8.3-4, pp. 229–256 (cit. on p. 50).

Wu, Chao-Yuan, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick (2019a). "Long-Term Feature Banks for Detailed Video Understanding". In: *CVPR* (cit. on p. 45).

Wu, Chao-Yuan, Ross Girshick, Kaiming He, Christoph Feichtenhofer, and Philipp Krähenbühl (2019b). "A Multigrid Method for Efficiently Training Video Models". In: *arXiv preprint arXiv:1912.00998* (cit. on p. 143).

Wu, Chao-Yuan, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl (2018). "Compressed video action recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6026–6035 (cit. on p. 143).

Wu, D., L. Pigou, P.-J. Kindermans, N. Do-Hoang Le, L. Shao, J. Dambre, and J.M. Odobez (2016). "Deep dynamic neural networks for multimodal gesture segmentation and recognition". In: *IEEE TPAMI* 38.8, pp. 1583–1597 (cit. on pp. 65, 66).

Wu, Jiajun, Erika Lu, Pushmeet Kohli, Bill Freeman, and Josh Tenenbaum (2017). "Learning to See Physics via Visual De-animation". In: *NIPS* (cit. on p. 55).

Wu, JIajun, Ilker Yildirim, Joseph J. Lim, Bill Freeman, and Josh Tenenbaum (2015). "Galileo: Perceiving Physical Object Properties by Integrating a Physics Engine with Deep Learning". In: *NIPS* (cit. on pp. 55, 121).

Wu, Yuxin, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick (2019c). *Detectron2* (cit. on p. 2).

Xie, S., C. Sun, J. Huang, Z. Tu, and K. Murphy (2017). "Rethinking Spatiotemporal Feature Learning For Video Understanding". In: *Pre-print: arxiv:1712.04851* (cit. on pp. 44, 45, 102, 108, 143).

Xu, K., J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio (2015). "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention". In: *International Conference in Machine Learning (ICML)*. ICML'15. Lille, France: JMLR.org, pp. 2048–2057. URL: http://dl.acm.org/citation.cfm?id=3045118.3045336 (cit. on p. 50).

Yamato, J., J. Ohya, and K. Ishii (1992). "Recognizing human action in time-sequential images using hidden markov model". In: *CVPR* (cit. on p. 41).

Yan, Sijie, Yuanjun Xiong, and Dahua Lin (2018). "Spatial temporal graph convolutional networks for skeleton-based action recognition". In: *Thirty-second AAAI conference on artificial intelligence* (cit. on pp. 46, 61).

Yang, Y. and D. Ramanan (2011). "Articulated pose estimation with flexible mixtures-ofparts". In: *CVPR* (cit. on p. 38).

Yang, Zhilin, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le (2019). "XLNet: Generalized Autoregressive Pretraining for Language Understanding". In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., pp. 5754–5764. URL: http://papers.nips.cc/paper/8812-xlnet-generalized-autoregressive-pretraining-for-language-understanding.pdf (cit. on p. 54).

Yang, Zichao, Xiaodong He, Jianfeng Gao, Li Deng, and Alexander J. Smola (2016). "Stacked Attention Networks for Image Question Answering". In: *cvpr*, pp. 21–29. URL: http://dx.doi.org/10.1109/CVPR.2016.10 (cit. on p. 51).

Yarbus, Alfred L (1967). *Eye movements and vision*. Springer (cit. on p. 4).

Ye, Tian, Xiaolong Wang, James Davidson, and Abhinav Gupta (2018). "Interpretable Intuitive Physics Model". In: *ECCV* (cit. on pp. 55, 123).

Yeung, S., O. Russakovsky, G. Mori, and L. Fei-Fei (2016). "End-to-end Learning of Action Detection from Frame Glimpses in Videos". In: *CVPR* (cit. on p. 50).

Yeung, Serena, Olga Russakovsky, Ning Jin, Mykhaylo Andriluka, Greg Mori, and Li Fei-Fei (2015). "Every Moment Counts: Dense Detailed Labeling of Actions in Complex Videos". In: *arXiv preprint arXiv:1507.05738* (cit. on pp. 50, 69, 103).

Yilmaz, A. and M. Shah (2005). "Recognizing human actions in videos acquired by uncalibrated moving cameras". In: *ICCV* (cit. on p. 41).

Yin, Wenpeng, Hinrich Schütze, Bing Xiang, and Bowen Zhou (2016). "Abcnn: Attention-based convolutional neural network for modeling sentence pairs". In: *Transactions of the Association for Computational Linguistics* 4, pp. 259–272 (cit. on p. 49).

Yu, Ruichi, Ang Li, Vlad I. Morariu, and Larry S. Davis (Oct. 2017a). "Visual Relationship Detection With Internal and External Linguistic Knowledge Distillation". In: *iccv* (cit. on p. 51).

Yu, Zhou, Jun Yu, Jianping Fan, and Dacheng Tao (2017b). "Multi-modal Factorized Bilinear Pooling with Co-Attention Learning for Visual Question Answering". In: *iccv* (cit. on p. 59).

Yun, Kiwon, Jean Honorio, Debaleena Chattopadhyay, Tamara L. Berg, and Dimitris Samaras (2012). "Two-person interaction detection using body-pose features and multiple instance learning". In: *CVPR Workshop*, pp. 28–35 (cit. on pp. 74, 75).

Yun, Kiwon, Jean Honorio, Debaleena Chattopadhyay, Tamara L. Berg, and Dimitris Samaras (2014). "Interactive body part contrast mining for human interaction recognition." In: *ICMEW* (cit. on p. 75).

Zanfir, Mihai, Marius Leordeanu, and Cristian Sminchisescu (2013). "The Moving Pose: An Efficient 3D Kinematics Descriptor for Low-Latency Action Recognition and Detection". In: *ICCV*, pp. 2752–2759 (cit. on pp. 66, 75).

Zeiler, Matthew D and Rob Fergus (2014). "Visualizing and understanding convolutional networks". In: *ECCV* (cit. on p. 33).

Zhang, P., C. Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng (2017). "View Adaptive Recurrent Neural Networks for High Performance Human Action Recognition From Skeleton Data". In: *ICCV* (cit. on pp. 74, 75, 95).

Zhang, Z, C Wang, B Xiao, W Zhou, S Liu, and C Shi (2013). "Cross-view action recognition via a continuous virtual path". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (cit. on p. 94).

Zheng, David, Vinson Luo, Jiajun Wu, and Joshua B. Tenenbaum (2018). "Unsupervised Learning of Latent Physical Properties Using Perception-Prediction Networks". In: *UAI* (cit. on p. 55).

Zhou, Peng, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu (2016). "Attention-based bidirectional long short-term memory networks for relation classification". In: *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*, pp. 207–212 (cit. on p. 49).

Zhou, Xi, Kai Yu, Tong Zhang, and Thomas Huang (2010). "Super-vector Coding of Local Image Descriptors". In: *ECCV* (cit. on p. 32).

Zhu, W., W. Chen, and G. Guo (2015). "Fusing multiple features fordepth-based action recognition." In: *ACM TIST* (cit. on p. 75).

Zhu, Wentao, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie (2016). "Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks”. In: *Thirtieth AAAI Conference on Artificial Intelligence* (cit. on pp. 46, 75).