

Learning with Privileged Information using Multimodal Data



Nuno Cruz Garcia

DITEN, Università di Genova
PAVIS, Istituto Italiano di Tecnologia
PhD in Science and Technology for Electronic
and Telecommunication Engineering
Curriculum: Computational Vision,
Automatic Recognition and Learning

Supervisor: Vittorio Murino *Co-Supervisor:* Pietro Morerio

In partial fulfillment of the requirements for the degree of
Doctor of Philosophy

November 22, 2019

Acknowledgements

I had the pleasure to share these three years of PhD with a lot of kind, brilliant, and inspiring people.

The first words of deep gratitude is to my supervisor, Vittorio Murino, for the opportunity to learn from him, the guidance, the inspiration, and the mentorship. To my co-supervisor Pietro Morerio, for all the brilliant insights on deep learning, all the good ideas and experiments that make some of my favorite parts in this document. They constitute a research role model that I hope I can be one day to my students.

To Sarah, Vitaly, Stan, and IVC friends, for the amazing six months I spent at Boston University. The positivity, kindness, wisdom, and scientific rigor I learned in all the meetings, lunches, walks, and conversations are lessons that go beyond research.

I'm fortunate to have crossed paths with many people that made my stay in Genova a beautiful experience. PAVIS, VGM, IIT, and Unige are incredible places to be as a young researcher. I'm grateful to Alessio del Bue, Prof. Mario Marchese, and all staff for the helpful, cooperative and lively environment. Thank you to all the friends from IIT football team, to the Portuguese friends, and to those with whom I shared fraternal evenings.

I am most grateful to my family. They are the most inspiring examples I have. They taught me to follow my dreams by example, when they decided to continue to study late in life, during hard times. They taught me to be curious, to have fun, to stay focused, and to like science. To my favorite person in the world, Patrícia, for the unique care, understanding and closeness. To Vanessa, for all the love, patience, and support.

”If you can think - and not make thoughts your aim”

Para a minha família.

Abstract

Computer vision is the science related to teaching machines to see and understand digital images or videos. During the last decade, computer vision has seen tremendous progress on perception tasks such as object detection, semantic segmentation, and video action recognition, which lead to the development and improvements of important industry applications such as self-driving cars and medical image analysis. These advances are mainly due to fast computation offered by GPUs, the development of high capacity models such as deep neural networks, and the availability of large datasets, often composed by a variety of modalities. In this thesis we explore how multimodal data can be used to train deep convolutional neural networks.

Humans perceive the world through multiple senses, and reason over the multimodal space of stimuli to act and understand the environment. One way to improve the perception capabilities of deep learning methods is to use different modalities as input, as it offers different and complementary information about the scene. Recent multimodal datasets for computer vision tasks include modalities such as depth maps, infrared, skeleton coordinates, and others, besides the traditional RGB.

This thesis investigates deep learning systems that learn from multiple visual modalities. In particular, we are interested in a very practical

scenario in which an input modality is missing at test time. The question we address is the following: how can we take advantage of multimodal datasets for training our model, knowing that, at test time, a modality might be missing or too noisy? The case of having access to more information at training time than at test time is referred to as learning using privileged information.

In this work, we develop methods to address this challenge, with special focus on the tasks of action and object recognition, and on the modalities of depth, optical flow, and RGB, that we use for inference at test time. This thesis advances the art of multimodal learning in three different ways. First, we develop a deep learning method for video classification that is trained on RGB and depth data, and is able to hallucinate depth features and predictions at test time. Second, we build on this method and propose a more generic mechanism based on adversarial learning to learn to mimic the predictions originated by the depth modality, and is able to automatically switch from true depth features to generated depth features in case of a noisy sensor. Third, we develop a method that learns a single network trained on RGB data, that is enriched with additional supervision information from other modalities such as depth and optical flow at training time, and that outperforms an ensemble of networks trained independently on these modalities.

Contents

1	Introduction	1
1.1	Objective, Motivation, and Challenges	1
1.2	Contributions and Outline	4
1.2.1	List of Publications	5
2	Related Work	7
2.1	Generalized Distillation	7
2.2	Adversarial Learning	10
2.3	Multimodal Deep Learning	11
2.3.1	RGB-D Vision	11
2.3.2	Ensemble Learning	13
3	Modality Distillation with Multiple Stream Networks for Action Recognition	15
3.1	Introduction	15
3.2	Model	17
3.2.1	Cross-stream multiplier networks	17
3.2.2	Hallucination stream	20
3.2.3	Training Paradigm	23
3.3	Experiments	25

3.3.1	NTU RGB+D Dataset	25
3.3.2	Comparison with state of the art	26
3.3.3	Ablation study	28
3.3.4	Inference with noisy depth	32
3.3.5	Inverting the data modalities: RGB distillation	33
3.3.6	Implementation details	34
3.4	Summary	37
4 Learning with Privileged Information via Adversarial Discriminative Modality Distillation		38
4.1	Introduction	38
4.2	Model	41
4.2.1	Training procedure	43
4.2.2	Architectural details	45
4.3	Experiments	49
4.3.1	Datasets	49
4.3.2	Ablation Study	50
4.3.3	Action recognition performance and comparisons	53
4.3.4	Object recognition performance and comparisons	57
4.3.5	Inference with noisy depth	59
4.3.6	Discussion	62
4.4	Summary	63
5 Distillation Multiple Choice Learning		64
5.1	Introduction	64
5.2	Model	67
5.2.1	Distillation Multiple Choice Learning	68
5.2.2	Relationship to other MCL methods	71

CONTENTS

5.3	Experiments	72
5.3.1	Datasets	73
5.3.2	Architecture and Setup	74
5.3.3	Results	74
5.4	Summary	81
6	Conclusions	85
	References	100

List of Figures

1.1	What is the best way of using all data available at training time, considering a missing (or noisy) modality at test time?	2
3.1	Training procedure described in section 3.2.3 (see also text therein). The 1 st step represents the segregate training of the appearance and depth stream networks. The 2nd step illustrates the two-stream joint training. The 3 rd step refers to the hallucination learning step using the soft labels with temperature s_i (eq. 3.6) and the novel distillation loss L (eq. 3.7), where the weights of the depth stream network are frozen. The 4 th step refers to a fine-tuning step, and exemplifies also the testing setup, in which RGB data is the only input to the model.	18
3.2	Detail of the ResNet residual unit, showing the multiplicative connections and temporal convolutions [1]. In our architecture, the signal injection occurs before the 2 nd residual unit of each of the four ResNet blocks.	20
3.3	Example of RGB and depth frames from the NTU RGB+D Dataset.	26

3.4	The plot shows the hallucination loss L_{hall} of Eq. 3.3: the gray and blue curves refers to the model where no multiplicative connections are used to learn the hallucination stream (row #14 of Table 3.2). We started the experiment with learning rate set to 0.001, and continued after a while with learning rate set to 0.0001. The red curve shows instead L_{hall} after plugging the inter-stream connections (row #13 of Table 3.2).	31
4.1	Architecture and training steps (solid lines - module is <i>trained</i> ; dashed lines - module is <i>frozen</i>). Step 1: Separate pretraining of RGB and Depth networks (Resnet-50 backbone with temporal convolutions). The bottleneck described in section 4.2.2 is highlighted as a separate component. At test time the raw predictions (logits) of the two separate streams are simply averaged. The complementary information carried by the two streams bring a significant boost in the recognition performance. Step 2: The depth stream is frozen. The hallucination stream H is initialized with the depth stream’s weights and adversarially trained against a discriminator. The discriminator is fed with the concatenation of the bottleneck feature vector and the temporal frame ordering label y^t , as detailed in Section 4.2.1. The discriminator also features an additional classification task, i.e. not only it is trained to discriminate between hallucinated and depth features, but also to assign samples to the correct class (Eq. 4.2). The hallucination stream thus learns monocular depth features from the depth stream while maintaining discriminative power. At test time, predictions from the RGB and the hallucination streams are fused.	42

LIST OF FIGURES

4.2	Detail of the ResNet residual unit with temporal convolutions (blue block).	47
4.3	Architectures for the discriminators used for the two different tasks. Left: D1 for object recognition. Right: D2 for action recognition.	48
4.4	Examples of RGB and depth frames from the NYUD (RGB-D) dataset.	50
4.5	Discriminator confidence at predicting 'fake' label as a function of noise in the depth frames. The more corrupted the frame, the more confident D , and the lower the accuracy of the Two-stream model (NYUD dataset).	60
5.1	Distillation Multiple Choice Learning (DMCL) allows multiple modalities to cooperate and strengthen one another. For each training sample, the modality specialist m that achieves the lowest loss ℓ distills knowledge to strengthen other modality specialists. At test time, any subset of available modalities can be used by DMCL to make predictions.	66
5.2	Distillation Multiple Choice Learning (DMCL) In the Forward Pass, we calculate the classification cross-entropy losses ℓ for each modality and identify the teacher network - in this case, the Depth network. In the Backward Pass, we compute the soft targets of the teacher, S_D , and use them as an extra supervision signal for the student networks. The loss for the student networks ℓ^{GD} refers to the Generalized Distillation loss, defined on Eq. 5.3. The loss for the teacher network D uses the normal logits, <i>i.e.</i> soft targets with temperature $T = 1$. At test time, we are able to cope with missing modalities. The final prediction is obtained by averaging the predictions of the available modalities.	68

5.3 The cross-entropy loss of three networks independently trained for action recognition on the UWA3DII dataset, using RGB (blue), depth (green), and optical flow (orange). These plots are averaged over three runs. We observe that for the first 10K steps, the training loss of the optical flow network is consistently lower, resulting in a winner-takes-all behavior in traditional MCL algorithms. However, in DMCL, the winner network also teaches the loser networks, strengthening the other modality networks and avoiding this behavior. 80

List of Tables

3.1	Classification accuracies and comparisons with the state of the art. Performances referred to the several steps of our approach (ours) are highlighted in bold. \times refers to comparisons with unsupervised learning methods. \triangle refers to supervised methods: here train and test modalities coincide. \square refers to privileged information methods: here training exploits RGB+D data, while test relies on RGB data only.	27
3.2	Ablation study. A full set of experiments is provided for cross-subject evaluation protocol, and for the cross-view protocol, only the most important results are reported.	29
3.3	Accuracy of the model tested with clean RGB and noisy depth data. Accuracy of the proposed hallucination model, i.e. with <i>no depth</i> at test time, is 77.21%.	33
3.4	Inverting the cross-stream connection study. The last section of the table refers to results where the direction of the cross-stream connection has been inverted. The other results are also reported in the paper, as they refer to the model proposed.	35
4.1	Ablation Study - Bottleneck size. Hallucination network under-performing with $F_x \in \mathbb{R}^{2048}$	51

4.2	Ablation Study - Investigating different bottleneck implementations. The Table reports Hallucination network performances on NTU-mini.	52
4.3	Ablation Study - Investigating different inputs and tasks for the discriminator. The Table reports Hallucination network performances (NTU-mini).	53
4.4	Classification accuracies and comparisons with the state of the art for video action recognition. Performances referred to the several steps of our approach (ours) are highlighted in bold. \times refers to comparisons with unsupervised learning methods. \triangle refers to supervised methods: here train and test modalities coincide. \square refers to privileged information methods: here training exploits RGB+D data, while test relies on RGB data only. The 4th column refers to cross-subject and the 5th to the cross-view evaluation protocols on the NTU dataset. The results reported on the other two datasets are for the cross-view protocol. . .	56
4.5	Object Recognition	57
4.6	Accuracy values for the two-stream model trained on RGB and depth, and tested with RGB and noisy depth data.	61
5.1	Comparing MCL methods. We compare the performance of SMCL and CMCL with our proposed DMCL on the NWUCLA, UWA3DII, and NTU120 datasets. We also compare against independently trained modality networks. For each method we present the accuracy of the RGB modality network, the sum of all modality network predictions (\sum), and the oracle accuracy (Φ). For each row, corresponding to one dataset, we highlight in bold the best result using RGB only at test time. Using our DMCL methods results in better RGB networks for three out of four datasets. . .	77

5.2 Selecting the right teacher network is important. We present the action recognition classification accuracy on the NWUCLA and UWA3DII datasets for three scenarios, where: modality networks are trained independently; a random teacher is assigned for every sample to guide the other modality networks; and DMCL, where the best-performing teacher (lowest loss) is selected to guide other modality networks. For each column, corresponding to a test modality, we highlight in bold the best result across the three scenarios. 77

5.3 Accuracy of a KNN classifier with varying k on the NWUCLA dataset. Classified features are computed using randomly initialized networks for each modality. Although all features are randomly generated, optical flow random features tend to achieve a significantly higher accuracy. This helps to explain why optical flow networks learn faster than other modalities. 79

5.4	Accuracy for UWA3DII and NWUCLA dataset. The first part of the table refers to methods that use unsupervised feature learning (*) or that use the same number of modalities for training and testing. The second part of the table refers to methods that use more modalities for training than for testing. Methods that use RGB ⁺ at test time use an additional network that mimics the missing modality. For each column, corresponding to one dataset, we highlight in colored bold the best result and in normal colored font the second best between our method and the baselines. Each color corresponds to a different test modality. To conduct a fair comparison with baseline methods, this table presents results for the most common view setting for UWA3DII and NWUCLA. Other view settings follow the same trend and results are presented in the supplementary material.	83
5.5	Evaluation on NTU datasets. The test sets for NTU120 ^{mini} and NTU120 are the same. For each column, corresponding to one dataset, we highlight in bold the best result and in normal colored font the second best between our method and the baselines. Each color corresponds to a different test modality. The approximated values are inferred from a plot in [2]. We note that the effect of the distillation method is more visible on the smaller scale versions NTU60 and NTU120 ^{mini} of the dataset.	84

Chapter 1

Introduction

1.1 Objective, Motivation, and Challenges

Depth perception is the ability to reason about space in the 3D world, critical for the survival of many hunting predators and an important skill for humans to understand and interact with the surrounding environment. It develops very early in humans when babies start to crawl [3], and emerges from a variety of mechanisms that jointly contribute to the sense of relative and absolute position of objects, called depth cues. Besides binocular cues, *e.g.* stereovision, humans use monocular cues that relate to *a priori* visual assumptions derived from 2D single images through shadows, perspective, texture gradient, and other signals - *e.g.* the assumption that objects look blurrier the further they are, or that if an object occludes another it must be closer, *etc.* [4]. As matter of fact, although humans underestimate object distance in a monocular vision setup [5], we are still able to perform most of our vision-related tasks with good efficiency even with one eye covered.

Similarly, depth perception is often of paramount importance for many computer vision tasks related to robotics, autonomous driving, scene understanding,

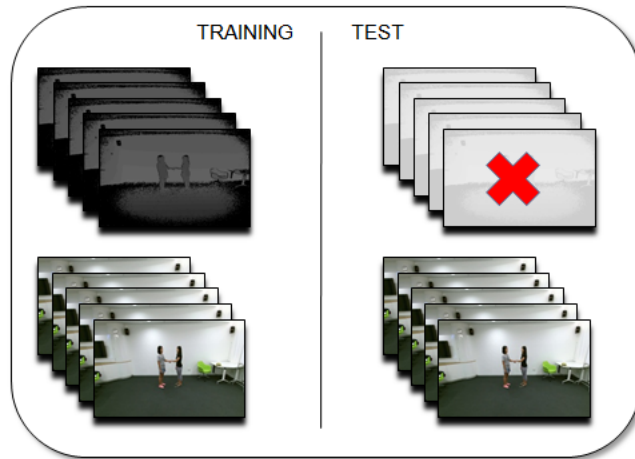


Figure 1.1: What is the best way of using all data available at training time, considering a missing (or noisy) modality at test time?

to name a few. The emergence of cheap depth sensors and the need for big data led to big multimodal datasets containing RGB, depth, infrared, and skeleton sequences [6], which in turn stimulated multimodal deep learning approaches. Traditional computer vision tasks like action recognition, object detection, or instance segmentation have been shown to benefit performance gains if the model considers other modalities, namely depth, instead of RGB only [7; 8; 9; 10].

However, it is reasonable to expect that depth data is not going to be always available when a model is deployed in real scenarios, either due to the impossibility to collect depth data with enough quality, *e.g.* due to far-distance or reflectance issues, or to install depth sensors everywhere, sensor or communications failure, or other unpredictable events.

Considering this limitation, we would like to answer the following question (depicted in Fig. 1.1): what is the best way of using all data available at training time, in order to learn robust representations, knowing that there are missing (or noisy) modalities at test time? In other words, is there any added value in training a model by exploiting multimodal data, even if only one modality is

1.1 Objective, Motivation, and Challenges

available at test time?

Unsurprisingly, the simplest and most commonly adopted solution consists in training the model using only the modality in which it will be tested. Nevertheless, a more interesting alternative is to exploit the potential of the available data and train the model using all modalities, being however aware of the fact that not all of them will be accessible at test time. This learning paradigm, *i.e.*, when the model is trained using extra information, is generally known as *learning with privileged information* [11] or *learning with side information* [12].

This work investigates multimodal learning with the goal to develop computer vision models that leverage the complementarity offered by diverse modalities at training time, while being robust to missing modalities at test time. We are mainly interested in developing methods that are flexible regarding the input modalities and training or evaluation tasks. The idea of learning from multimodal data while being aware that modalities may be missing at test time is central for perception in general, and in particular for the next-generation of computer vision applications *e.g.* concerning robotics.

The ability to reason how different data modalities relate to each other is linked to the practical low-level task of predicting one modality from the other. A classical example is depth estimation from RGB images. This task can also be defined in the feature space, rather than the input space. In this thesis, we approach this problem from a high-level perspective, *i.e.* we are interested in estimating high-level features and predictions that correspond to the depth network, instead of the actual depth map of the scene. One of the main challenges of multimodal learning is to develop a method that efficiently leverages the different advantages that diverse modalities offer. We address this problem from the more challenging perspective of being able to account for a missing modality for inference. We develop deep learning methods that learn using RGB, Depth, and

Optical Flow data. We extensively evaluate our methods on the task of video action recognition, and also provide results on the task of object recognition. The next section discusses the main contributions of this thesis.

1.2 Contributions and Outline

This thesis will describe several models for multimodal learning using privileged information.

Chapter 2 describes the related work and places our work at the intersection of three topics, namely Generalized Distillation, Adversarial Learning, and Multimodal Deep Learning. Specific works that are closer to the methods presented in the following chapters are discussed within the corresponding chapter.

Chapter 3 introduces a model that learns from RGB and depth, and uses RGB only at test time for video action classification. This is accomplished by means of an additional network that learns to mimic the missing modality features and predictions, called hallucination network, using the modality that is available as input. This chapter is mainly based on the publication:

- [13] N. C. Garcia, P. Morerio, and V. Murino, "Modality distillation with multiple stream networks for action recognition", in The European Conference on Computer Vision (*ECCV*), September 2018.

Chapter 4 extends the previous work to the task of object recognition, and presents a novel method to learn the hallucination network. We develop an adversarial learning strategy to align the features and predictions across modalities. We also evaluate this method using noisy data, and present a mechanism to automatically switch to hallucinated features and predictions in case the input data is too noisy. This chapter is based on the publication:

- [14] - N. C. Garcia, P. Morerio, and V. Murino, "Learning with privileged information via adversarial discriminative modality distillation", Transactions on Pattern Analysis and Machine Intelligence (*TPAMI*), 2019.

Chapter 5 investigates how multimodal data can be used in a cooperative learning setting. We present an algorithm to learn an ensemble of multimodal networks simultaneously, that leverage the strengths of each corresponding modality to the benefit of the ensemble and themselves. This algorithm is robust to missing modalities, hence also related to the privileged information learning framework. We evaluate this method using RGB, Depth, and Optical Flow data for the task of video action recognition. This work refers to a paper under revision.

- [15] - N. C. Garcia, S. A. Bargal, V. Ablavsky, P. Morerio, V. Murino, and S. Sclaroff, "DMCL: Distillation Multiple Choice Learning", *under revision*, 2019.

Chapter 6 discusses future directions and applications of multimodal learning with privileged information, and in particular of techniques developed in this thesis.

1.2.1 List of Publications

To summarize the publications described in this thesis are:

- [13] N. C. Garcia, P. Morerio, and V. Murino, "Modality distillation with multiple stream networks for action recognition", in The European Conference on Computer Vision (*ECCV*), September 2018.
- [16] N. C. Garcia, P. Morerio, and V. Murino, "Chapter 12 - cross-modal learning by hallucinating missing modalities in rgb-d vision," in Multimodal Scene Understanding (M. Y. Yang, B. Rosenhahn, and V. Murino, eds.), pp. 383 – 401, Academic Press, 2019

1.2 Contributions and Outline

- [14] - N. C. Garcia, P. Morerio, and V. Murino, "Learning with privileged information via adversarial discriminative modality distillation", Transactions on Pattern Analysis and Machine Intelligence (*TPAMI*), 2019.
- [15] - N. C. Garcia, S. A. Bargal, V. Ablavsky, P. Morerio, V. Murino, and S. Sclaroff, "DMCL: Distillation Multiple Choice Learning", *under revision*, 2019.

Chapter 2

Related Work

The book of reference by Goodfellow *et al.* [17] gives a detailed perspective on the field of Deep Learning. In this chapter, we review mainly deep learning methods that are closer to our work, namely related to the topics of Knowledge Distillation, Adversarial Learning, and Multimodal Deep Learning.

2.1 Generalized Distillation

The Generalized Distillation framework, proposed in [18], gives a unifying perspective on two distinct theories related to the concept of machines-teaching-machines: *Privileged Information* [11] and *Knowledge Distillation* [19][20]. The former, also known as Learning Using Privileged Information (*LUPI*), introduces to the learning process the concept of a "teacher" model that provides additional information to a "student" model, in addition to the label supervision. The intuition is that the teacher's additional explanations enable the student to learn a better model than if it would be trained using label supervision only. Importantly, the additional information provided by the teacher is only available to the student at training time, thus the term *privileged* information.

On the other hand, Knowledge Distillation (*KD*) proposes a training procedure to transfer knowledge from a previously trained large model or ensemble of models to a small model, thus distilling information from a heavier to a lighter model. This idea comes from the fact that speed and computation requirements for training and testing phases are very different.

These ideas have in common the concept of machines-teaching-machines: the model used for inference learns from a model that was previously trained in a more advantageous condition, *e.g.* using additional information, better quality data, or simply is an aggregate of several large models. The work presented in this thesis are both related to the privileged information theory and to knowledge distillation, and address these from a multimodal perspective.

We are interested in exploring additional modalities only available at training time, such as depth and optical flow, which are considered to be privileged information in our approaches. The knowledge distillation framework is at the core of our methods as the mechanism to distill the knowledge offered by models that use the additional modalities.

The idea of using privileged information was explored in many applications. Luo *et. al.* [21] proposed an interesting model that is first trained on several modalities (RGB, depth, and three features joints-based), but tested only in one of these. The method uses a graph-based distillation mechanism to distill information between all modalities at training time. The training process is split in different stages, a first one of pretraining using all modalities and a large dataset, and then a second one using a subset of modalities and a smaller dataset. The test set consists of a single modality from the smaller dataset. This achieves state-of-the-art results in action recognition and action detection tasks. Learning with privileged information for action recognition has also been explored for recurrent neural networks. In [22], the authors devise a method that uses skeleton

joints as privileged information to learn a better action classifier that uses depth, even with scarce data.

The work of Hoffman *et al.* [12] introduced a model to hallucinate depth features from RGB input for object detection task. This approach learns the hallucination network by minimizing an Euclidean loss between the true depth features and hallucinated feature maps. In addition, the final loss function includes more than ten classification and localization losses, balanced using the corresponding hyperparameters. Our work is inspired in this approach and we extend some of these ideas to other tasks, and by formulating our problem within the generalized distillation framework.

An interesting work lying at the intersection of multimodal learning and learning with privileged information is ModDrop by Neverova *et al.* [23]. The authors propose a modality-based dropout strategy, where each input modality is entirely dropped (actually zeroed) with some probability during training. The resulting model is proved to be more resilient to missing modalities at test time.

The idea of knowledge distillation was initially applied to network compression [24], and have since then been applied in many creative ways to a variety of domains such as language tasks [25], defending from adversarial attacks [26], transfer labels across domains [27], unifying classifiers using unlabeled data [28], or using distillation without a pre-trained teacher [29] [30]. The gains provided by Knowledge Distillation are still not completely understood in the literature. With this work, we hope to provide insights on its application to multimodal data.

2.2 Adversarial Learning

Chapter 4 is closely related to this body of work. Our method implements an adversarial strategy to generate features from the missing modality feature space, using RGB as input.

In the seminal paper of Goodfellow *et al.* [31], the authors propose a generative model that is trained by having two networks playing the so called *minimax* game. A generator network is trained to generate images from noise vectors, and a discriminator network is trained to classify the generated images as false and images sampled from the dataset as true. As the game evolves, the generator becomes better and better at generating samples that look like the true images from the data distribution. This is usually referred to as Generative Adversarial Networks (GANs).

The concept of adversarial training was explored in many different tasks and domains other than image generation, such as disentangling semantic concepts [32], network compression [33] [34] [35], feature augmentation [36], image to image translation [37]. The training stability was improved by exploring different losses [38] and other tricks related to the implementation of GANs [39][40].

An important variant of the GAN framework are Conditional GANs (CGANs) [41], that propose to concatenate the label of desired class to be generated, to the noise vector. The CGAN model has been used in different domains, from image synthesis [42] to domain adaptation [36]. We extend the CGAN idea to achieve the goal of temporal correspondence between the generated and the target feature vectors. This is discussed in more detail in Chapter 4. Perhaps more similar to our work is the interesting paper by Roheda *et al.*[43], that also approaches the problem of missing modalities in the context of adversarial learning. The authors address the binary task of person detection using images, seismic, and acoustic sensors, where the latter two are absent at test time. A CGAN is conditioned

on the available images and the generator maps a vector noise to representative information from the missing modalities, with an auxiliary L2 loss. In contrast to this work, our CGAN model learns a mapping directly from the test modality to the feature space of the missing modality, with no auxiliary loss. We also focus on different tasks, namely video action recognition and object recognition.

2.3 Multimodal Deep Learning

2.3.1 RGB-D Vision

Video action recognition and object detection have a long and rich field of literature, spanning from classification methods using handcrafted features, *e.g.* [44; 45; 46; 47; 48; 49] to modern deep learning approaches, *e.g.* [9; 50; 51; 52; 53; 54], using either RGB-only, representations obtained from RGB such as optical flow, depth data, or a combination of these. We point to some of the more relevant works in video action recognition and object recognition using multimodal data and also to state-of-the-art methods that consider the privileged information scenario or a missing modality at test time.

Multimodal Video Action Recognition

A more comprehensive review is presented in [55] [56] [57]. The two-stream model introduced by Simonyan and Zisserman [50] is a landmark on video analysis, and since then has inspired a series of variants that achieved state-of-the-art performance on diverse datasets. This architecture is composed by a RGB and an optical flow stream, which are trained separately, and then fused at the prediction layer. The RGB network models mainly appearance features and the optical flow, due to being specifically designed to represent movement, models motion. In [1], the authors propose a variant of this architecture, which models

spatiotemporal features by injecting the motion stream’s signal into the residual unit of the appearance stream. They also employ 1D temporal convolutions along with 2D spatial convolutions. The combination of 2D spatial and 1D temporal convolutions has shown to learn better spatiotemporal features than 3D convolutions [58]. The current state of the art in video action recognition [59] uses 3D temporal convolutions and a new building block dedicated to capture long range dependencies, using RGB data only. We explore some of these architectures on Chapters 3, 4, and 5.

Some interesting works use modules specifically developed to learn motion features, which are then incorporated in models that use RGB only [60] [61] [62] [63]. Other methods learn an additional hallucination network to mimic the features of optical flow [64].

In [7], the complementary properties of RGB and depth data are explored, taking the NTU RGB+D dataset as testbed. This work designed a deep autoencoder architecture and a structured sparsity learning machine, and showed to achieve state-of-the-art results for action recognition. Liu *et al.* [8] also use RGB and depth complementary information to devise a method for viewpoint invariant action recognition, extensively evaluated on the NTU RGB+D dataset. Here, dense trajectories from RGB data are first extracted, which are then encoded in viewpoint invariant deep features, while a similar procedure is followed for the depth stream. The RGB and depth features are then used as a dictionary to predict the test label.

We mainly use three datasets for action recognition, which offer RGB and Depth data. These are the UWA3DII [65], the NWUCLA [66], and the NTU60 and NTU120 RGB+D [67] [2]. We describe these datasets in the experimental sections of the methods, along with the training and testing protocols.

Object Recognition

Over the years, object recognition based on RGB and depth have been an insightful task to reason on the complementarity of these two modalities, and whether depth data should be handled differently compared to RGB. An example of this is [9], in which the authors propose to encode depth images using a geocentric embedding that encodes height above ground and angle with gravity for each pixel in addition to the horizontal disparity, showing that it works better than using raw depth. Differently, in [54], the authors focus on carefully designing a convolutional neural network including a multimodal layer to fuse RGB and depth. Our work differs from these approaches since we focus on learning a model that has access to depth only at training time, which fundamentally changes the feature learning approach.

2.3.2 Ensemble Learning

A comprehensive review about ensemble methods is presented in [68]. The most relevant method to our work, specially to Chapter 5, is the Multiple Choice Learning (MCL) framework. Guzman-Rivera *et al.* [69] proposed MCL to optimize the oracle accuracy of an ensemble of models. The oracle accuracy refers to the top-1 accuracy from the set of predictions produced by the ensemble models. Lee *et al.* [70] proposed Stochastic MCL, an adaptation of MCL to an ensemble of neural networks that have as input RGB, and learn via stochastic gradient descent. Each network of the ensemble trained via Stochastic MCL produces a set of diverse outputs. The inability to output a single prediction compromises its use in real applications. Lee *et al.* [71] addressed this issue with Confident MCL. The main idea is to avoid confident predictions for the classes not assigned to a given specialist. This allows for the sum of all ensemble's networks outputs to get a single prediction. Tian *et al.* [72] also addressed this issue by training

an additional network to estimate the weight of the outputs of each specialist. While [71] and [72] propose ways to get a single prediction out of the ensemble, they do not address how such methods can be used with multimodal data.

We draw inspiration on these works to address this issue within the MCL framework. Chapter 5 addresses multimodal learning from the perspective of ensemble learning, *i.e.* learning an ensemble of networks that have as input different modalities and learn simultaneously and cooperatively.

Chapter 3

Modality Distillation with Multiple Stream Networks for Action Recognition

3.1 Introduction

Imagine to have a large multimodal dataset to train a deep learning model on, for example consisting in RGB video sequences, depth maps, infrared, and skeleton joints data. However, at test time, this model may be used in scenarios where not all of these modalities are available - for example, most of the cameras capture RGB only, which is the most common and cheapest available data modality. ¹

Considering this limitation, what is the best way of using all data available to learn robust representations to be exploited when there are missing modalities at test time? In other words, is there any added value to train a model by exploiting more data modalities, even if only one can be used at test time? The simplest and most commonly adopted solution could be to train the model using only the

¹This chapter is based on the publications [16; 73]

modality in which it will be tested. However, a more interesting alternative is trying to exploit the potential of the available data and train the model using all available modalities, realizing, however, that not all of them will be accessible at test time. This learning paradigm, i.e., when the model is trained using extra information, is generally known as *learning with privileged information* [11] or *learning with side information* [12].

In this work, we propose a multimodal stream framework that learns from different data modalities and can be deployed and tested on a subset of these. We design a model able to learn from RGB *and* depth video sequences, but due to its general structure, it can also be used to manage whatever combination of other modalities as well. To show its potential, we evaluate the performance on the task of video action recognition. In this context, we introduce a new learning paradigm, depicted in Fig. 3.1, to *distill* the information conveyed by depth into an *hallucination* network, which is meant to “mimic” the missing stream at test time. Distillation [19][20] refers to any training procedure where knowledge is transferred from a previously trained complex model to a simpler one. Our learning procedure also introduces a new loss function which is inspired to the *generalized distillation* framework [18], that unifies distillation and privileged information learning theories. Our model is inspired to the two-stream network introduced by Simonyan and Zisserman [50], which uses RGB and optical flow, and has been notably successful in the traditional setting for video action recognition task [74][1]. Differently, we use multimodal data, deploying one stream for each modality (RGB and depth in our case), and use it in the framework of privileged information.

Another inspiring work is [12], which proposed a hallucination network to learn with side information. We build on this idea, extending it by devising a new mechanism to *learn* and *use* such hallucination stream through a more

general loss function and inter-stream connections.

To summarize, the main contributions of this work are:

- we propose a new multimodal stream network architecture able to exploit multiple data modalities at training while using only one at test time;
- we introduce a new learning paradigm to learn a hallucination network within a novel two-stream model;
- in this context, we have designed an inter-stream connection mechanism to improve the learning process of the hallucination network, and a general loss function, based on the generalized distillation framework;
- we report state-of-the-art results – in the privileged information scenario – on the largest multimodal dataset for video action recognition, the NTU RGB+D [67].

The implementation of our method is available at <https://github.com/ngarcia/modality-distillation>.

The rest of the chapter is organized as follows. Section 3.2 details the proposed architecture and the novel learning paradigm. Section 3.3 reports the results obtained on the NTU dataset, including a detailed ablation study and a comparative performance with respect to the state of the art. Finally, we draw conclusions and future research directions in section 3.4.

3.2 Model

3.2.1 Cross-stream multiplier networks

We design our model (Figure 3.1) based on the architecture presented in [1], which in turn derives from the two-stream architecture originally proposed in

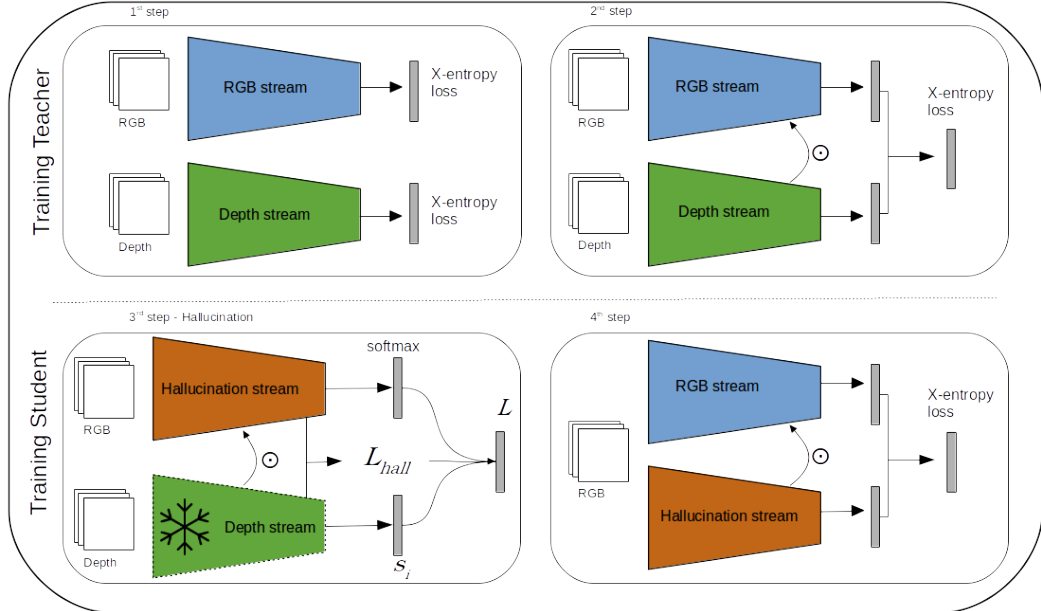


Figure 3.1: Training procedure described in section 3.2.3 (see also text therein). The 1st step represents the segregate training of the appearance and depth stream networks. The 2nd step illustrates the two-stream joint training. The 3rd step refers to the hallucination learning step using the soft labels with temperature s_i (eq. 3.6) and the novel distillation loss L (eq. 3.7), where the weights of the depth stream network are frozen. The 4th step refers to a fine-tuning step, and exemplifies also the testing setup, in which RGB data is the only input to the model.

[50]. Typically, the two streams are trained separately and the predictions are fused with a late fusion mechanism. These models use as input appearance (RGB) and motion (optical flow) data, which are fed separately into each stream, both in training and testing. Instead, in this work we use RGB and depth frames as inputs for training, but only RGB at test time, as already discussed.

We use the ResNet-50-based [75][76] model proposed in [1] as baseline architecture for each stream block of our model. In this paper, Feichtenhofer *et al.* proposed to connect the appearance and motion streams with multiplicative connections at several layers, as opposed to previous models which would only

interact at the prediction layer. Such connections are depicted in Figure 3.1 with the \odot symbol. Figure 3.2 illustrates this mechanism at a given layer of the multiple stream architecture, but, in our work, it is actually implemented at the four convolutional layers of the Resnet-50 model. The underlying intuition is that these connections enable the model to learn better spatiotemporal representations, and help to distinguish between identical actions that require the combination of appearance and motion features. Originally, the cross-stream connections consisted in the injection of the motion stream signal into the other stream’s residual unit, without affecting the skip path. ResNet’s residual units are formally expressed as:

$$\mathbf{x}_{l+1} = f(h(\mathbf{x}_l) + F(\mathbf{x}_l, \mathcal{W}_l)), \quad (3.1)$$

where \mathbf{x}_l and \mathbf{x}_{l+1} are l -th layer’s input and output, respectively, F represents the residual convolutional layers defined by weights \mathcal{W}_l , $h(\mathbf{x}_l)$ is an identity mapping and f is a ReLU non-linearity. The cross-streams connections are then defined as

$$\mathbf{x}_{l+1}^a = f(\mathbf{x}_l^a) + F(\mathbf{x}_l^a \odot f(\mathbf{x}_l^m), \mathcal{W}_l), \quad (3.2)$$

where \mathbf{x}^a and \mathbf{x}^m are the appearance and motion streams, respectively, and \odot is the element-wise multiplication operation. Such mechanism implies a spatial alignment between both feature maps, and therefore between both modalities. This alignment comes for free when using RGB and optical flow, since the latter is computed from the former in a way that spatial arrangement is preserved. However, this is an assumption we can not generally make. For instance, depth and RGB are often captured from different sensors, likely resulting in spatially misaligned frames. We cope with this alignment problem in the method’s initialization phase (described in the supplementary material). In order to augment the

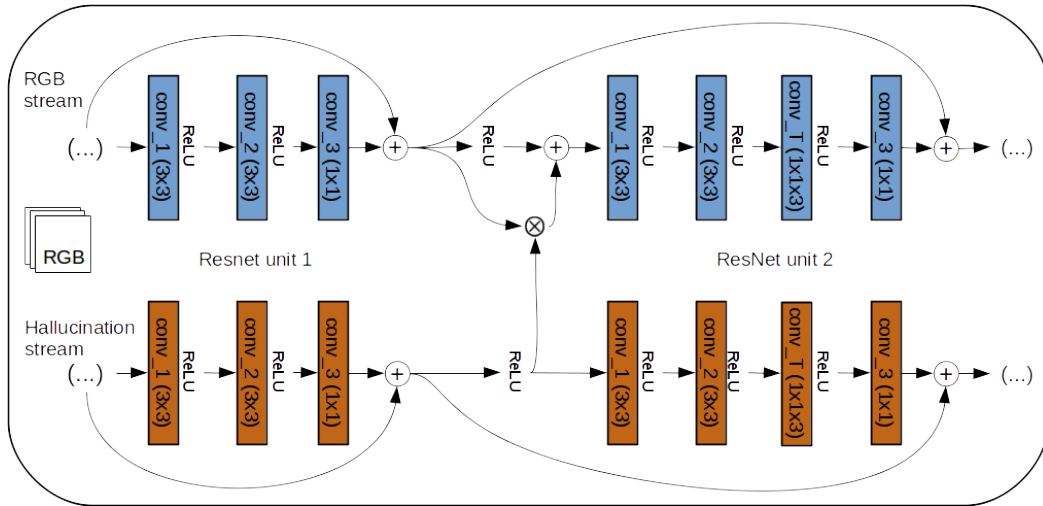


Figure 3.2: Detail of the ResNet residual unit, showing the multiplicative connections and temporal convolutions [1]. In our architecture, the signal injection occurs before the 2^{nd} residual unit of each of the four ResNet blocks.

model temporal support, 1D temporal convolutions into the second residual unit of each ResNet layer is also included [1], as illustrated in Fig. 3.2. The weights $W_l \in \mathbb{R}^{1 \times 1 \times 3 \times C_l \times C_l}$ are convolutional filters initialized as identity mappings at feature level, and centered in time, and C_l are the number of channels in layer l .

3.2.2 Hallucination stream

We also introduce and learn a hallucination network [12], using a new learning paradigm, loss function and design mechanism. The hallucination stream network has the same architecture as the appearance and depth stream models. This network receives RGB as input, and is trained to “imitate” the depth stream at different levels, *i.e.* at feature and prediction layers. In this work, we explore several ways to implement such learning paradigm, including both the training procedure and the loss, and how they affect the overall performance of the model.

In [12], a regression loss between the hallucination and depth feature maps is

designed, defined as:

$$L_{hall}(l) = \lambda_l \|\sigma(A_l^d) - \sigma(A_l^h)\|_2^2, \quad (3.3)$$

where σ is the sigmoid function, and A_l^d and A_l^h are the l -th layer activations of depth and hallucination network. This Euclidean loss forces both activation maps to be similar. In [12], this loss is weighted along with another ten classification and localization loss terms, making it hard to balance the total loss. One of the main motivations behind our proposed new staged learning paradigm, described in section 3.2.3, is to avoid the inefficient, heuristic-based tweaking of so many loss weights, aka hyper-parameter tuning.

Instead, we adopt an approach inspired by the generalized distillation framework [18], in which a *student* model $f_s \in \mathcal{F}_s$ distills the representation $f_t \in \mathcal{F}_t$ learned by the *teacher* model. This is formalized as:

$$f_s = \arg \min_{f \in \mathcal{F}_s} \frac{1}{n} \sum_{i=1}^n L_{GD}(i), n = 1, \dots, N \quad (3.4)$$

where N is the number of examples in the dataset. The generalized distillation loss is so defined as:

$$L_{GD}(i) = (1 - \lambda)\ell(y_i, \sigma(f(x_i))) + \lambda\ell(s_i, \sigma(f(x_i))), \lambda \in [0, 1] \quad (3.5)$$

and s_i are the soft predictions from the teacher network, that is:

$$s_i = \sigma(f_t(x_i)/T), T > 0. \quad (3.6)$$

The parameter λ in equation 3.5 allows to tune the loss by giving more importance either to imitating hard or soft labels, y_i and s_i , respectively, actually improving

the transfer of information from the depth (teacher) to the hallucination (student) network. The temperature parameter T in equation 3.6 allows to smooth the probability vector predicted by the teacher network. The intuition is that such smoothing may expose relations between classes that would not be easily revealed in raw predictions, further facilitating the distillation by the student network F_s .

We suggest that both Euclidean and generalized distillation losses are indeed useful in the learning process. In fact, by encouraging the network to decrease the distance between hallucinated and true depth feature maps, it can help to distill depth information encoded in the generalized distillation loss. Thus, we formalize our final loss function as follows:

$$L = (1 - \alpha)L_{GD} + \alpha L_{hall}, \quad \alpha \in [0, 1], \quad (3.7)$$

where α is a parameter balancing the contributions of the 2 loss terms during training. The parameters λ , α and T are estimated by utilizing a validation set. The details for their setting will be provided in the supplementary material.

In summary, the generalized distillation framework proposes to use the student-teacher framework introduced in the distillation theory to extract knowledge from the privileged information source. We explore this idea by proposing a new learning paradigm to train an hallucination network using privileged information, which we will describe in the next section. In addition to the loss functions introduced above, we also allow the teacher network to share information with the student network by design, through the cross-stream multiplicative connections. We test how all these possibilities affect the model’s performance in the experimental section through an extensive ablation study.

3.2.3 Training Paradigm

In general, the proposed training paradigm, illustrated in Fig. 3.1, is divided in two core parts: the first part (Step 1 and 2 in the figure) focuses on learning the teacher network F_t , leveraging RGB and depth data (the privileged information in this case); the second part (Step 3 and 4 in the figure) focuses on learning the hallucination network, referred to as student network F_s in the distillation framework, using the general hallucination loss defined in Eq. 3.7.

The *first* training step consists in training both streams separately, which is a common practice in two-stream architectures. Both depth and appearance streams are trained minimizing cross-entropy, after being initialized with a pre-trained ImageNet model for all experiments. As in [77], depth frames are encoded into color images using a jet colormap.

The *second* training step is still focused on further training the teacher model. This step gives the basis for the following hallucination network training, which, receiving in input RGB data, should behaves like an actual depth stream network. For this reason, we must train the depth stream network in the same setting as the hallucination model will act, hence, it is trained considering the cross-stream connections and adding the prediction fusion layer with the RGB stream model. Since the model trained in this step has the architecture and capacity of the final one, and *has access to both modalities*, its performance represents an upper bound for the task we are addressing. This is one of the major differences between our approach and the one used in [12]: by decoupling the teacher learning phase with the hallucination learning, we are able to both learn a better teacher *and* a better student, as we will show in the experimental section.

In the *third* training step, we focus on learning the hallucination network from the teacher model, *i.e.*, the depth stream network just trained. Here, the weights of the depth network are frozen, while receiving in input depth data. Instead,

the hallucination network, receiving in input RGB data, is trained with the loss defined in 3.7, while also receiving feedback from the cross-stream connections from the depth network. We found that this helps the learning process.

In the *fourth* and last step, we carry out fine tuning of the whole model, composed by the RGB and the hallucination streams. This step uses RGB only as input, and it also precisely resembles the setup used at test time. The cross-stream connections inject the hallucinated signal into the appearance RGB stream network, resulting in the multiplication of the hallucinated feature maps and the RGB feature maps. The intuition is that the hallucination network has learned to inform the RGB model where the action is taking place, similarly to what the depth model would do with real depth data.

A summary of the whole training process is reported as in the following box:

- **training step 1**
 - initialize RGB and depth streams with ImageNet-pretrained weights;
 - train depth and RGB streams *separately*, with depth and RGB data respectively and standard cross entropy classification loss;
- **training step 2** (*learning the teacher network*)
 - initialize both streams with weights learned in step 1;
 - train both streams jointly as a two-stream model [1] (*i.e.* with multiplier connections), using both RGB and depth data, with cross entropy loss;
- **training step 3** (*learning the student network*)
 - freeze depth network weights learned in step 2;
 - initialize hallucination network with depth weights;
 - train with cross-stream connections and the proposed loss L (eq. 3.7);
- **training step 4** (*finetune the final model*)
 - initialize the hallucination stream with weights learned in step 3;
 - initialize RGB stream with weights from step 2;
 - fine-tune the joint model composed by hallucination + RGB branches (with cross-stream connections) using RGB data only and cross entropy loss;

3.3 Experiments

3.3.1 NTU RGB+D Dataset

We evaluate our model on the NTU RGB+D dataset [67], which is one of the largest public dataset for multimodal video action recognition. It is composed by 56,880 videos, available in four modalities: RGB videos, depth sequences, infrared frames, and 3D skeleton data of 25 joints. It was acquired with a Kinect v2

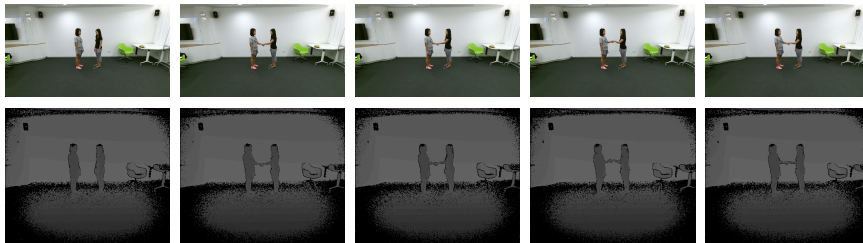


Figure 3.3: Example of RGB and depth frames from the NTU RGB+D Dataset.

sensor in 80 different viewpoints, and includes 40 subjects performing 60 distinct actions, including daily simple actions (*e.g.*, brushing teeth, drinking, writing), interactions (*e.g.*, kicking other person, hugging other person), and health-related actions (*e.g.*, nausea or vomiting condition, sneeze/cough). We follow the two evaluation protocols originally proposed in [67], which are cross-subject and cross-view. As in the original paper, we use about 5% of the training data as validation set for both protocols, in order to select the parameters λ , α and T . In this work, we use only RGB and depth data. The masked depth maps are converted to a three channel map via a jet mapping, as in [77].

3.3.2 Comparison with state of the art

Table 3.1 compares performances of different methods on the NTU RGB+D dataset. Classification accuracy is the standard performance measure used for this dataset: it is estimated according to the protocols (training and testing splits) reported in the respective works we are comparing with. The first part of the table (indicated by \times symbol) refers to the unsupervised method proposed in [78], which achieve surprisingly high results even without relying on labels in learning representations. The second part refers to supervised methods (indicated by \triangle), divided according to the modalities used for training and testing. Here, we list the performance of the separate RGB and depth streams trained in step

3.3 Experiments

Method	Test Modalities	Cross Subject	Cross View	
Luo [78]	Depth	66.2%	-	×
Luo [78]	RGB	56.0%	-	
HOG-2 [79]	Depth	32.4%	22.3%	
Ours - depth model, step 1	Depth	70.44%	75.16%	
Ours - RGB model, step 1	RGB	66.52%	71.39%	
Deep RNN [67]	Joints	56.3%	64.1%	△
Deep LSTM [67]	Joints	60.7%	67.3%	
Sharoudy [67]	Joints	62.93%	70.27%	
Kim [80]	Joints	74.3%	83.1%	
Sharoudy [7]	RGB+D	74.86%	-	
Liu [8]	RGB+D	77.5%	84.5%	
Ours - step 2	RGB+D	79.73%	81.43%	
Hoffman <i>et al.</i> [12]	RGB	64.64%	-	
Ours - step 3	RGB	71.93%	74.10%	□
Ours - step 4	RGB	73.42%	77.21%	

Table 3.1: Classification accuracies and comparisons with the state of the art. Performances referred to the several steps of our approach (ours) are highlighted in bold. × refers to comparisons with unsupervised learning methods. △ refers to supervised methods: here train and test modalities coincide. □ refers to privileged information methods: here training exploits RGB+D data, while test relies on RGB data only.

1, as a reference. Of course, we expect our final model to perform better than the one trained on RGB only. We also propose our baseline, consisting in the teacher model trained in step 2. Its accuracy represents an upper bound for the final model, which will not rely on depth data at test time. The last part of the table (indicated by □) reports our model’s performances at 2 different stages together with the other privileged information method [12]. For both protocols, we can see that our privileged information approach outperforms [12], which is the only fair *direct* comparison we can make (same training & test data). Besides, as ex-

pected, our final model performs better than “Ours - RGB model, step 1” since it exploits more data at training time, and worse than “Ours - step 2”, since it exploits less data at test time. Other RGB+D methods perform better (which is comprehensible since they rely on RGB+D in both training and test) but not by a large margin. More details and additional comments on the compared methods are provided in the supplementary material.

3.3.3 Ablation study

In this subsection, we discuss the results of the experiments carried out to understand the contribution of each part of the model and of the training procedure. Table 3.2 reports performances at the several training steps, different losses and model configurations.

Rows #1 and #2 refers to the first training step, where depth and RGB streams are trained separately. We can note that the depth stream network provides better performance with respect to the RGB one, as expected.

The second part of the table (Rows #3-5) shows the results using Hoffman *et al.*’s method [12], *i.e.* adopting a model initialized with the pre-trained networks from the first training step, and the hallucination network initialized using the depth network. Row #3 refers to the original paper [12] (*i.e.*, using the loss L_{hall} , Eq. 3.3), and rows #4 and #5 refer to the training using the proposed losses L_{GD} and L , in Eqs. 3.5 and 3.7, respectively. It can be noticed that the accuracies achieved using our proposed loss functions overcome that obtained in [12] by a significant margin (about 6% in the case of the total loss L).

The third part of the table reports performances after the training step 2. Rows #6 and #7 refer to the depth and RGB stream networks belonging to the model of row #8. This model corresponds to the architecture described in [1] and constitutes the upper bound for our hallucination model, since it uses RGB and

3.3 Experiments

#	Method	Test Modality	Loss	Cross-Subject	Cross-View
1	Ours - step 1, depth stream	Depth	x-entr	70.44%	75.16%
2	Ours - step 1, RGB stream	RGB	x-entr	66.52%	71.39%
3	Hoffman [12] w/o connections	RGB	eq. (3.3)	64.64%	-
4	Hoffman [12] w/o connections	RGB	eq. (3.5)	68.60%	-
5	Hoffman [12] w/o connections	RGB	eq. (3.7)	70.70%	-
6	Ours - step 2, depth stream	Depth	x-entr	71.09%	77.30%
7	Ours - step 2, RGB stream	RGB	x-entr	66.68%	56.26%
8	Ours - step 2	RGB & Depth	x-entr	79.73%	81.43%
9	Ours - step 2 w/o connections	RGB & Depth	x-entr	78.27%	82.11%
10	Ours - step 3 w/o connections	RGB (<i>hall</i>)	eq. (3.3)	69.93%	70.64%
11	Ours - step 3 w/ connections	RGB (<i>hall</i>)	eq. (3.3)	70.47%	-
12	Ours - step 3 w/ connections	RGB (<i>hall</i>)	eq. (3.4)	71.52%	-
13	Ours - step 3 w/ connections	RGB (<i>hall</i>)	eq. (3.7)	71.93%	74.10%
14	Ours - step 3 w/o connections	RGB (<i>hall</i>)	eq. (3.7)	71.10%	-
15	Ours - step 4	RGB	x-entr	73.42%	77.21%

Table 3.2: Ablation study. A full set of experiments is provided for cross-subject evaluation protocol, and for the cross-view protocol, only the most important results are reported.

depth for training and testing. Performances obtained by the model in row #8 and #9, with and without cross-stream connections, respectively, are the highest

in absolute when using both modalities (around 78-79% for cross-subject and 81-82% for cross-view protocols, respectively), largely outperforming the accuracies obtained using only one modality (in rows #6 and #7).

The fourth part of the table (rows #10-14) shows results for our hallucination network after the several variations of learning processes, different losses and using or not using the cross-stream connections. One can note that the achieved performances when *only* RGB data are given in input, are in line with those achieved by the model fed by depth data. Depending on the variant adopted, accuracies are around 70-72%, reaching about 72% in the case of application of our full model before the fine-tuning step (row #14, cross-subject protocol). The depth stream model (in row #6) reaches 71%, whereas the model with both modalities in input (fixing the upper bound, row #8) reaches about 79%: only 6 percentage points separate the 2 models, showing the goodness of our proposed approach.

Finally, the last row, #15, reports results after the last fine-tuning step, which allows to reach the best accuracy with only the RGB modality as input, increasing the previous performance of about 1.5%, so narrowing the gap to the upper bound to about 4.5%.

Contribution of the cross-stream connections

We claim that the signal injection provided by the cross-stream connections helps the learning of a better hallucination network. Row #13 and #14 show the performances for the hallucination network learning process, starting from the same point and using the same loss. The hallucination network that is learned using multiplicative connections performs better than its counterpart. This is illustrated in figure 3.4: even after approximately half the number of iterations, the hallucination network learned with the multiplicative cross-stream connections is

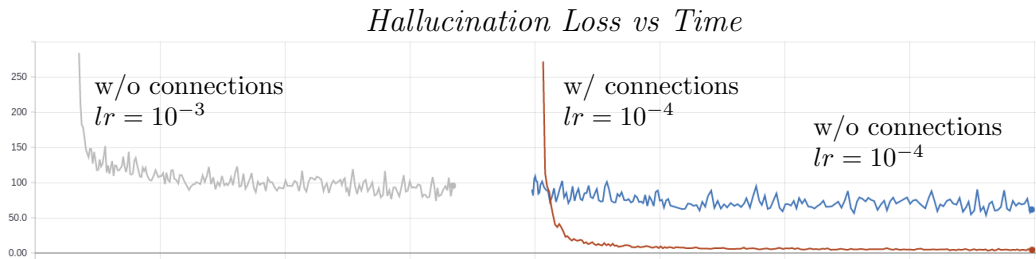


Figure 3.4: The plot shows the hallucination loss L_{hall} of Eq. 3.3: the gray and blue curves refers to the model where no multiplicative connections are used to learn the hallucination stream (row #14 of Table 3.2). We started the experiment with learning rate set to 0.001, and continued after a while with learning rate set to 0.0001. The red curve shows instead L_{hall} after plugging the inter-stream connections (row #13 of Table 3.2).

able to better minimize the Euclidean loss of Eq. 3.3.

Contributions of the proposed distillation loss (Eq. 3.7)

The distillation and Euclidean losses have complementary contributions to the learning of the hallucination network. This is observed by looking at the performances reported in rows #3, #4 and #5, and also #11, #12 and #13. Within both the training procedure proposed by Hoffman *et al.* [12] and our staged training process, the distillation loss improves over the Euclidean loss, and the combination of both improves over the rest. This suggests that both Euclidean and distillation losses have its own share and act differently to align the hallucination (student) and depth (teacher) feature maps and outputs' distributions.

Contributions of the proposed training procedure

The intuition behind the staged training procedure proposed in this work can be ascribed to the *dividi et impera* (divide-and-conquer) strategy. In our case, it means breaking the problem in two parts: learning the actual task we aim to solve and learning the hallucination network to face test-time limitations. Row #5

reports accuracy for the architecture proposed by Hoffman *et al.*, and rows #15 report the performance for our model with connections. Both use the same loss to learn the hallucination network, and both start from the same initialization. We observe that our method outperform the one in row #5, which justifies the proposed staged training procedure.

Finally, we motivate for the use of the hallucination model in comparison with other naive approaches when dealing with missing or noisy modalities. Comparing rows #2 with #15, we further confirm (if still needed) that using the hallucination model is in fact more useful than training only with RGB data. We also observe that it is more useful to use our hallucination model than naively use totally corrupted depth data as input to the two-stream model. This is observed by comparing results in Table 3.3 and the performance at row #15 in Table 3.2. The following section studies with further detail the behavior of our model when tested using noisy depth data as input.

3.3.4 Inference with noisy depth

Suppose that in a real test case we can only access unreliable, *i.e.* noisy, depth data. Now the question is: how much we can trust such data? How better would it be to use a model in which depth is provided by an hallucination network, like that proposed in this work? In other words, we are finally interested in exploring how our model works under stress, and, more precisely, at which level of noise, hallucinating the depth modality becomes favorable with respect to using the full model with both input modalities (step 2).

The depth sensor used in the NTU dataset (Kinect), is an IR emitter coupled with an IR camera, and has very complex noise characterization comprising at least 6 different sources [81]. It is beyond the scope of this work to investigate noise models affecting the depth channel, so, for our analysis we choose the most

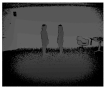
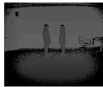

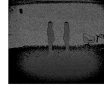
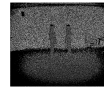
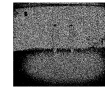

							
σ^2	<i>no noise</i>	10^{-3}	10^{-2}	10^{-1}	10^0	10^1	<i>void</i>
Accuracy	81.43%	81.34%	81.12%	76.85%	62.47%	51.43%	14.24%

Table 3.3: Accuracy of the model tested with clean RGB and noisy depth data. Accuracy of the proposed hallucination model, i.e. with *no depth* at test time, is 77.21%.

commonly adopted noise model, i.e., the multiplicative speckle noise.

Hence, we inject multiplicative Gaussian noise in the depth image I in order to simulate speckle noise: $I = I * n$, $n \sim \mathcal{N}(1, \sigma)$. Table 3.3 shows how performances of the network degrade when depth is corrupted with such Gaussian noise with increasing variance (cross-view protocol only). Results show that accuracy significantly decreases wrt the one guaranteed by our hallucination model (row #15 in Table 3.2), even with low noise variance. This means, in conclusion, that training an hallucination network is an effective way not only to obviate to the problem of a missing modality, but also to deal with noise affecting the input data channel.

3.3.5 Inverting the data modalities: RGB distillation

Despite the proposed architecture is general and can be applied to any multimodal pair of data streams, our model is not symmetric under the swap of the depth and RGB modalities. The connection between streams is engineered such that the RGB stream is fed with a signal coming from the depth stream, and not vice versa. The intuition for such choice of direction is that the depth stream learns from cleaner, more representative data (foreground depth maps), agnostic to texture, and is able to inform the RGB stream where the action is taking place, practically working as an augmentation tool for those regions of the feature map.

In fact, the depth stream alone performs better than the RGB alone.

In [1], the authors tested different locations where to inject the optical flow signal, *e.g.* inside or outside the ResNet residual unit. Bi-directional connections were also investigated, *i.e.* both streams were injected one into the other. It was concluded that injecting signal into the optical flow stream decreases the model performance, and suggest that the reason can be ascribed to the RGB stream becoming dominant during training. We hypothesize that the same reasoning can be applied to the depth stream, which in our model takes the place of optical flow. In [1], the authors did not try to invert the connection, *i.e.* to inject signal from RGB to optical flow. We report the results of such experiment in Table 3.4.

Line #8a reports the accuracy obtained by the teacher network at the end of step 2: not only such accuracy is lower than the one of our original teacher network (line #8), but also is only marginally higher than the one obtained by the final model (line #15), which only uses RGB at test time. Line # 8a represents thus a very poor upper bound (as compared to line # 8). This translates in a worse hallucination network (lines #13a) and worse distilled model (#15a).

3.3.6 Implementation details

Pre-processing & alignment

The multiplicative cross-stream connections present in our model require both RGB and depth frames to be spatially aligned, since they are element-wise operations over the feature maps. Such alignment comes for free when using RGB and optical flow - which is computed directly from the appearance frames. However, this is not normally the case when using depth and RGB frames that are acquired with different sensors, and have different dimensions and aspect ratios as in the NTU RGB+D dataset, or other Kinect-acquired data. Fortunately, the NTU dataset provides the joints' spatial coordinates in every

3.3 Experiments

#	Method	Test Modality	Loss	Cross-Subject	Cross-View
1	Ours - step 1, depth stream	Depth	x-entr	70.44%	75.16%
2	Ours - step 1, RGB stream	RGB	x-entr	66.52%	71.39%
Depth \rightarrow RGB (<i>compare to Table 2 of the paper</i>)					
6	Ours - step 2, depth stream	Depth	x-entr	71.09%	77.30%
7	Ours - step 2, RGB stream	RGB	x-entr	66.68%	56.26%
8	Ours - step 2	RGB & Depth	x-entr	79.73%	81.43%
13	Ours - step 3	RGB (<i>hall</i>)	eq. (7)	71.93%	74.10%
15	Ours - step 4	RGB	x-entr	73.42%	77.21%
Inverted - RGB \rightarrow Depth					
6a	Ours - step 2, depth stream	Depth	x-entr	66.6%	73.68%
7a	Ours - step 2, RGB stream	RGB	x-entr	63.98%	61.18%
8a	Ours - step 2	RGB & Depth	x-entr	<i>74.45%</i>	<i>78.55%</i>
13a	Ours - step 3	RGB (<i>hall</i>)	eq. (7)	<i>68.47%</i>	<i>72.77%</i>
15a	Ours - step 4	RGB	x-entr	<i>66.86%</i>	<i>73.34%</i>

Table 3.4: Inverting the cross-stream connection study. The last section of the table refers to results where the direction of the cross-stream connection has been inverted. The other results are also reported in the paper, as they refer to the model proposed.

RGB and depth frames, $rgb_{x,y}$ and $depth_{x,y}$ respectively, which we use to align both modalities. For every frame of a given video, we first compute the ratio $ratio_x^{A,B} = (rgb_x^A - rgb_x^B) / (depth_x^A - depth_x^B) \forall A, B \in S$, using all depth and RGB x coordinates from the frame’s well-tracked joints set S , and similarly for the y dimension. The video aspect ratio is then calculated as the mean between the median aspect ratio for x and the median aspect ratio for y dimensions. The RGB frames of a given video are scaled according to this ratio. Finally, both RGB and depth frames are overlaid by aligning both skeletons, and the intersection is cropped on both modalities. The cropped sections are then rescaled according to the network’s input dimension, in this case 224x224. Similarly to what was done in [1], we sample 5 frames evenly spaced in time for each video, both for training and testing. For training, we also flip horizontally the video frames with probability $P = 0.5$.

Hyperparameters and validation set

After validation, we have selected the following set of hyperparameters: $\alpha = 0.5$, $\lambda = 0.5$, $T = 10$. The validation set is not defined in the original paper where the dataset is presented [67]. For the sake of experiments reproducibility, we explain here how we defined the validation set. For the cross-subject protocol, we choose the subject #1 (from the training set), which corresponds to around 5% of the training set. For the cross-view protocol, we do the following: 1) create a dictionary of sorted videos for each key=action (from the training set); 2) set numpy random seed equal to 0; 3) sample 31 videos using `numpy.random.choice` for each action, which in the end will correspond to around 5% of the training set.

3.4 Summary

In this chapter, we addressed the task of video action recognition in the context of privileged information. We have proposed a new learning paradigm to teach an hallucination network to mimic the depth stream, yet receiving RGB as input. We have confirmed the value of knowledge distillation for multimodal learning, which we continue to explore in the following chapters. Our model outperforms many of the supervised methods evaluated on the NTU RGB+D dataset to the date, as well as the hallucination model proposed in [12]. We conducted an extensive ablation study to verify how the several parts composing our learning paradigm contribute to the model performance.

Chapter 4

Learning with Privileged Information via Adversarial Discriminative Modality Distillation

4.1 Introduction

Similarly to Chapter 3, this work addresses the problem of learning with multi-modal data in the context of privileged information ¹. We continue to investigate the case of having RGB and depth data for training, but RGB only for testing. In this work, we propose an adversarial strategy within a multimodal-stream framework to learn a hallucination network. We evaluate its performance on the task of video action recognition and object classification.

We introduce a new adversarial learning strategy to learn a hallucination network (Fig. 4.1), which goal is to mimic the test time missing modality features,

¹This chapter is based on the publication [14].

while preserving their discriminative power. The implementation of the adversarial strategy replaces the distance-base metric usually used to align the feature vectors, such as the Euclidean loss. It can be thought as a sort of programmable loss composed by the discriminator network and the adversarial loss. The hallucination network uses RGB only as input and tries to recover useful depth features for the task at hand. Such network can be thought as a source of monocular depth cues, *i.e.* a source of depth cues from a single 2D RGB image.

We would like to stress the fact that, in contrast to estimating real depth maps from RGB, we operate at feature level. Conceptually, it may seem that directly estimating depth maps from RGB is a more straightforward approach to deal with missing depth at test time. However, the task of depth estimation is arguably a much more difficult task to accomplish compared to the primary task at hand, which is action/object recognition from RGB sequences. A more reasonable approach is to reduce the depth estimation problem from the pixel space to a low dimensional space, while continuing to profit to some extent of the discriminative benefits offered by the depth modality.

On the one hand, our work is inspired by previous works using hallucination networks in the context of learning with privileged information. This was primarily proposed in [12], that presented an end-to-end single step training method to learn a hallucination network. This work was recently revisited in [13] considering a multi-step learning paradigm using a loss inspired by the generalized distillation framework [18]. On the other hand, adversarial learning has been shown to be a powerful tool to model data distributions [31; 82]. Building upon these ideas, we propose a new approach to learn the hallucination network via a discriminative adversarial learning strategy. Our proposed method has several advantages: it is agnostic regarding the pair of modalities used, which greatly simplifies its extension beyond RGB and depth data; and it is able to deal with videos by design,

by exploiting a form of temporal supervision as auxiliary information. Furthermore, it dumps the need to balance the different losses used in the other methods [12] [13]. Finally, thanks to the discriminator design, which includes an auxiliary classification task, our method is able to transfer the discriminative capability from a so-called *teacher network* [18] (depth network) to a *student* (hallucination network), up to a full recovery of the teacher’s accuracy. The implementation of this method is available at <https://github.com/pmorerio/admd>.

To summarize, the main contributions of this work are the following:

- We propose a new approach to learn a hallucination network within a multimodal-stream network architecture: it consists in an adversarial learning strategy that exploits multiple data modalities at training while using only one at test time. It proved to outperform its distance-based method counterparts [12; 13], and to augment its flexibility by being agnostic to components like distance metrics, data modalities, and size of the hallucinated feature vectors.
- More technically, we propose a discriminator network which is time-aware, and jointly solves 1) the classical binary classification task (real/generated), and 2) an auxiliary task, which inherently endows the learned features with discriminative power.
- We report results – in the privileged information scenario – on the NYUD [83] dataset for the task of object classification, and on the large-scale NTU RGB+D [67] and the Northwestern-UCLA [66] datasets for the task of action recognition.

The rest of the chapter is organized as follows. Section 4.2 presents the details of the proposed architecture and the novel learning strategy. Section 4.3 reports results on object recognition and video action recognition datasets, comparing

them to the current state of the art, and investigating how the different parts of our approach contribute to the overall performance through an extensive ablation study. Finally, we draw conclusions and future research directions in Section 4.4.

4.2 Model

Our goal is to train a hallucination network that, having as input RGB, is able to produce similar features to the ones produced by the depth network. The reasoning behind this idea is that on one hand depth and RGB provide complementary information for the task, but on the other hand RGB alone contains some cues for depth perception. Therefore, the goal of the hallucination network is to extract from RGB frames the complementary information that depth data would provide. It is important to emphasize that we are interested in recovering useful depth *features*, in contrast to estimating real depth maps from RGB.

This is accomplished in a two-step training procedure, illustrated in Fig. 4.1, and described in the following. The *first step* (Fig. 4.1, top) consists in training the RGB and depth streams individually, with the respective input modality, as two standard, separate, supervised learning problems. The resulting ensemble, obtained by fusing the predictions of the two sub-networks (not fine-tuned), represents the full model (two-stream) that can be used when both modalities are available at test time. Its accuracy should be taken as an upper bound for the model we are proposing. In the *second step* (Fig. 4.1, bottom), we actually train the hallucination network by means of the proposed adversarial learning strategy. As the hallucination network is trained in the context of adversarial learning to generate depth features, it can be also interpreted as the generator network in the traditional GAN framework [31]. However, strictly speaking, it is clearly to be considered as an encoder, which tries to extract monocular depth features from

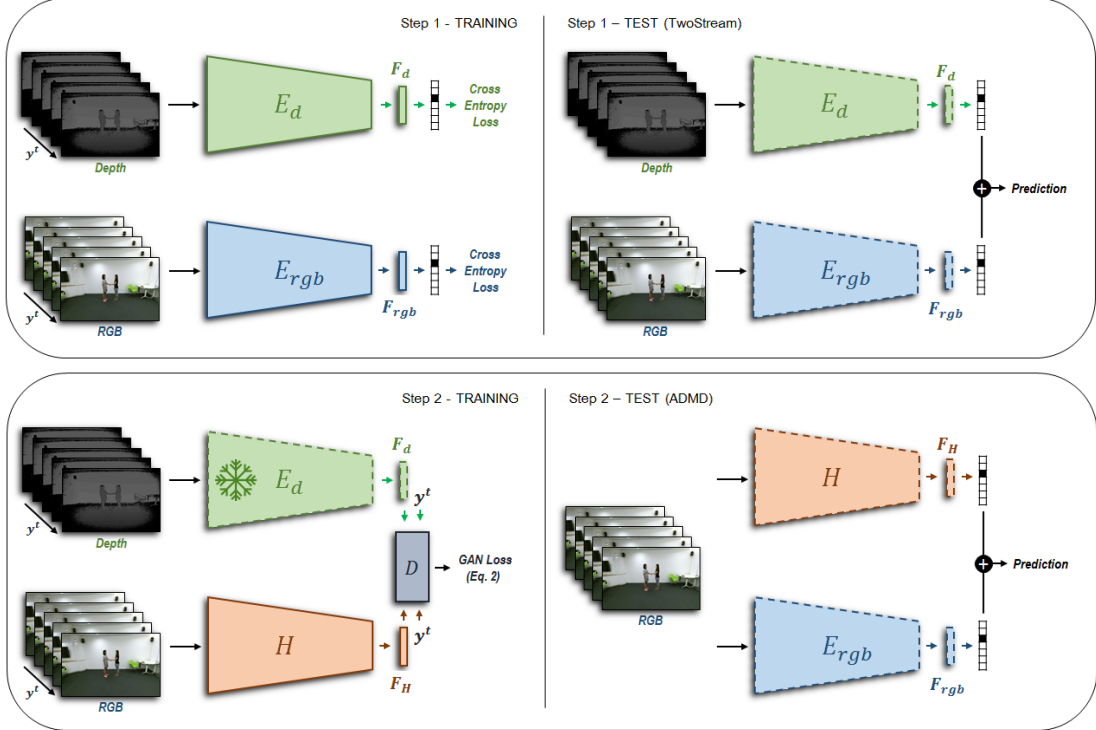


Figure 4.1: Architecture and training steps (solid lines - module is *trained*; dashed lines - module is *frozen*). **Step 1:** Separate pretraining of RGB and Depth networks (Resnet-50 backbone with temporal convolutions). The bottleneck described in section 4.2.2 is highlighted as a separate component. At test time the raw predictions (logits) of the two separate streams are simply averaged. The complementary information carried by the two streams bring a significant boost in the recognition performance. **Step 2:** The depth stream is frozen. The hallucination stream H is initialized with the depth stream’s weights and adversarially trained against a discriminator. The discriminator is fed with the concatenation of the bottleneck feature vector and the temporal frame ordering label y^t , as detailed in Section 4.2.1. The discriminator also features an additional classification task, i.e. not only it is trained to discriminate between hallucinated and depth features, but also to assign samples to the correct class (Eq. 4.2). The hallucination stream thus learns monocular depth features from the depth stream while maintaining discriminative power. At test time, predictions from the RGB and the hallucination streams are fused.

RGB input data. The test time setup of step 2 is again a two-stream model (not fine-tuned), composed by the RGB and hallucination networks, both having RGB

data as input.

4.2.1 Training procedure

Inspired by the generalized distillation paradigm, we follow a staged learning procedure, where the “teacher” net is trained first (Step 1) and separately from the “student” (Step 2). This is in contrast with [12], where everything is learned end-to-end, but in line with [13], where separated learning steps proved to be more effective.

Step 1. The RGB and depth streams are trained separately, which is common practice in two-stream architectures. Both depth and appearance streams are trained by minimizing the cross-entropy loss, after being initialized with a pre-trained ImageNet model for all experiments as common practice [12; 13; 21]. We test both streams individually and in a two-stream setup, where the final prediction results from the average of the two streams’ logits. We found that fine-tuning the two-stream model does not increase performance consistently. This step can also be regarded as training the teacher network - depth stream - for the next step (see Fig. 4.1, top).

Step 2. The depth stream E_d , trained in the previous step, is now frozen, in order to provide a stable target for the hallucination network (generator) H , which plays the adversarial game with a discriminator D (see Fig. 4.1, bottom). The primary task of the discriminator D is to distinguish between the features F_H generated by the hallucination network H and F_d generated by the depth network E_d . However, as already mentioned, the discriminator is also assigned an auxiliary discriminative task, as detailed in the following.

The architecture of the networks E_d and H is a mix of 2D and 3D convolutions

that process a set of frames, and output a feature vector for every frame t of the input volume, *i.e.* F_H^t and F_d^t . This means that each frame have a corresponding feature vector, and these may vary even if sampled from the same video, depending on its dynamics and its position t in the input volume. For example, the first frame (and feature vector) of a clip belonging to the action "shaking hand" might be very different from its the middle frame, but similar to the first frame of a clip belonging to the class "pushing other person". This increases the complexity for the generator, that have not only to generate features similar to F_d , but also to match the order in which they are generated. Namely, F_H^t should be similar to F_d^t , for every frame t of the input volume. We ease this issue by providing as input to D the one-hot encoding vector of the relative index t , which we denote y^t , concatenated with the respective feature vector, which relates to the CGAN mechanism [41].

In standard adversarial training, the discriminator D would try to assign the binary label true/fake to the feature vector coming from the two different streams. However, we found that features F_H generated with this mechanism, although being very well mixed and indistinguishable from F_d , were struggling to achieve good accuracy for the classification tasks, *i.e.* were lacking discriminative power. For this reason we assign to the discriminator the auxiliary task of classifying feature vectors with their correct class.

The adversarial learning problem is formalized as follows. Consider the RGB-D dataset (X_{rgb}, X_d, Y) where $x_{rgb}^t, x_d^t \sim (X_{rgb}, X_d)$ are time aligned RGB and depth frames, $y \sim Y$, is the C -dimensional one-hot encoding of the class label, and C is the number of classes for the problem at hand.

Now, let the *extended label vector* with $C + 1$ components (classes):

$$\hat{y} = \begin{cases} [\text{zeros}(C) \parallel 1], & \text{for } x_{rgb} \\ [y_i \parallel 0] & \text{for } x_d \end{cases} \quad (4.1)$$

where $\text{zeros}(C)$ represents a vector of zeros of dimension C , and \parallel is the concatenation operator. Using this label vector instead of the classical 0/1 (real/-generated) binary label in the discriminator encourages feature representations F_H learned by H to encode not only depth (monocular) features, but also to be discriminative. This is possibly why the hallucination network often recovers the accuracy of the teacher and sometimes performs even better, as further discussed in the experimental section. In summary, we want F_H features to be as discriminant as real ones: the adversarial procedure produces fake features which not only are classified as real by the discriminator, but are also assigned to the correct class.

Based on the above definitions, we define the following minimax game:

$$\begin{aligned} \min_{\theta_D} \max_{\theta_H} \ell = & \mathbb{E}_{(x_i, y_i) \sim (X_{rgb}, Y)} \mathcal{L}(D(H(x_i) \parallel y^t), \hat{y}_i) \\ & + \mathbb{E}_{(x_i, y_i) \sim (X_d, Y)} \mathcal{L}(D(E_d(x_i) \parallel y^t), \hat{y}_i) \end{aligned} \quad (4.2)$$

where θ_H and θ_D indicate the parameters of the hallucination stream H and of the discriminator D , \parallel denotes a concatenation operation and \mathcal{L} is the softmax cross-entropy function. Eq. 4.2 is optimized via the well known "label flipping hack" [84], which makes the loss function easier to minimize in practice.

4.2.2 Architectural details

All three networks (depth stream - E_d , RGB stream - E_{rgb} , and hallucination stream H) are modified Resnet-50 [75] augmented with *temporal convolutions*

and endowed with a final *bottleneck layer*. The hallucination networks H are initialized with the respective depth stream weights E_d , following the findings of [12] for object detection, and [13] for action recognition.

Temporal convolutions

1D temporal convolutions are inserted in the second residual unit of each ResNet layer as illustrated in Fig. 4.2, following the recent work of Feichtenhofer *et al.* [1]. For layer l , the weights $W_l \in \mathbb{R}^{1 \times 1 \times 3 \times C_l \times C_l}$ are convolutional filters initialized as identity mappings at feature level, and centered in time, where C_l is the number of channels in layer l . More in detail, all the $[1 \times 1 \times 3]$ temporal kernels contained in W_l are initialized as $[0, 1, 0]$, *i.e.* only the information of the central frame is used at the beginning. This progressively changes as training goes on. Very recently, in [58], the authors explored various network configurations using temporal convolutions, comparing different combinations for the task of video classification. This work suggests that decoupling 3D convolutions into 2D (spatial) and 1D (temporal) filters is the best setup in action recognition tasks, producing best accuracies. The intuition for the latter setup is that factorizing spatial and temporal convolutions in two consecutive convolutional layers eases training of the spatial and temporal tasks (also in line with [85]).

Bottleneck

Generating, encoding, or aligning high dimensional feature vectors via adversarial training is often a difficult task, due to the inherent instability of the saddle point defined by the GAN minimax game. For this reason, [36] proposes to align a lower dimensional vector, obtained by adding a *bottleneck layer* to standard architectures. This usually does not affect performances of baseline models.

Indeed, the size of the last ResNet-50 layer (before the logits) is $[7, 7, 2048]$,

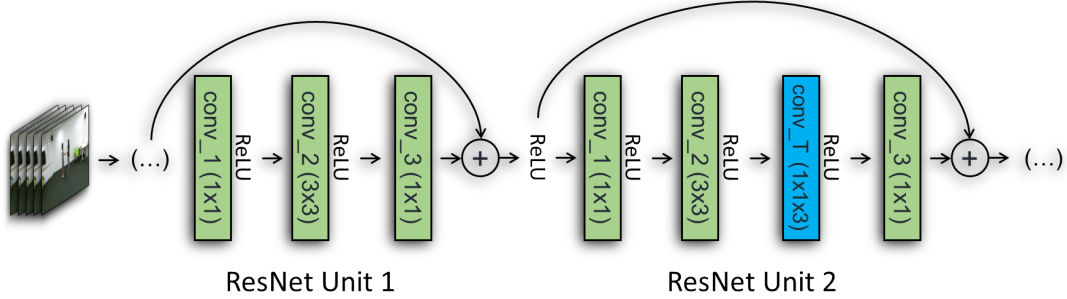


Figure 4.2: Detail of the ResNet residual unit with temporal convolutions (blue block).

or simply [2048] after pooling. For this reason, we further modify the ResNet-50 by adding either i) an additional convolutional layer, whose weights $W_b \in \mathbb{R}^{7 \times 7 \times 2048 \times 128}$, applied with no padding, reduce the dimensionality to 128; or ii) a simple 128-dim fully connected layer after pooling. In Section 4.3.2 we further explore the choice of the bottleneck.

Input

For the task of action recognition, the input to the encoder networks E and H is five 3-channel frames, uniformly sampled from each video sequence, which motivates temporal convolution. Instead, for the task of object classification (from single images), no temporal kernels are added to the architecture. We try different encodings for the depth channel: for the task of action recognition they are encoded into color images using a jet colormap, as in [77]; for the object recognition task, HHA encoding [86] is already provided in the dataset considered.

Discriminator

The discriminator used to play the adversarial game has different architectures depending on the task. These architectures follow the empirically validated common practices in the adversarial learning literature, more specifically to what is

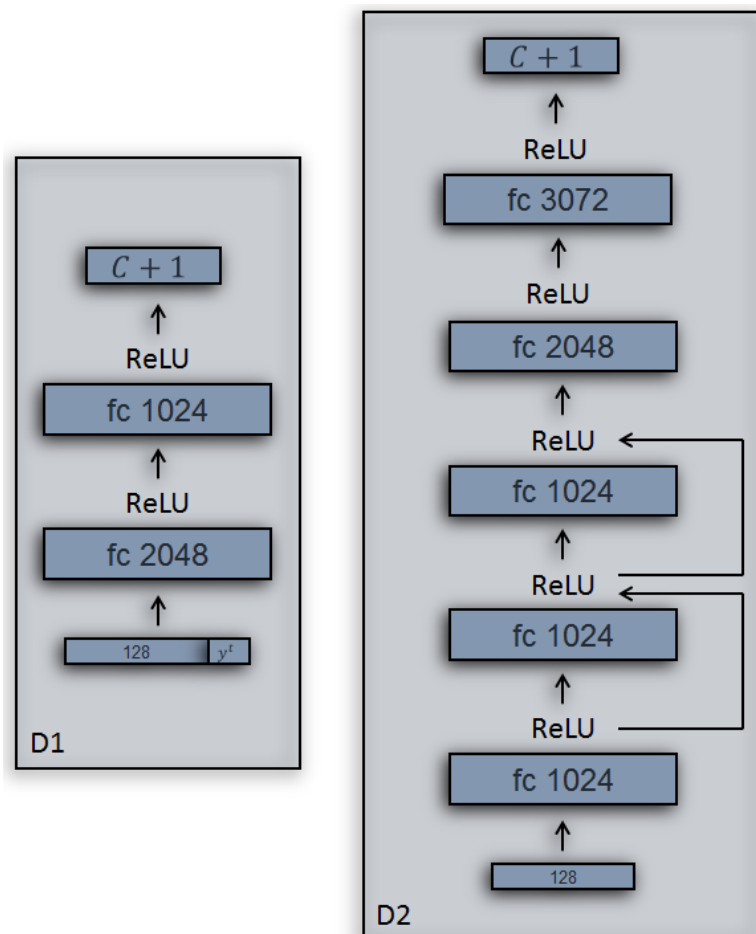


Figure 4.3: Architectures for the discriminators used for the two different tasks. Left: D1 for object recognition. Right: D2 for action recognition.

described in [36]. Its basic structure is that of a multilayer perceptron, stacking fully connected (fc) layers only, since it takes a vector as input (bottleneck features, possibly concatenated with temporal ordering for tasks involving time). For the task of action recognition, the structure is quite shallow, consisting in $D1=[fc(2048), fc(1024), fc(C+1)]$. For the task of object classification the structure is instead more complex $D2=[fc(1024), fc(1024), fc(1024), fc(2048), fc(3072), fc(C+1)]$, with skip connections in the lower layers. Being the former discriminator quite deep, residual connections were inserted in order to allow gradient

to flow through the underlying hallucination stream. Details of the architectures are sketched in Fig. 4.3.

4.3 Experiments

4.3.1 Datasets

We evaluate the performance of our method on one object classification and two video action classification datasets. For both tasks the model is initialized with ImageNet pretrained weights. For the experiments on the smaller action recognition dataset NW-UCLA, we fine-tune the model starting from the RGB and depth streams trained on the larger NTU RGB+D dataset.

NTU RGB+D [67]. We follow the two evaluation protocols originally proposed in [67], which are cross-subject and cross-view. As in the original paper, we use about 5% of the training data as validation set for both protocols. The masked depth maps are converted to a three channel map via a jet mapping, as in [77].

Northwestern-UCLA [66]. This action recognition dataset provides RGB, depth and skeleton sequences for 1475 samples. It features 10 subjects performing 10 actions captured in 3 different views.

NYUD (RGB-D) This dataset of objects (see examples in Fig. 4.4) is gathered by cropping out tight bounding boxes around instances of 19 object classes present in the NYUD [83] dataset. It comprises 2,186 paired labeled training images and 2,401 test images (RGB-D). Depth images are HHA-encoded [86]. This version of the dataset was proposed in [12] but also used in [36; 87; 88] for the task of modality adaptation, in the framework of domain adaptation (train on



Figure 4.4: Examples of RGB and depth frames from the NYUD (RGB-D) dataset.

one modality, adapt and test the model on the other modality). The task here is object classification, training on both modalities and testing on RGB only.

4.3.2 Ablation Study

The ablation study is performed on part of the NTU RGB+D dataset, designated as mini-NTU, which consists of random samples from the training set, considering approximately a third of the original dataset size. The test set is still the same as used in the other experiments and defined originally in [67].

We study how the hallucination network performance is affected by (1) feeding different types of input to the discriminator, and (2) having the discriminator to perform different tasks.

Bottleneck size

The discriminator receives as input the feature vector F_H or F_d from either the hallucination or the depth stream, respectively, along with the frame index label

4.3 Experiments

Network	Dataset	X-Subject
Depth stream, normal - (target)	NTU	70.53%
Hall. net, $F_x \in \mathbb{R}^{2048}$	NTU	54.25%
Hall. net, $F_x \in \mathbb{R}^{2048}$	NTU-mini	60.95%
Depth stream, w/ bottleneck - (target)	NTU	69.13%
Hall. net, $F_x \in \mathbb{R}^{128}$	NTU	72.14%

Table 4.1: Ablation Study - Bottleneck size. Hallucination network underperforming with $F_x \in \mathbb{R}^{2048}$.

y^t . It is known that a too big feature vector may cause the GAN training to underperform [36], which we also observe in our experiments, reported in Table 4.1.

We first trained our depth network without bottleneck on the full NTU dataset, reaching 70.53% accuracy. This network is then used as target to learn the hallucination model. We observed that the hallucination model trained without bottleneck, *i.e.*, the input to the discriminator is the 2048-dimensional feature vector, is far from recovering the performance of the target (reaching only 54.25%), even if the training space is reduced to the NTU-mini dataset (60.95%).

We then train a network with a 128-dimensional bottleneck (69.13%), initialized with the previous depth stream, except for the bottleneck that is randomly initialized with the MSRA initialization [89]. The hallucination model that learns using the bottleneck feature vector is able not only to recover, but to surpass the performance of the depth stream, reaching 72.14% accuracy. We observed this behaviour in other experiments along the paper, and we comment that later in Section 4.3.4.

Bottleneck implementation

In Table 4.2 we investigate different ways to decrease the size of F_x from \mathbb{R}^{2048} to \mathbb{R}^{128} , as suggested in [36]. After the last feature map, which is of dimension

Depth stream - versions	X -Subject	X -View
Depth stream wo/ bottleneck	63.95%	62.70%
One conv	55.64%	57.91%
Spatial conv + 1D conv	53.21%	52.58%
pool + conv	61.41%	63.15%

Table 4.2: Ablation Study - Investigating different bottleneck implementations. The Table reports Hallucination network performances on NTU-mini.

7*7*2048, we tested the three following ways:

- convolution of [128,7,7] to 1*1*128,
- spatial convolution of [7,7] to 1*1*2048 followed by 1D convolution to 1*1*128, and
- pooling layer to 1*1*2048 followed by 1D convolution to 1*1*128

Even though the depth stream is just trained on the NTU-mini (63.95% for cross subject, and 62.70% for cross view), the hallucination stream that implements the pool+conv bottleneck is able to recover almost completely (61.41% for cross subject), or even surpass (63.15% for cross view), the original depth stream performance. This was the architectural choice we used in the rest of the experiments.

Discriminator: inputs and tasks

In this section, we explore whether the task assigned to the discriminator influences the hallucination performance. As introduced in Section 4.2, our hypothesis is that the generator has the difficult task of generating features that not only correspond to depth features, but also need to be temporally paired with these. We solve this by introducing the additional information of the frame index y^t , which specifies the desired alignment. Table 4.3 shows results regarding the (1)

Input	Task	X-Subject
Teacher network (pool + conv, Table 4.2)	-	61.41%
$F(x)$	0/1 classification	1.81%
$F(x)$	\hat{y} classification	59.87%
$F(x) y_t$	\hat{y} classification	63.03%

Table 4.3: Ablation Study - Investigating different inputs and tasks for the discriminator. The Table reports Hallucination network performances (NTU-mini).

traditional binary task of a GAN generator having as input the feature bottleneck, (2) the \hat{y} classification task having the same input as before, and (3) the proposed approach. The traditional binary task (1) converges to a perfect equilibrium, but the hallucination stream’s accuracy is close to random chance, meaning that the learned features are not discriminant at all. The second approach (2) is able to learn discriminative features, but the addition of the frame order supervision y^t (3) shows an increase in performance. It is reasonable that this mechanism produces maximized gains on more challenging and diverse datasets, as the full NTU dataset, or in fully 3d-convolutional architectures such as I3D [74], due to the higher dependence on temporal convolutions.

4.3.3 Action recognition performance and comparisons

Table 4.4 compares performances of different methods in the literature, across the two datasets for action recognition - two protocols for the NTU RGB+D and the NW-UCLA. The standard performance measure used for this task and datasets is classification accuracy, estimated according to the protocols, training and testing splits defined in the respective works. The first part of the table (indicated by \times symbol) refers to unsupervised methods, which achieve surprisingly high results even without relying on labels in learning representations.

The second part refers to supervised methods (indicated by \triangle), divided according to the modalities used for training and testing. Here, we report the performance of the separate RGB and depth (with and without bottleneck) streams trained in step 1 (rows #7 and #8). The small increase in performance is probably due to the extra training steps with small learning rate, after initialized with the bottleneck version trained on the mini-NTU (used for the ablation study). Importantly, the depth stream with bottleneck represents the teacher network used for the hallucination learning. We expect our final model to perform better than the one trained on RGB only, whose accuracy constitutes a lower bound for the usefulness of our hallucination model. The values reported for our step 1 models for the NW-UCLA dataset, *i.e.* the RGB and depth streams, refer to the fine-tuning of our NTU model. In contrast with [13], and for clearer analysis, the two-stream setup is always not finetuned. Its accuracy represents an upper bound for the final model, which will not rely on depth data at test time. We have experimented training using pre-trained ImageNet weights instead of the NTU, but it led to lower accuracy.

The last part of the table (indicated by \square) reports the performance of methods in the privileged information framework, thus directly comparable to ours. The performance values that refer to the Hoffman *et al.* method [12] (row #20 of Table 4.4) are taken from the implementation and experiments in [13]. Row #21 refers to the method by Luo and colleagues [21], that uses 6 modalities at training time (RGB, depth, optical flow, and three different encoding methods for skeleton data), and RGB only at test time. Step 3 and 4 of [13] (row #22 and #23) refer to the two-stream model after the hallucination learning, and its fine-tuning, respectively. We note that, for simplicity, the results of ADMD Two-Stream models are merely the outcome of the average of the two streams' logits, and are not subject to any fine-tuning, which means that they are directly

comparable with row #22. In addition, results of row #24 correspond to the hallucination stream only.

We note that the hallucination stream (row #24) manages to recover and surpass the depth teacher stream (row #8) for the NW-UCLA dataset (83.94% compared to 71.09%), while for the NTU p1 (67.57%) and p2 (71.80%) protocols is around 4% below the respective teacher (71.87% and 75.32%). Nevertheless, when combined with the RGB stream, it performs better (NTU p2 - 81.50%) or comparable (NTU p1 - 73.11%) to the fine-tuned model presented in [13]. Since the RGB stream is performing equally well in this work and in [13], we can conclude that the gains in performance are due to better hallucination features.

4.3 Experiments

#	Method	Test Mods.	NTU (p1)	NTU (p2)	NW-UCLA	
1	Luo [78]	Depth	66.2%	-	-	
2	Luo [78]	RGB	56.0%	-	-	×
3	Rahmani [90]	RGB	-	-	78.1%	
4	HOG-2 [79]	Depth	32.4%	22.3%	-	
5	Action Tube [91]	RGB	-	-	61.5%	
6	Depth stream [13]	Depth	70.44%	75.16%	72.38%	
7	ADMD - Depth stream	Depth	70.53%	76.47%	-	
8	ADMD - Depth stream w/ bott.	Depth	71.87%	75.32%	71.09%	
9	RGB stream - [13]	RGB	66.52%	80.01%	85.22%	△
10	ADMD - RGB stream	RGB	67.95%	80.01%	85.87%	
11	Deep RNN [67]	Joints	56.3%	64.1%	-	
12	Deep LSTM [67]	Joints	60.7%	67.3%	-	
13	Sharoudy [67]	Joints	62.93%	70.27%	-	
14	Kim [80]	Joints	74.3%	83.1%	-	
15	Sharoudy [7]	RGB+D	74.86%	-	-	
16	Liu [8]	RGB+D	77.5%	84.5%	-	
17	Rahmani [92]	Depth+ Joints	75.2	83.1	-	
18	Two-stream, step 2 [13]	RGB+D	79.73%	81.43%	88.87%	
19	ADMD - Two-stream (no finetune)	RGB+D	77.74%	85.49%	89.93%	
20	Hoffman <i>et al.</i> [12]	RGB	64.64%	-	83.30%	
21	Luo <i>et al.</i> [21]	RGB	89.50%	-	-	
22	Hallucination model, step 3 [13]	RGB	71.93%	74.10%	76.30%	□
23	Hallucination model, step 4 [13]	RGB	73.42%	77.21%	86.72%	
24	ADMD - Hall. stream alone	RGB	67.57%	71.80%	83.94%	
25	ADMD - Hall. two-stream model	RGB	73.11%	81.50%	91.64%	

Table 4.4: Classification accuracies and comparisons with the state of the art for video action recognition. Performances referred to the several steps of our approach (ours) are highlighted in bold. × refers to comparisons with unsupervised learning methods. △ refers to supervised methods: here train and test modalities coincide. □ refers to privileged information methods: here training exploits RGB+D data, while test relies on RGB data only. The 4th column refers to cross-subject and the 5th to the cross-view evaluation protocols on the NTU dataset. The results reported on the other two datasets are for the cross-view protocol.

4.3 Experiments

Method	Trained on	Tested on	Accuracy
Depth alone	Depth	Depth	40.19%
RGB alone	RGB	RGB	52.90%
RGB ensemble	RGB	RGB	54.14%
Two-stream (average logits)	RGB+D	RGB+D	57.39%
Two-stream after finetuning	RGB+D	RGB+D	58.73%
ModDrop [23] (finetuned from Two-stream)	RGB+D	RGB+D	58.93%
ModDrop [23]	RGB+D	RGB+blankD	47.86%
ModDrop [23]	RGB+D	RGB	53.73%
Autoencoder	RGB+D	RGB	50.52%
FCRN [93] depth estimation	RGB+D	RGB	50.23%
Hallucination model [11]	RGB+D	RGB	55.94%
Ours (naive adversarial)	RGB+D	RGB	50.81%
Ours (ADMD)	RGB+D	RGB	57.52%

Table 4.5: Object Recognition

4.3.4 Object recognition performance and comparisons

Table 4.5 illustrates the main results obtained for NYUD dataset for the object recognition task.

As opposed to action recognition, depth information is often noisy here (cfr. Fig. 4.4 - chair and lamp), probably due to the small resolution of the bounding box crops. Depth alone is in fact performing worse than RGB alone (more than 10% gap). Still, the amount of *complementary information* carried by the two modalities is able, when fused in the two-stream model, to boost recognition accuracy by more than 5 percentage points, despite the poor depth performance (RGB→52.90%, Depth→40.19% ⇒ two-stream→57.39%).

It is well established that ensemble methods tend to outperform their single-model counterparts: an ensemble of two CNNs, each trained started from a different initialization, outperforms either independent model [94]. Since, in principle, the proposed ADMD strategy is the combination of an RGB model trained using a

standard supervised approach and *another* adversarially trained RGB model, we additionally compare our approach to an ensemble of RGB classifiers (third line of Table 4.5). Interestingly, despite starting from a two relatively high single-stream performances, the fusion process of two RGB networks only marginally increases the final accuracy (RGB1→53.19%, RGB2→52.60% ⇒ Ensemble→54.14%).

As noticed for the task of action recognition, we found that fine-tuning the fused streams does not always bring significant improvements, as opposed to [13], were the architecture features cross-stream multiplier connections, which need to be trained in an further step. Fine-tuning with the strategy proposed by Neverova *et al.* [23] looks slightly more effective, since ModDrop introduces a light dropout at the input layers, both on the images and on the whole modalities. The resulting model is tested in both the original setup proposed in [23], namely by blanking out the depth stream, and by simply using RGB predictions. The latter scheme slightly improves the performance of the RGB stream, possibly thanks to dropout. However, although the model shows more robustness to missing depth at test time, it clearly fails to extract any monocular depth cue.

Another interesting comparison we perform is the following: we train a cross-modal autoencoder with an L2 loss in order to reconstruct depth maps from RGB. The encoder-decoder architecture consists in the very same RGB ResNet-50 for the encoder, and in 5 stacked deconvolutional blocks intertwined with batch-norm layers for the decoder. At test time, when depth is not available, we provide RGB frames to the autoencoder, which reconstructs the missing modality to feed the corresponding branch of the two-stream architecture. The performance of this setup is quite poor. We observe that the autoencoder easily overfits the training set, generating high quality depth maps for the training set, while it performs very poorly for the test set. Similarly, we reconstruct depth by means of FCRN [93], a state-of-the-art depth estimator trained on the entire NYUD dataset. Again,

performance is quite poor, since depth estimated by FCRN misses many fine details needed for object classification. This suggest that, for the recognition task, *hallucinating task specific features is more effective than estimating depth*.

This claim is again confirmed with the result for the Hallucination model proposed in [13], adapted in this case for object recognition (Table 5, 3rd to last row). This method outperforms both the RGB stream and the RGB ensemble, confirming the value of hallucinating depth. It also outperforms the other baselines that use RGB only at test time (3rd section of Table 5). In particular, it performs considerably better than FCRN depth estimation, which indicates again that depth feature hallucination is more effective than predicting depth maps at pixel level. More importantly, we can directly compare it with ADMD proposed in this paper (55.94% vs 57.52%), concluding that, similarly to action recognition experiments, the adversarial approach performs better.

Eventually, we tested our adversarial scheme in two different setups: i) the naive setup where the discriminator D is assigned the binary task only, and ii) the ADMD setup, where the discriminator is also assigned the classification task. While the former performs as the autoencoder, the latter is able to fully recover the accuracy of the Two-stream model, being only slightly below that of the fine-tuned model.

4.3.5 Inference with noisy depth

In real test scenarios, it is often the case that we can only access *noisy* depth data. In this section, we address two questions: i) how much such noisy data can degrade the performance of a multimodal setup? ii) At which level of noise does it become favorable to hallucinate the depth modality with respect to using the teacher model (Two-stream) with noisy depth data?

The depth sensor used in the NTU dataset (Kinect), is an IR emitter coupled

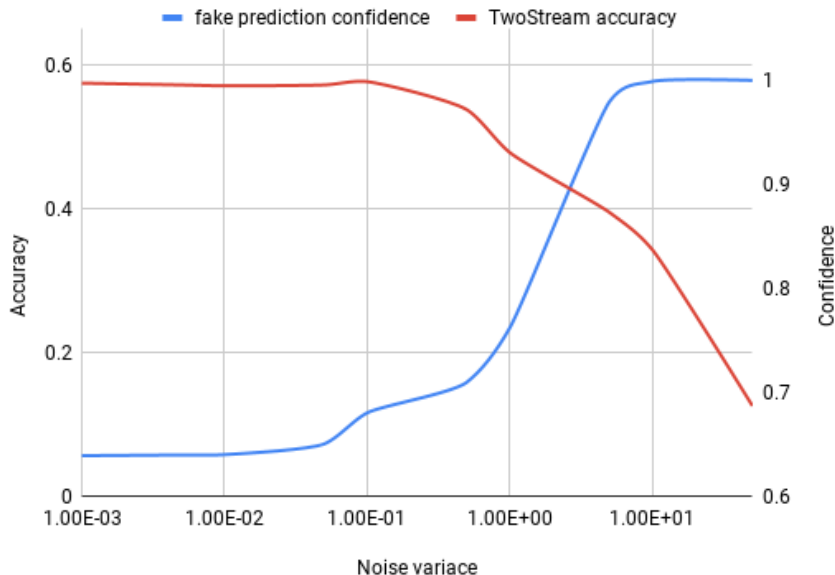
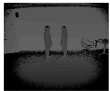
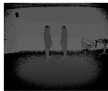
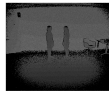
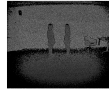
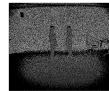
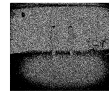



Figure 4.5: Discriminator confidence at predicting 'fake' label as a function of noise in the depth frames. The more corrupted the frame, the more confident D , and the lower the accuracy of the Two-stream model (NYUD dataset).

with an IR camera, and has very complex noise characterization comprising at least 6 different sources [81]. It is beyond the scope of this work to investigate noise models affecting the depth channel, so, for our analysis we choose the most influencing one, i.e., multiplicative speckle noise. Hence, we inject Gaussian noise in the depth images I in order to simulate speckle noise: $I = I * n$, $n \sim \mathcal{N}(1, \sigma)$. Table 4.6 shows how performances of our Two-stream network degrade when depth is corrupted with such Gaussian noise with increasing variance (NTU cross-view protocol and NYUD). Results show that accuracy significantly decreases with respect to the one guaranteed by our hallucination model (81.50% - row #25) in Table 4.4, even with low noise variance of $\sigma^2=10^{-1}$. For the task of object recognition, we can see that ModDrop [23] is slightly more resilient to depth corruption than the simple Two-stream, since fine-tuned with noise (dropout) in the input layer.

NTU RGB+D action dataset - ADMD performance is 81.50%.							
							
σ^2	<i>no noise</i>	10^{-3}	10^{-2}	10^{-1}	10^0	10^1	<i>void</i>
Two-stream	85.49%	85.52%	82.05%	68.99%	2.16%	3.35%	8.55%



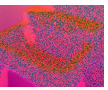
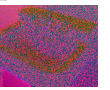

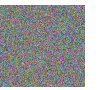

NYUD object dataset - ADMD performance is 57.52%.							
							
σ^2	<i>no noise</i>	10^{-3}	10^{-2}	10^{-1}	10^0	10^1	<i>void</i>
Two-stream	58.73%	58.68%	58.23%	57.18%	48.27%	28.40%	47.44%
ModDrop [23]	58.93%	58.89%	58.56%	57.49%	48.90%	25.95%	47.86%

Table 4.6: Accuracy values for the two-stream model trained on RGB and depth, and tested with RGB and noisy depth data.

This experiment shows, in conclusion, that *ADMD is able not only deal with a missing modality, but also with a noisy one*. In an online scenario, the discriminator D , trained in step 2, can give an indication on when to operatively switch from Two-stream to ADMD, that is, when to substitute the depth branch with the hallucination. When training reaches equilibrium, D is maximally fooled by the features generated by H , and cannot distinguish them from those encoded by E_d . In practice, this means that the predicted probability for the fake class (last class in \hat{y} , eq. 4.1) is $p(\hat{y} = C + 1) \approx .5$ on average. However, when features computed from corrupted depth start to flow inside D , its prediction for the fake class starts to be more and more confident. Figure 4.5 plots the behavior of D as noise increases, together with accuracy of the Two-stream model. There is a clear turning point in both accuracy and confidence, which can be employed in practice to decide when to switch from E_d to H *i.e.* when to drop depth as a modality and start using monocular depth features extracted from RGB.

4.3.6 Discussion

Some interesting points arise from the analysis of our findings, which we summarize in the following.

1. RGB and depth actually carry complementary information. As a matter of fact, the Two-stream setup always provides a surprisingly better accuracy than the two streams alone. As additional evidence, a multimodal ensemble (*i.e.* the Two-stream) performs better than a mono-modal ensemble (Table 4.5), despite the lower accuracy of one of its single-stream components (either depth or RGB, depending on task and dataset).

2. There is (monocular) depth information in RGB images. This is evident from the fact that the hallucination stream often recovers and sometimes surpasses the accuracy of its depth-based teacher network. Besides, fusing hallucination and RGB streams always bring the benefits, as fusing RGB and Depth.

3. Standard supervised learning has limitations in extracting information. In fact, given the evidence that there is depth information to exploit

in RGB images, minimizing cross-entropy loss is not enough to fully extract it. For that we need a student-teacher adversarial framework. This has an interesting parallel in *adversarial network compression* [33], where the performance of a fully supervised small network can be boosted by adversarial training against a high-capacity (and better performing) teacher net. In [33], it is also observed that the student can surpass the teacher in some occasions.

4. Adversarial training alone only is not enough. The naive discriminator trained for the binary task (real/generated) is not sufficient to force the hallucination network to produce discriminative features. The auxiliary discriminative

task is necessary to extract monocular depth cues which are also discriminative for a given task (on the other hand, the auxiliary task only is not enough, as suggested by the performance of the RGB ensemble).

5. Hallucinating task-specific depth features is more effective than estimating full depth images. Not only estimated depth is often missing details needed for classification, but also its estimation is driven by mere reconstruction objectives. On the contrary, feature hallucination addresses a specific classification task and requires estimating low dimensional vectors instead of images.

4.4 Summary

In this work, we have introduced a novel technique to exploit additional information, in the form of depth images at training time, to improve RGB only models at test time. This is done by adversarially training a hallucination network which learns from a teacher depth stream how to encode monocular depth features from RGB frames. The proposed approach outperforms previous ones in the privileged information scenario in the tasks of object classification and action recognition on three different datasets. Additionally, the hallucination framework is shown to be very effective in cases where depth is noisy.

Chapter 5

Distillation Multiple Choice Learning

5.1 Introduction

Humans perceive the environment by processing a combination of modalities. Such modalities can include audio, touch and sight, with each modality being distinct from and complementary to the others. Deep learning methods may likewise benefit from multimodal data. In this work, we explore how to leverage the complementary nature of multimodal data at training time, in order to learn a better classifier that takes as input only RGB data for inference.

One popular way to train multimodal deep learning models is to train one network per modality, and mean pool all the network predictions for inference. This is a sub-optimal use of multimodal training data, as modalities do not exchange information while training. For example, considering the task of action recognition, some actions are easier to discriminate using certain modalities over others: the action “open a box” may be confused with “fold paper” when solely relying on the RGB modality, while it is easily classified using depth data [2].

This suggests that an ensemble of networks could use multimodal data in a more efficient way, *e.g.* by encouraging the network trained with a given modality to focus on the set of classes or samples that maximizes its discriminative power. In this case, each network is referred to as a *specialist network*, as it only sees part of the dataset and specializes in that part of the problem. Assuming that all modalities are available, the ensemble should be able to fuse the specialists' predictions and produce a single output.

The problem of multimodal fusion becomes more challenging when some modalities are not available at test time. This is particularly problematic if the training process encourages the specialization of each modality network of the ensemble. In this case, a missing modality means that the ensemble loses the ability to correctly classify the corresponding part of the task assigned to this specialist.

In this work, we propose a novel method that is at the intersection of MCL framework and Knowledge Distillation [18; 95], called Distillation Multiple Choice Learning (DMCL). DMCL addresses two practical dimensions of multimodal learning: a) leveraging the complementarity of multiple modalities, and b) being robust to missing modalities at test-time.

We take inspiration from the Multiple Choice Learning (MCL) framework, which is a popular way to train an ensemble of RGB networks [70; 71; 72]. This method chooses the best performing network of the ensemble to backpropagate the task loss. However, extending it to multiple modalities is not straightforward. Networks that are trained using different modalities learn at different speeds. Consequently, the network that learns faster in the beginning of the training dominates the traditional MCL algorithm, and is encouraged to remain dominant during training. We extend MCL to a) address such challenges associated with multimodal data, and b) deal with modalities that may be missing at test time.

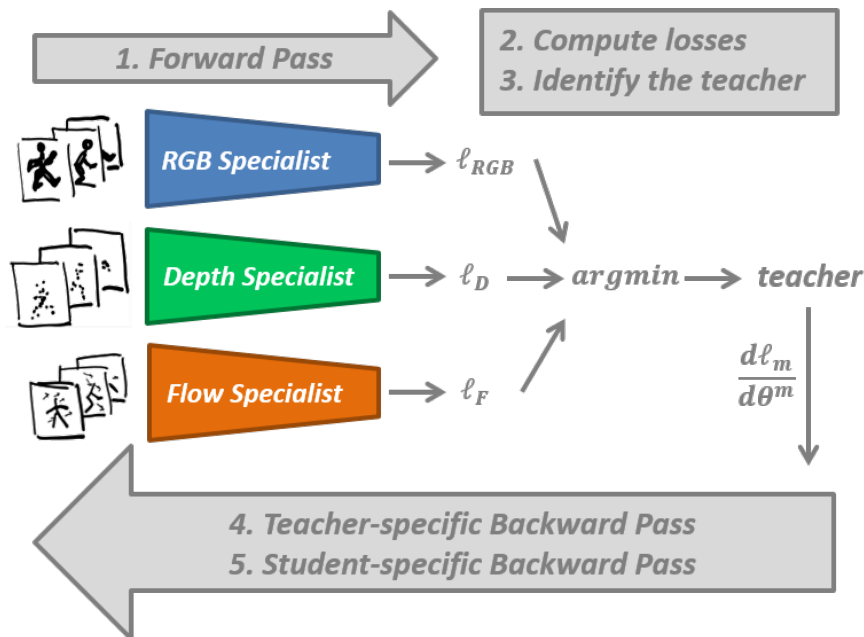


Figure 5.1: **Distillation Multiple Choice Learning (DMCL)** allows multiple modalities to cooperate and strengthen one another. For each training sample, the modality specialist m that achieves the lowest loss ℓ distills knowledge to strengthen other modality specialists. At test time, any subset of available modalities can be used by DMCL to make predictions.

The case of a missing modality at test time is related to learning using Privileged Information [11] and Knowledge Distillation [95]. This type of approaches is usually structured as a two-step process: training a teacher network, and then using its knowledge to train a student network. The teacher network has usually a larger capacity, or has access to more data than the student. For example, consider the problem of learning a model for action recognition using a multimodal dataset composed of RGB, depth, and optical flow videos. In practice, it is reasonable to assume that only RGB modality is present for test inference: depth sensors are expensive and optical-flow computation incurs runtime cost that may not meet real-time budget. At the same time, depth and optical flow can provide valuable information on the samples or classes that it perform better, and that could be distilled to the RGB network [50] [96].

We build on these ideas to develop a model that learns from multimodal data, exploiting the strength of each modality in a cooperative setting as the training proceeds. This is summarized in Figure 5.1. Furthermore, our proposed model is able to account for one or more missing modalities at test time. The code of our Tensorflow [97] implementation will be made publicly available at <https://github.com/ncgarcia/DMCL>. Our main contributions are:

- We conduct a deep evaluation of the MCL framework in the context of multimodal learning and give insights on how multiple modalities behave in such ensemble learning methods.
- We propose DMCL, a MCL framework designed for multimodal data where modalities cooperate to strengthen one another. Moreover, DMCL is able to account for missing modalities at test time.
- We present competitive to or state-of-art results for multimodal action recognition using privileged information on three video action recognition benchmark datasets.

This work is at the intersection of three topics: generalized distillation [18], video action recognition, and ensemble learning. These topics are discussed in Chapter 2, and the comparison with the several MCL algorithms is discussed in detail later in section 5.2.

5.2 Model

Our goal is to learn an ensemble of multimodal specialists that leverages the specific strengths of each modality to the benefit of the ensemble. This is accomplished by setting a cooperative learning strategy where stronger networks teach weaker networks through knowledge distillation, depicted in Figure 5.2. For a

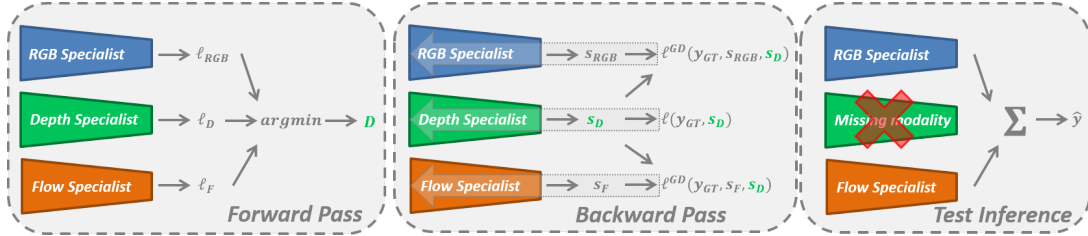


Figure 5.2: **Distillation Multiple Choice Learning (DMCL)** In the Forward Pass, we calculate the classification cross-entropy losses ℓ for each modality and identify the teacher network - in this case, the Depth network. In the Backward Pass, we compute the soft targets of the teacher, s_D , and use them as an extra supervision signal for the student networks. The loss for the student networks ℓ^{GD} refers to the Generalized Distillation loss, defined on Eq. 5.3. The loss for the teacher network D uses the normal logits, *i.e.* soft targets with temperature $T = 1$. At test time, we are able to cope with missing modalities. The final prediction is obtained by averaging the predictions of the available modalities.

given data point at training time, we identify the best-performing network as a teacher for the remaining networks in the ensemble.

5.2.1 Distillation Multiple Choice Learning

Algorithm 1 describes our method DMCL. Let $\mathbb{D} = \{(x_i, y_i)\}^N$ be a multimodal dataset having N training samples. Each sample x_i represents the data for the M modalities available, $x_i = \{x_i^1, \dots, x_i^M\}$, and y_i represents its label.

Our ensemble is composed of a set of M networks f , each using as input a different modality $f^1(x_i^1), \dots, f^M(x_i^M)$. The MCL algorithm maximizes the ensemble accuracy, often referred to as oracle accuracy. The oracle accuracy assumes that we can choose the correct prediction out of the set of outputs produced by each network. This translates to the minimization of the ensemble loss L , which is defined as the lowest of the individual networks' loss values, calculated for a given data point.

Formally, MCL minimizes the ensemble loss L with respect to a specific task

loss $\ell(y_i, \hat{y}_i)$ for each network prediction $\hat{y}_i = f^m(x_i^m)$ for a specific modality m :

$$L(\mathbb{D}) = \sum_{i=1}^N \min_{m \in \{1, \dots, M\}} \ell(y_i, f^m(x_i^m)). \quad (5.1)$$

In practice, we get all the networks’ predictions for each sample of the batch. We calculate the loss $\ell_{\text{criterion}}$ for each network and sample (line 5, Algorithm 1). In this case, $\ell_{\text{criterion}}$ corresponds to the standard cross-entropy loss. The network with the lowest loss value is designated as the winner network, and the others are set to be loser networks. The loss and gradient updates for a network depend on whether it is a winner or loser network (lines 10-14, Algorithm 1). In our proposed privileged-information formulation, we view the winner network as a teacher, and the loser networks as students.

DMCL function of `update_winner` and `update_losers` of Algorithm 1 define how the teacher network distills information to the student networks, strengthening them. DMCL updates teachers with respect to the cross-entropy training loss computed using the ground-truth label. The loser networks are updated using a distillation loss, which aims to transfer knowledge from the winner network.

Knowledge Distillation

Matching the students’ with the teachers’ soft targets is one way to transfer knowledge from one model to another. Soft targets are a smoothed probability distribution than the originally produced by the modality network f^m :

$$s_i^m = \sigma(f_i^m(x_i^m)/T), \quad (5.2)$$

where σ is the softmax function, f_i^m are the logits, and T is a scalar value. The default temperature T value is set to 1 for models that do not incorporate distillation. Setting T to a higher value produces a smoother probability distri-

bution that reveals valuable information about the relative probabilities between classes, which has shown to improve knowledge transfer and generalization of the new model. In practice, very small probability values become more evident with higher temperatures.

The Generalized Distillation (GD) [18] method consists of three sequential steps: (1) learn the teacher network; (2) fix the teacher and compute the soft target for all samples; (3) use the teacher’s soft targets as additional targets to the ground truth to learn student networks. The Generalized Distillation loss is defined as:

$$\begin{aligned} \ell^{GD}(i) = (1 - \lambda)\ell(y_i, \sigma(f(x_i))) \\ + \lambda\ell(s_i, \sigma(f(x_i))), \lambda \in [0, 1] \end{aligned} \tag{5.3}$$

In contrast, we use distillation in an online fashion in the context of the MCL framework. The role of teacher / student network is assigned to the winner / loser network respectively, for each sample of the batch. The soft targets are computed using the winner network output, which is used to compute the loss and update the loser networks. We do not pretrain teachers as per conventional distillation, *i.e.* all networks are randomly initialized. In DMCL, teachers and students learn together in a cooperative setting.

This cooperative setting is beneficial in two ways: It gives loser networks the opportunity to build good representations even if they are not the *argmin* chosen network ; It still enables networks to specialize in parts of the problem.

Missing Modalities

Our training method encourages each network to learn using ground truth labels for its specialty samples (those obtaining lowest loss), and from the other specialist networks for samples otherwise. By doing so, each specialist incorporates

knowledge related to all samples/classes of the task. This enables each network to classify any sample at test time, therefore rendering the ensemble able to account for missing modalities.

5.2.2 Relationship to other MCL methods

The general framework for MCL is described in lines 1-17 of Algorithm 1. The main idea is to enable each of the networks of the ensemble to specialize in different parts of the problem. This algorithm was first devised for RGB ensembles. Two recent instances of MCL are Stochastic MCL (SMCL) [70] and Confident MCL (CMCL) [71]. These methods differentiate from each other and from the general MCL framework in two fundamental ways: 1) the criterion loss used to decide whether a network is a winner or a loser (line 10, Algorithm 1), and 2) how winner and loser models are updated (line 11 and 14, Algorithm 1). In SMCL, $\ell_{\text{criterion}}$ corresponds to the task loss, *e.g.* standard cross-entropy for classification. The winner model is updated with respect to that same loss, while the loser models are not updated. This update scheme is also used in [72]. In CMCL, the $\ell_{\text{criterion}}$ corresponds to the task loss plus an additional loss that measures how well the other networks predict the uniform distribution, for the given sample. The winner model is updated as in the SMCL method and the loser models are updated with respect to the KL divergence between its predictions and the uniform distribution.

Neither variations of MCL satisfy our problem statement. SMCL does not result in a single prediction. While CMCL does result in a single prediction by averaging the predictions, it does not account for the idiosyncrasies of multimodal data. The first aspect has to do with heterogeneous training dynamics resulting from having multimodal data as input. Figure 5.3 shows the cross-entropy loss of three networks independently trained for action recognition, using RGB (blue),

optical flow (orange), and depth (green). Optical flow learns at a much faster speed than the other modalities. This results in an undesired effect when using CMCL: the optical flow network repeatedly achieves the lowest loss. This behavior is reinforced by the *argmin* operator and the update scheme of CMCL, that does not allow useful gradients to pass to the loser networks. Eventually, the optical flow network ends up winning for all the training samples, which renders the other networks and modalities useless. The second challenge is the probable overfitting. The current training update scheme dictates that only the winner network gets useful gradients to build good representations for the given task, which reduces the data used to train each network. To address this and prevent overfitting, CMCL proposes to share the lower layers of the feature encoders. This is not feasible when the different networks are learning from different modalities as their representations/domains are significantly different.

DMCL addresses these issues for multimodal data by using a cooperative learning setting where the ensemble networks teach each other via Knowledge Distillation. At the same time, DMCL leverages the ensemble learning strategy of the traditional MCL framework, where models specialize depending on their performance with respect to a given input.

5.3 Experiments

In this section, we present the action recognition benchmark datasets we use to evaluate our approach. We then present the architecture and setup of our experiments. We analyze the performance of our DMCL in comparison to other MCL training strategies. We give insight into why other MCL training strategies fall short for multimodal data. We then demonstrate our privileged information state-of-the-art results and conclude with a discussion of our experimental results.

5.3.1 Datasets

We test DMCL on three video action recognition datasets that offer RGB and depth data. We augment the three datasets with optical flow frames obtained using the implementation available at [98], based on Liu *et al.* [99].

Northwestern-UCLA (NW-UCLA). This dataset [66] features ten people performing ten actions, captured simultaneously at three different viewpoints. We follow the cross-view protocol suggested by the authors in [66], using two views for training and the remaining for testing.

UWA3DII. This dataset [65] features ten subjects performing thirty actions for four different trials, each trial corresponding to a different viewpoint. As suggested in [65], we follow the cross-view protocol using two views for training and two for testing.

NTU120. The very recent NTU RGB+D 120 dataset [2] is one of the largest multimodal dataset for video action recognition. It consists of a total of 114,480 trimmed video clips of 106 subjects performing 120 classes, including single person and two-person actions, across 155 different viewpoints and 96 background scenes. We follow the cross-subject evaluation protocol proposed in the original paper, using fifty three subjects for training and the remaining for testing. We also create three versions of NTU120, which we refer to as NTU120^{mini} , that contains 50% sampled training data from the 120 classes. We note that NTU120 and NTU120^{mini} share the same test data. When results are reported on NTU120^{mini} they are averaged over the three runs. We also evaluate our method on the smaller less recent version of this dataset, NTU60 [67], that has 60 classes, in order to compare against state-of-the-art reported results.

5.3.2 Architecture and Setup

Each modality network is implemented as the R(2+1)D-18 architecture proposed in [58]. This architecture is based on a Resnet-18 network [75], modified such that a 1D temporal convolution is added after every 2D convolution, thus giving the network the ability to learn spatiotemporal features. The factorization of a 3D convolution into a combination of 2D + 1D convolution has shown to be more effective for video classification tasks. The ensemble of modality networks is simultaneously trained following Algorithm 1.

The input of each modality network is a clip of eight frames of the corresponding modality. For each training step, a video is split into eight equal parts and we randomly sample a frame from each of them. Each training input frame is a crop of dimension [224,224,3], cropped around a randomly shifted center, for each video. We also use other data augmentation techniques such as random horizontal flipping and random color distortions. The networks are trained from scratch for all the experiments, using SGD optimizer with Momentum 0.9, and an initial learning rate of 10^{-3} . At test time, we sample ten clips per video, each clip consisting of eight frames randomly sampled, centered, and with no data augmentation techniques. The final prediction for each video is the average of the ten clip predictions. We have experimented with different values of temperature T and hyperparameter λ , and found that $T=\{2,5\}$ and $\lambda=\{1, 0.5\}$ works best, with little accuracy variations. Further details related to hyperparameters are given in the supplementary material.

5.3.3 Results

In this section, we demonstrate how DMCL leverages multiple modalities to learn an RGB network that outperforms an independently trained RGB classifier - our baseline, and other MCL training strategies. All MCL strategies are trained using

the same training process as our method, including data augmentation techniques, optimizer, and number of steps, and are considered as ablation experiments of our method. We then demonstrate state-of-the-art privileged information results.

Comparison vs. MCL variants

Table 5.1 shows the action classification performance on the three video action recognition benchmark datasets for MCL variants and independently trained modality networks. We present the classification accuracy using the RGB modality, the sum of predictions of RGB, Flow, and Depth modalities (Σ), and the oracle accuracy (Φ). An oracle Φ is assumed to have the ability to select the modality that gives the best prediction among the ensemble. Our DMCL approach performs better than modalities trained independently, *i.e.* without MCL, and better than SMCL and CMCL variants. While Table 5.1 focuses on improvement with regard to the RGB modality, we provide similar results for Depth and Optical Flow in the supplementary material. We note that the effect of knowledge distillation is more visible in the three smaller datasets.

Table 5.1 also shows that combining the predictions of three modalities (Σ) generally improves accuracy. The fact that the oracle accuracy (Φ) is significantly higher than Σ indicates that, for some cases, at least one modality predicted the correct class, however, the sum of predictions (Σ) resulted in an incorrect prediction. However, the gap between Σ and Φ is lower for DMCL compared to the other approaches. This indicates that DMCL combines modality predictions in a more optimal fashion to improve overall accuracy. The low accuracies of SMCL and CMCL are due to artifacts created by the use of multimodal data, which we investigate in the next section. We have checked the implementation of these methods on RGB-only ensembles, which lead to similar results to those reported in the original papers.

Algorithm 1: DMCL

Input: Dataset $\mathbb{D} = \{(x_i, y_i)\}_i^N$, and randomly initialized networks f^1, \dots, f^M parameterized by $\theta^1, \dots, \theta^M$

Output: M trained networks f^1, \dots, f^M

```

1 for step  $\leftarrow 1$  to convergence do
2   | Sample batch  $\mathbb{B} \subset \mathbb{D}$ 
3   | for  $m \leftarrow 1$  to  $M$  do
4     | Forward Pass:
5     |  $\ell_{\text{criterion}}^m = \text{cross\_entropy}(y_i, \hat{y}^m)$ 
6     | end
7     | for  $i \leftarrow 1$  to  $|\mathbb{B}|$  do
8       | // Backward Pass:
9       | // Update winner network  $m^*$ 
10      |  $m^* \leftarrow \arg \min_{m \in \{1, \dots, M\}} \{\ell_{\text{criterion}}^m\}$ 
11      |  $\theta^{m^*} = \text{update\_winner}(\theta^{m^*}, x_i^{m^*}, y_i, f)$ 
12      | // Update loser networks  $m^c$ 
13      |  $m^c \leftarrow \{1, \dots, M\} \setminus \{m^*\}$ 
14      |  $\theta^{m^c} = \text{update\_losers}(\theta^{m^c}, x_i^{m^c}, y_i, f)$ 
15      | end
16   | end
17 return  $f^1, \dots, f^M$ 
18 // Function Definitions
19 Function  $\text{update\_winner}(\theta^{m^*}, x_i^{m^*}, y_i, f)$ :
20   | // Compute the gradient w.r.t. cross-entropy loss;
21   |  $\nabla_{\theta^{m^*}} \ell = \frac{\partial \ell(y_i, f^{m^*}(x_i^{m^*}))}{\partial \theta^{m^*}}$ ;
22   | // Update parameters of the winner network;
23   |  $\theta^{m^*} \leftarrow \theta^{m^*} - \eta \nabla_{\theta^{m^*}} \ell$ ;
24   | return  $\theta^{m^*}$ ;
25 Function  $\text{update\_losers}(\theta^{m^c}, x_i^{m^c}, y_i, f)$ :
26   | // Compute soft targets of  $f^{m^*}$  using Eq. 5.2;
27   |  $s_i^{m^*} = \sigma(f_i^{m^*}(x_i^{m^*})/T)$ ;
28   | // Compute soft targets of  $f^{m^c}$  using Eq. 5.2;
29   |  $s_i^{m^c} = \sigma(f_i^{m^c}(x_i^{m^c})/T)$ ;
30   | // Compute the gradient w.r.t. GD loss using Eq. 5.3;
31   |  $\nabla_{\theta^{m^c}} \ell^{GD} = \frac{\partial \ell^{GD}(y_i, f_i^{m^c}, s_i^{m^*}, s_i^{m^c})}{\partial \theta^{m^c}}$ ;
32   | // Update parameters of the loser networks;
33   |  $\theta^{m^c} \leftarrow \theta^{m^c} - \eta \nabla_{\theta^{m^c}} \ell^{GD}$ ;
34   | return  $\theta^{m^c}$ ;

```

	Independent			SMCL			CMCL			Our DMCL		
	RGB	Σ	Φ	RGB	Σ	Φ	RGB	Σ	Φ	RGB	Σ	Φ
NWUCLA	87.53	93.79	97.86	24.83	49.00	86.79	11.13	84.73	89.65	93.64	93.28	97.64
UWA3DII	73.74	89.75	95.52	25.19	60.70	88.51	22.28	31.90	83.89	78.39	89.50	94.96
NTU120 ^{mini}	79.66	86.57	92.11	26.67	62.22	86.19	29.61	5.28	86.29	81.25	86.23	91.71
NTU120	84.86	89.74	94.36	22.31	5.54	79.81	22.37	5.06	85.20	84.31	88.46	93.21

Table 5.1: Comparing MCL methods. We compare the performance of SMCL and CMCL with our proposed DMCL on the NWUCLA, UWA3DII, and NTU120 datasets. We also compare against independently trained modality networks. For each method we present the accuracy of the RGB modality network, the sum of all modality network predictions (Σ), and the oracle accuracy (Φ). For each row, corresponding to one dataset, we highlight in bold the best result using RGB only at test time. Using our DMCL methods results in better RGB networks for three out of four datasets.

Dataset Test Modality	NWUCLA					UWA3DII				
	RGB	Depth	Flow	Σ	Φ	RGB	Depth	Flow	Σ	Φ
Independent	87.53	80.30	89.58	93.79	97.86	73.74	77.09	89.66	89.75	95.52
Random Teacher	89.57	57.81	89.43	86.93	95.71	71.07	79.07	85.03	84.47	92.60
Our DMCL	93.64	83.29	91.07	93.28	97.64	78.39	81.87	88.26	88.51	94.59

Table 5.2: Selecting the right teacher network is important. We present the action recognition classification accuracy on the NWUCLA and UWA3DII datasets for three scenarios, where: modality networks are trained independently; a random teacher is assigned for every sample to guide the other modality networks; and DMCL, where the best-performing teacher (lowest loss) is selected to guide other modality networks. For each column, corresponding to a test modality, we highlight in bold the best result across the three scenarios.

Learning speed for different modalities

One of the goals of this work is to investigate and bring new insights on multi-modal learning. In a MCL setting, having a specific modality learn at a faster pace compared to others often leads to an imbalance of the number of data points each modality network is presented with at training time. Networks specializing in different modalities typically do not share a backbone of parameters due to the very different nature of the inputs - in contrast to the SMCL and CMCL variants where there is a shared backbone. As a consequence, if a modality network dominates the training process, *i.e.* being the one to consistently achieve the lowest loss for training batches, it will be presented with significantly more training data compared to the other modality networks. We observed that optical flow often dominates the ensemble training process particularly when training using CMCL. This is depicted in Figure 5.3 where the training loss curves of the independently trained networks for Optical Flow, Depth, and RGB are shown over the training steps. Namely, looking at the first steps of the curve we see that Optical Flow curve is consistently lower than Depth, which in turn has lower values than RGB. This is consistent with what we find during training of CMCL, where the RGB network is often ignored, the Depth network learns from a few samples and overfits early, and the Optical Flow network sees the vast majority of the samples.

We further investigate why optical flow dominates the learning process in our action recognition setting. We compute random features extracted from a randomly initialized untrained network for each of the modalities using the same architecture described previously. We then run a k NN classifier using the random features. Table 5.3 shows results of this experiment on the NWUCLA dataset for $k = 1, 5, 10, 50, 120$. The accuracy of the random features of the optical flow modality is almost twice that achieved using Depth and RGB. The fact that the k NN classifier achieves such good performance compared to the other modalities

suggests that Optical Flow data naturally clusters better per class. From the perspective of a deep neural network learning process, this could be interpreted as a better initialization, thus speeding the initial stage of learning.

KNN accuracy with random features					
Modality	$k=1$	$k=5$	$k=10$	$k=50$	$k=120$
RGB	10.53	10.74	11.11	11.32	12.26
Depth	9.72	10.68	10.77	15.37	13.31
Optical Flow	23.23	23.96	25.31	26.35	24.53

Table 5.3: Accuracy of a KNN classifier with varying k on the NWUCLA dataset. Classified features are computed using randomly initialized networks for each modality. Although all features are randomly generated, optical flow random features tend to achieve a significantly higher accuracy. This helps to explain why optical flow networks learn faster than other modalities.

Leveraging Teacher Strength

In this section, we ablate the mechanism by which the teacher role is determined. The teacher role is assigned to the network that achieves the lowest loss for each sample of the batch, therefore being in the best position to guide/strengthen the other networks. To verify this claim, we train our model with a random assignment of a teacher for each sample of the batch. This can be thought of as a randomized distillation process. We then compare the overall action recognition classification accuracy of both approaches in Table 5.2. Choosing the right network as teacher consistently achieves better performance compared to a randomly assigned teacher, for every modality. This is in-line with work that combines distillation and graphs, where the distillation process has a specific direction specified by the direction of the edges [100]. It is interesting to note that random teacher assignment may result in better performance than individual modality networks, *e.g.* for NWUCLA the RGB individual network accuracy is 87.53% *vs.* 89.57% for

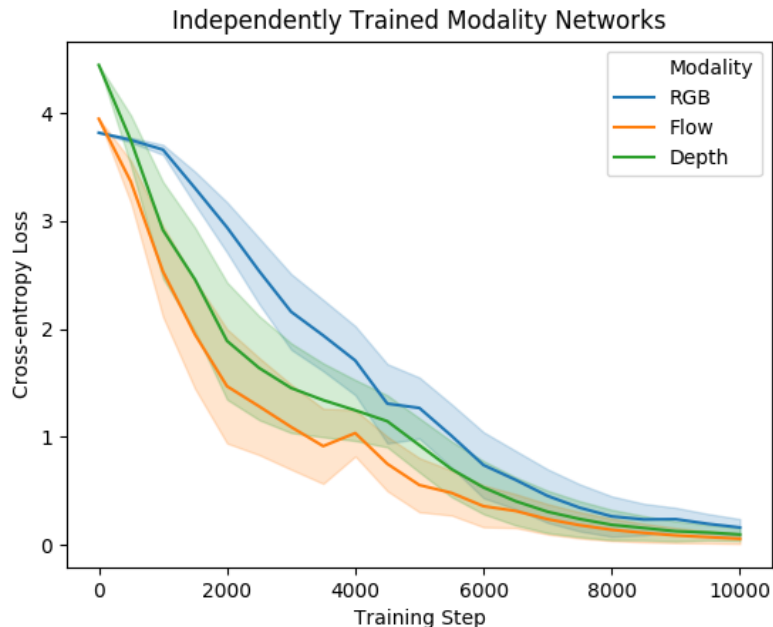


Figure 5.3: The cross-entropy loss of three networks independently trained for action recognition on the UWA3DII dataset, using RGB (blue), depth (green), and optical flow (orange). These plots are averaged over three runs. We observe that for the first 10K steps, the training loss of the optical flow network is consistently lower, resulting in a winner-takes-all behavior in traditional MCL algorithms. However, in DMCL, the winner network also teaches the loser networks, strengthening the other modality networks and avoiding this behavior.

a random teacher assignment. These may be related to the known regularization effect of knowledge distillation, that has been empirically shown to lead to better performance [25; 95].

State-of-the-art Comparisons

We now compare DMCL to state-of-the-art privileged information methods, and modality baselines, for the task of human action recognition from videos. Table 5.4 shows results for the UWA3DII and NWUCLA datasets. The top part of the table presents modality baselines for methods that use the same number of modalities in training and testing, including our individually trained modality

networks. The bottom part of the table refers to methods that have missing modalities at test time. Our DMCL using RGB only for testing achieves higher accuracy compared to all baselines that use RGB at training and testing, and compared to all state-of-the-art privileged information methods that use RGB at test time, including those that use additional hallucination networks at test time, achieving an absolute improvement of 4.7% for UWA3DII and 6.1% for NWUCLA. Similarly, our DMCL outperforms all baselines when the only available modality is Depth by 4.8% absolute improvement and the state-of-the-art method by 1.3% on UWA3DII.

Table 5.5 presents results on three versions of the NTU dataset: NTU60, NTU120^{mini}, and the full NTU120. We see that the distillation effect is much more visible in the case of less data. For example, for NTU^{mini}, we achieve an absolute improvement of 1.6% over the baseline for the RGB modality, and of 6% for NTU60. Our best modality network for NTU60 achieves 85.65% compared to the 89.5% of [100] that uses twice the number of modalities we use for training and an additional graph network module.

5.4 Summary

MCL is a powerful way for training ensembles of networks, originally proposed for RGB data. We demonstrate undesirable behaviors of this framework when naively applied to multimodal data. We propose DMCL that extends MCL frameworks to leverage the complementary information offered by the multimodal data to the benefit of the ensemble. The cooperative learning is enabled via knowledge distillation that allows the ensemble networks to exchange information and learn from each other. We demonstrate that modality networks trained using our DMCL achieve competitive to or state-of-the-art results compared to the privileged infor-

mation literature, and significantly higher accuracy compared to independently trained modality networks for human action recognition in videos.

	Method	Training Modalities	Testing Modalities	UWA3DII	NWUCLA
<i>Modality Baselines</i>	R-NKTM [90]	Syn*	RGB	66.3	78.1
	Action Tubes [91]	RGB	RGB	33.7	61.5
	Long-term RCNN [101]	RGB	RGB	74.5	64.7
	Baseline (RGB)	RGB	RGB	73.74	87.52
	MVDI+CNN [102]	Depth	Depth	68.3	84.2
	Baseline (D)	Depth	Depth	77.09	80.30
	Baseline (F)	Flow	Flow	89.66	89.58
	Baseline (RGB, D, F)	RGB, Depth, Flow	RGB, Depth, Flow	89.75	93.9
<i>Privileged Info.</i>	Hoffman <i>et al.</i> [12]	RGB, Depth	RGB ⁺	66.67	83.30
	Garcia <i>et al.</i> [73]	RGB, Depth	RGB ⁺	73.23	86.72
	ADMD [14]	RGB, Depth	RGB ⁺	-	91.64
	DMCL	RGB, Depth, Flow	RGB	78.39	93.64
	DMCL	RGB, Depth, Flow	Depth	81.87	83.29
DMCL	RGB, Depth, Flow	Flow	88.26	91.07	

Table 5.4: Accuracy for UWA3DII and NWUCLA dataset. The first part of the table refers to methods that use unsupervised feature learning (*) or that use the same number of modalities for training and testing. The second part of the table refers to methods that use more modalities for training than for testing. Methods that use RGB⁺ at test time use an additional network that mimics the missing modality. For each column, corresponding to one dataset, we highlight in colored bold the best result and in normal colored font the second best between our method and the baselines. Each color corresponds to a different test modality. To conduct a fair comparison with baseline methods, this table presents results for the most common view setting for UWA3DII and NWUCLA. Other view settings follow the same trend and results are presented in the supplementary material.

	Method	Training Modalities	Testing Modalities	NTU60	NTU120 ^{mini}	NTU120
<i>Modality Baselines</i>	ST-LSTM [103][2]	Skeleton	Skeleton	69.2	~ 50.0	55.7
	VGG [2]	RGB	RGB	-	~ 40.0	58.5
	Baseline (RGB)	RGB	RGB	77.59	79.66	84.86
	VGG [2]	Depth	Depth	-	~ 20.0	48.7
	Baseline (D)	Depth	Depth	78.97	78.67	83.32
	Baseline (F)	Flow	Flow	81.43	84.21	86.72
	VGG [2]	RGB,Depth	RGB, Depth	-	-	61.9
	VGG [2]	RGB, Depth, 3D Skeleton	RGB, Depth, 3D Skeleton	-	-	64.0
	Baseline (RGB, D, F)	RGB, Depth, Flow	RGB, Depth, Flow	87.25	86.57	89.74
<i>Privileged Info.</i>	Garcia <i>et al.</i> [14]	RGB, depth	RGB	73.11	-	-
	ADMD [73]	RGB, Depth	RGB	73.4	-	-
	Luo <i>et al.</i> [100]	RGB, OF, Depth, 3D Skeleton ^{1,2,3}	RGB	89.5	-	-
	DMCL	RGB, Depth, Flow	RGB	83.61	81.25	84.31
	DMCL	RGB, Depth, Flow	Depth	80.56	78.98	82.22
	DMCL	RGB, Depth, Flow	Flow	85.65	84.45	86.44

Table 5.5: Evaluation on NTU datasets. The test sets for NTU120^{mini} and NTU120 are the same. For each column, corresponding to one dataset, we highlight in bold the best result and in normal colored font the second best between our method and the baselines. Each color corresponds to a different test modality. The approximated values are inferred from a plot in [2]. We note that the effect of the distillation method is more visible on the smaller scale versions NTU60 and NTU120^{mini} of the dataset.

Chapter 6

Conclusions

In this thesis, we have explored the problem of multimodal learning in the context of privileged information, *i.e.* having access to more modalities for training than for testing. We have developed three deep learning methods to this end, and evaluated these on the tasks of video action recognition and object recognition. Our work is framed around the concept of machines-teaching-machines, *i.e.* we are interested in learning models using multimodal data that are able to teach or guide the learning process of other models, used at test time, and with access to less resources, namely less modalities.

The first solution consists in learning an hallucination network that, using RGB data, mimics the features and predictions of a network trained with depth data. This is accomplished using a teacher network that was trained using depth images, and provide targets for every training data point. At test time, since RGB is the only modality available, the hallucination network is used to compensate for the missing depth data, and enhance the RGB model. The second solution extends this work by introducing a novel learning algorithm to train the hallucination network. Instead of using a distance-based distance, we develop an adversarial learning algorithm to align the features and predictions of both networks. This achieves a better performance, and allows for the interesting ap-

plication of automatically switching to hallucinated features in case of a noisy input. The third solution focus on leveraging the complementarity offered by the diverse modalities. We developed a method to learn an ensemble of multi-modal networks in a cooperative setting. The algorithm explores the networks and modalities that perform better for each data point. By dynamically assigning the role of teacher and students networks throughout the training, the strongest modality networks are used to the benefit of the ensemble and each individual networks.

In a deep learning era, where algorithms are flexible and able to learn from huge amounts of data from diverse modalities, the ability to understand the relation between modalities and leverage the complementarity of information is fundamental. With theses works, we have shown that multimodal learning may be used to produce better single modality models. The paradigm of machines-teaching-machines is a promising framework to develop multimodal learning. Future work could address this topic by extending to other modalities such as sound or text. This work focused on fully-supervised methods. The future of feature learning is probably moving towards a self-supervised setting, in which multi-modal data offers very exciting possibilities and challenges regarding cross-modal methods. We hope this thesis is a step forward in this direction, and hope to encourage the development of more methods in this field.

References

- [1] C. Feichtenhofer, A. Pinz, and R. P. Wildes, “Spatiotemporal multiplier networks for video action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4768–4777, 2017. [ix](#), [11](#), [16](#), [17](#), [18](#), [20](#), [25](#), [28](#), [34](#), [36](#), [46](#)
- [2] J. Liu, A. Shahroudy, M. L. Perez, G. Wang, L.-Y. Duan, and A. K. Chichung, “Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding,” *IEEE transactions on pattern analysis and machine intelligence*, 2019. [xvi](#), [12](#), [64](#), [73](#), [84](#)
- [3] E. J. Gibson and R. D. Walk, “The ”visual cliff”,” *Scientific American*, vol. 202, no. 4, pp. 64–71, 1960. [1](#)
- [4] M. R. Watson and J. T. Enns, “Depth perception,” *Encyclopedia of Human Behavior*, 2012. [1](#)
- [5] P. Servos, “Distance estimation in the visual and visuomotor systems,” *Experimental Brain Research*, vol. 130, pp. 35–47, Jan 2000. [1](#)
- [6] M. Firman, “Rgb-d datasets: Past, present and future,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 19–31, 2016. [2](#)
- [7] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang, “Deep multimodal feature

REFERENCES

- analysis for action recognition in rgb+ d videos,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. [2](#), [12](#), [27](#), [56](#)
- [8] J. Liu, N. Akhtar, and A. Mian, “Viewpoint invariant action recognition using rgb-d videos,” *arXiv preprint arXiv:1709.05087*, 2017. [2](#), [12](#), [27](#), [56](#)
- [9] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, “Learning rich features from rgb-d images for object detection and segmentation,” in *European Conference on Computer Vision*, pp. 345–360, Springer, 2014. [2](#), [11](#), [13](#)
- [10] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, “Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture,” in *Asian Conference on Computer Vision*, pp. 213–228, Springer, 2016. [2](#)
- [11] V. Vapnik and A. Vashist, “A new learning paradigm: Learning using privileged information,” *Neural networks*, vol. 22, no. 5, pp. 544–557, 2009. [3](#), [7](#), [16](#), [66](#)
- [12] J. Hoffman, S. Gupta, and T. Darrell, “Learning with side information through modality hallucination,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 826–834, 2016. [3](#), [9](#), [16](#), [20](#), [21](#), [23](#), [27](#), [28](#), [29](#), [31](#), [37](#), [39](#), [40](#), [43](#), [46](#), [49](#), [54](#), [56](#), [83](#)
- [13] N. Garcia, P. Morerio, and V. Murino, “Modality distillation with multiple stream networks for action recognition,” *European Conference on Computer Vision*, 2018. [4](#), [5](#), [39](#), [40](#), [43](#), [46](#), [54](#), [55](#), [56](#), [58](#), [59](#)
- [14] N. C. Garcia, P. Morerio, and V. Murino, “Learning with privileged information via adversarial discriminative modality distillation,” *IEEE transactions on pattern analysis and machine intelligence (TPAMI)*, 2019. [5](#), [6](#), [38](#), [83](#), [84](#)

-
- [15] N. C. Garcia, S. A. Bargal, V. Ablavsky, P. Morerio, V. Murino, and S. Sclaroff, “Dmcl: Distillation multiple choice learning,” *under revision*, 2019. [5](#), [6](#)
- [16] N. C. Garcia, P. Morerio, and V. Murino, “Chapter 12 - cross-modal learning by hallucinating missing modalities in rgb-d vision,” in *Multimodal Scene Understanding* (M. Y. Yang, B. Rosenhahn, and V. Murino, eds.), pp. 383 – 401, Academic Press, 2019. [5](#), [15](#)
- [17] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. 2016. [7](#)
- [18] D. Lopez-Paz, L. Bottou, B. Schölkopf, and V. Vapnik, “Unifying distillation and privileged information,” *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016. [7](#), [16](#), [21](#), [39](#), [40](#), [65](#), [67](#), [70](#)
- [19] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *Deep Learning and Representation Learning Workshop: NIPS 2014*, 2014. [7](#), [16](#)
- [20] L. J. Ba and R. Caruana, “Do deep nets really need to be deep?,” *Proceedings of Advances in Neural Information Processing Systems (NIPS)*, 2014. [7](#), [16](#)
- [21] Z. Luo, L. Jiang, J.-T. Hsieh, J. C. Niebles, and L. Fei-Fei, “Graph distillation for action detection with privileged information,” *arXiv preprint arXiv:1712.00108*, 2017. [8](#), [43](#), [54](#), [56](#)
- [22] Z. Shi and T.-K. Kim, “Learning and refining of privileged information-based rnns for action recognition from depth sequences,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. [8](#)

REFERENCES

- [23] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout, “Moddrop: Adaptive multi-modal gesture recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1692–1706, 2016. [9](#), [57](#), [58](#), [60](#), [61](#)
- [24] C. Buciluă, R. Caruana, and A. Niculescu-Mizil, “Model compression,” in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541, ACM, 2006. [9](#)
- [25] T. Furlanello, Z. C. Lipton, M. Tschannen, L. Itti, and A. Anandkumar, “Born again neural networks,” *arXiv preprint arXiv:1805.04770*, 2018. [9](#), [80](#)
- [26] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a defense to adversarial perturbations against deep neural networks,” in *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597, IEEE, 2016. [9](#)
- [27] S. Gupta, J. Hoffman, and J. Malik, “Cross modal distillation for supervision transfer,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2827–2836, 2016. [9](#)
- [28] J. Vongkulbhisal, P. Vinayavekhin, and M. Visentini-Scarzarella, “Unifying heterogeneous classifiers with distillation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3175–3184, 2019. [9](#)
- [29] Y. Zhang, T. Xiang, T. M. Hospedales, and H. Lu, “Deep mutual learning,” [9](#)
- [30] C. Yang, L. Xie, C. Su, and A. L. Yuille, “Snapshot distillation: Teacher-student optimization in one generation,” in *Proceedings of the IEEE Con-*

-
- ference on Computer Vision and Pattern Recognition*, pp. 2859–2868, 2019. [9](#)
- [31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems*, pp. 2672–2680, 2014. [10](#), [39](#), [41](#)
- [32] X. Chen, Y. Duan, R. Houthoofd, J. Schulman, I. Sutskever, and P. Abbeel, “Infogan: Interpretable representation learning by information maximizing generative adversarial nets,” in *Advances in neural information processing systems*, pp. 2172–2180, 2016. [10](#)
- [33] V. Belagiannis, A. Farshad, and F. Galasso, “Adversarial network compression,” *arXiv preprint arXiv:1803.10750*, 2018. [10](#), [62](#)
- [34] Z. Xu, Y.-C. Hsu, and J. Huang, “Training student networks for acceleration with conditional adversarial networks,” [10](#)
- [35] Y. Wang, C. Xu, C. Xu, and D. Tao, “Adversarial learning of portable student networks,” in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. [10](#)
- [36] R. Volpi, P. Morerio, S. Savarese, and V. Murino, “Adversarial feature augmentation for unsupervised domain adaptation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. [10](#), [46](#), [48](#), [49](#), [51](#)
- [37] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. [10](#)

REFERENCES

- [38] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *International Conference on Machine Learning*, pp. 214–223, 2017. [10](#)
- [39] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” in *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016. [10](#)
- [40] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” *arXiv preprint arXiv:1511.06434*, 2015. [10](#)
- [41] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *arXiv preprint arXiv:1411.1784*, 2014. [10](#), [44](#)
- [42] C. Li, Z. Wang, and H. Qi, “Fast-converging conditional generative adversarial networks for image synthesis,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 2132–2136, IEEE, 2018. [10](#)
- [43] S. Roheda, B. S. Riggan, H. Krim, and L. Dai, “Cross-modality distillation: A case for conditional generative adversarial networks,” *arXiv preprint arXiv:1807.07682*, 2018. [10](#)
- [44] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 886–893, IEEE, 2005. [11](#)
- [45] H. Wang and C. Schmid, “Action recognition with improved trajectories,” in *Proceedings of the IEEE international conference on computer vision*, pp. 3551–3558, 2013. [11](#)

- [46] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, “Learning realistic human actions from movies,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008. [11](#)
- [47] E. S. Ye and J. Malik, “Object detection in rgb-d indoor scenes,” *Technical Report of University of California at Berkeley*, 2013. [11](#)
- [48] S. Tang, X. Wang, X. Lv, T. X. Han, J. Keller, Z. He, M. Skubic, and S. Lao, “Histogram of oriented normal vectors for object recognition with a depth sensor,” in *Asian conference on computer vision*, pp. 525–538, Springer, 2012. [11](#)
- [49] A. Janoch, S. Karayev, Y. Jia, J. T. Barron, M. Fritz, K. Saenko, and T. Darrell, “A category-level 3d object dataset: Putting the kinect to work,” in *Consumer depth cameras for computer vision*, pp. 141–165, Springer, 2013. [11](#)
- [50] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Advances in neural information processing systems*, pp. 568–576, 2014. [11](#), [16](#), [18](#), [66](#)
- [51] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014. [11](#)
- [52] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3d convolutional networks,” in *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497, 2015. [11](#)

-
- [53] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” *arXiv preprint arXiv:1711.07971*, 2017. [11](#)
- [54] A. Wang, J. Lu, J. Cai, T.-J. Cham, and G. Wang, “Large-margin multi-modal deep learning for rgb-d object recognition,” *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1887–1898, 2015. [11](#), [13](#)
- [55] P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera, “Rgb-d-based human motion recognition with deep learning: A survey,” *Computer Vision and Image Understanding*, vol. 171, pp. 118–139, 2018. [11](#)
- [56] S. Herath, M. Harandi, and F. Porikli, “Going deeper into action recognition: A survey,” *Image and vision computing*, vol. 60, pp. 4–21, 2017. [11](#)
- [57] Y. Kong and Y. Fu, “Human action recognition and prediction: A survey,” *arXiv preprint arXiv:1806.11230*, 2018. [11](#)
- [58] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2018. [12](#), [46](#), [74](#)
- [59] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. [12](#)
- [60] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang, “Optical flow guided feature: A fast and robust motion representation for video action recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1390–1399, 2018. [12](#)

-
- [61] M. Lee, S. Lee, S. Son, G. Park, and N. Kwak, “Motion feature network: Fixed motion filter for action recognition,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 387–403, 2018. [12](#)
- [62] A. Piergiovanni and M. S. Ryoo, “Representation flow for action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9945–9953, 2019. [12](#)
- [63] J. Zhao and C. G. Snoek, “Dance with flow: Two-in-one stream action detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9935–9944, 2019. [12](#)
- [64] N. Crasto, P. Weinzaepfel, K. Alahari, and C. Schmid, “Mars: Motion-augmented rgb stream for action recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7882–7891, 2019. [12](#)
- [65] H. Rahmani, A. Mahmood, D. Huynh, and A. Mian, “Histogram of oriented principal components for cross-view action recognition,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 12, pp. 2430–2443, 2016. [12](#), [73](#)
- [66] J. Wang, X. Nie, Y. Xia, Y. Wu, and S.-C. Zhu, “Cross-view action modeling, learning and recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2649–2656, 2014. [12](#), [40](#), [49](#), [73](#)
- [67] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, “Ntu rgb+ d: A large scale dataset for 3d human activity analysis,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1010–1019, 2016. [12](#), [17](#), [25](#), [26](#), [27](#), [36](#), [40](#), [49](#), [50](#), [56](#), [73](#)

REFERENCES

- [68] O. Sagi and L. Rokach, “Ensemble learning: A survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1249, 2018. [13](#)
- [69] A. Guzman-Rivera, D. Batra, and P. Kohli, “Multiple choice learning: Learning to produce multiple structured outputs,” in *Advances in Neural Information Processing Systems*, pp. 1799–1807, 2012. [13](#)
- [70] S. Lee, S. P. S. Prakash, M. Cogswell, V. Ranjan, D. Crandall, and D. Batra, “Stochastic multiple choice learning for training diverse deep ensembles,” in *Advances in Neural Information Processing Systems*, pp. 2119–2127, 2016. [13](#), [65](#), [71](#)
- [71] K. Lee, C. Hwang, K. S. Park, and J. Shin, “Confident multiple choice learning,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2014–2023, JMLR. org, 2017. [13](#), [14](#), [65](#), [71](#)
- [72] K. Tian, Y. Xu, S. Zhou, and J. Guan, “Versatile multiple choice learning and its application to vision computing,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6349–6357, 2019. [13](#), [14](#), [65](#), [71](#)
- [73] N. C. Garcia, P. Morerio, and V. Murino, “Modality distillation with multiple stream networks for action recognition,” in *The European Conference on Computer Vision (ECCV)*, September 2018. [15](#), [83](#), [84](#)
- [74] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. [16](#), [53](#)
- [75] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image

-
- recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016. [18](#), [45](#), [74](#)
- [76] K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in *European Conference on Computer Vision*, pp. 630–645, Springer, 2016. [18](#)
- [77] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, “Multimodal deep learning for robust rgb-d object recognition,” in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pp. 681–687, IEEE, 2015. [23](#), [26](#), [47](#), [49](#)
- [78] Z. Luo, B. Peng, D.-A. Huang, A. Alahi, and L. Fei-Fei, “Unsupervised learning of long-term motion dynamics for videos,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, no. EPFL-CONF-230240, 2017. [26](#), [27](#), [56](#)
- [79] E. Ohn-Bar and M. M. Trivedi, “Joint angles similarities and hog2 for action recognition,” in *Computer vision and pattern recognition workshops (CVPRW), 2013 IEEE conference on*, pp. 465–470, IEEE, 2013. [27](#), [56](#)
- [80] T. Soo Kim and A. Reiter, “Interpretable 3d human action analysis with temporal convolutional networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 20–28, 2017. [27](#), [56](#)
- [81] T. Mallick, P. P. Das, and A. K. Majumdar, “Characterizations of noise in kinect depth images: A review,” *IEEE Sensors Journal*, vol. 14, pp. 1731–1740, June 2014. [32](#), [60](#)
- [82] S. Arora, A. Risteski, and Y. Zhang, “Do GANs learn the distribution?”

- some theory and empirics,” in *International Conference on Learning Representations*, 2018. 39
- [83] N. Silberman, D. Hoiem, P. Kohli, , and R. Fergus, “Indoor segmentation and support inference from rgb-d images,” in *European Conference on Computer Vision (ECCV)*, 2012. 40, 49
- [84] “How to train a gan? tips and tricks to make gans work.” <https://github.com/soumith/ganhacks>. 45
- [85] L. Sun, K. Jia, D.-Y. Yeung, and B. E. Shi, “Human action recognition using factorized spatio-temporal convolutional networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4597–4605, 2015. 46
- [86] S. Gupta, R. Girshick, P. A. aez, and J. Malik, “Learning rich features from rgb-d images for object detection and segmentation,” in *European Conference on Computer Vision (ECCV)*, 2014. 47, 49
- [87] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, “Adversarial discriminative domain adaptation,” in *CVPR*, 2017. 49
- [88] P. Morerio, J. Cavazza, and V. Murino, “Minimal-entropy correlation alignment for unsupervised deep domain adaptation,” in *International Conference on Learning Representations*, 2018. 49
- [89] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015. 51

REFERENCES

- [90] H. Rahmani, A. Mian, and M. Shah, “Learning a deep model for human action recognition from novel viewpoints,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 667–681, 2018. [56](#), [83](#)
- [91] G. Gkioxari and J. Malik, “Finding action tubes,” in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pp. 759–768, IEEE, 2015. [56](#), [83](#)
- [92] H. Rahmani and M. Bennamoun, “Learning action recognition model from depth and skeleton videos,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5832–5841, 2017. [56](#)
- [93] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab, “Deeper depth prediction with fully convolutional residual networks,” in *3D Vision (3DV), 2016 Fourth International Conference on*, pp. 239–248, IEEE, 2016. [57](#), [58](#)
- [94] J. Guo and S. Gould, “Deep cnn ensemble with data augmentation for object detection,” *CoRR*, vol. abs/1506.07224, 2015. [57](#)
- [95] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015. [65](#), [66](#), [80](#)
- [96] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. [66](#)
- [97] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint arXiv:1603.04467*, 2016. [67](#)

REFERENCES

- [98] D. Pathak, “Pyflow - python dense optical flow.” <https://github.com/pathak22/pyflow>. 73
- [99] C. Liu *et al.*, *Beyond pixels: exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009. 73
- [100] Z. Luo, J.-T. Hsieh, L. Jiang, J. Carlos Niebles, and L. Fei-Fei, “Graph distillation for action detection with privileged modalities,” in *The European Conference on Computer Vision (ECCV)*, September 2018. 79, 81, 84
- [101] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625–2634, 2015. 83
- [102] Y. Xiao, J. Chen, Y. Wang, Z. Cao, J. T. Zhou, and X. Bai, “Action recognition for depth video using multi-view dynamic images,” *Information Sciences*, vol. 480, pp. 287–304, 2019. 83
- [103] J. Liu, A. Shahroudy, D. Xu, and G. Wang, “Spatio-temporal lstm with trust gates for 3d human action recognition,” in *European Conference on Computer Vision*, pp. 816–833, Springer, 2016. 84