UNIVERSITY OF GENOA, ITALY

DITEN - Department of Electrical, Electronics and Telecommunication Engineering and Naval Architecture
PAVIS - Pattern Analysis and Computer Vision, Italian Institute of Technology
PhD Course: Science and Technology for Electronic and Telecommunication Engineering - Cycle XXXIV (2018-2021)

Collaborative Learning in Computer Vision

PhD Thesis

Thesis submitted for the degree of *Doctor of Philosophy*

PhD Candidate: **Waqar Ahmed**

Supervisors: Prof. Vittorio Murino & Dr. Alessio del Bue

Co-supervisors: Dr. Pietro Morerio & Dr. Andrea Zunino

Coordinator of the PhD Course: Prof. Maurizio Valle

November 30, 2021

To

My dearest wife Urooj and beloved family members

for all the love, patience, and support.

Acknowledgements

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

I would like to express my deep gratitude to my primary supervisor, **Prof. Vittorio Murino**, for the unparalleled guidance and support.

I would also like to thank my co-supervisors **Andrea Zunino** and **Pietro Morerio**, for all the useful insights on deep learning, ideas, and experiments that make some of my favorite parts in this dissertation. They serve as a research role model for me, which I hope to emulate in the future.

I would like to thank and acknowledge the help provided by **Abubakar**, **Shahid**, **Dahi**, **Usman**, **Ruggero**, **Federico**, **Valentina**, **Jacopo**, **Riccardo**, **Yiming**, **Vaibhav**, **Milind**, **Nawaz**, and **Nuno**. I have been 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 am grateful to **Alessio del Bue**, **Prof. Mario Marchese**, **Prof. Maurizio Valle**, and all staff members for the helpful, cooperative and lively environment.

Last but not least, I wish to extend my special appreciation to my parents, who have always been a source of inspiration and gave me strength when I thought of giving up, who continuously provide their moral, spiritual and emotional support.

Thank you everyone!

Waqar Ahmed
November 2021

Abstract

The science of designing machines to extract meaningful information from digital images, videos, and other visual inputs is known as Computer Vision (CV). Deep learning algorithms cope CV problems by automatically learning task-specific features. Especially, Deep Neural Networks (DNNs) have become an essential component in CV solutions due to their ability to encode large amounts of data and capacity to manipulate billions of model parameters.

Unlike machines, humans learn by rapidly constructing abstract models. This is undoubtedly due to the fact that good teachers supply their students with much more than just the correct answer; they also provide intuitive comments, comparisons, and explanations. In deep learning, the availability of such auxiliary information at training time (but not at test time) is referred to as learning by Privileged Information (PI). Typically, predictions (*e.g.,* soft labels) produced by a bigger and better network *teacher* are used as structured knowledge to supervise the training of a smaller network *student*, helping the student network to generalize better than that trained from scratch.

This dissertation focuses on the category of deep learning systems known as *Collaborative Learning*, where one DNN model helps other models or several models help each other during training to achieve strong generalization and thus high performance. The question we address here is thus the following: how can we take advantage of PI for training a deep learning model, knowing that, at test time, such PI might be missing? In this context, we introduce new methods to tackle several challenging real-world computer vision problems.

First, we propose a method for model compression that leverages PI in a *teacher-student* framework along with customizable block-wise optimization for *learning* a target-specific lightweight structure of the neural network. In particular, the proposed resource-aware opti-

mization is employed on suitable parts of the *student* network while respecting the expected resource budget (*e.g.,* floating-point operations per inference and model parameters). In addition, soft predictions produced by the *teacher* network are leveraged as a source of PI, forcing the student to preserve baseline performance during network structure optimization.

Second, we propose a multiple-model learning method for action recognition, specifically devised for challenging video footages in which actions are not explicitly visualized, but rather, only implicitly referred. We use such videos as stimuli and involve a large sample of subjects to collect a high-definition EEG and video dataset. Next, we employ collaborative learning in a multi-modal setting *i.e.,* the EEG (*teacher*) model helps the video (*student*) model by distilling the knowledge (implicit meaning of visual stimuli) to it, sharply boosting the recognition performance.

The goal of Unsupervised Domain Adaptation (UDA) methods is to use the labeled source together with the unlabeled target domain data to train a model that generalizes well on the target domain. In contrast, we cast UDA as a pseudo-label refinery problem in the challenging *source-free* scenario *i.e.,* in cases where the source domain data is inaccessible during training. We propose Negative Ensemble Learning (NEL) technique, a unified method for adaptive noise filtering and progressive pseudo-label refinement. In particular, the ensemble members collaboratively learn with a *Disjoint Set of Residual Labels*, an outcome of the output prediction consensus, to refine the challenging noise associated with the inferred pseudo-labels. A single model trained with the refined pseudo-labels leads to superior performance on the target domain, without using source data samples at all.

We conclude this dissertation with a method extending our previous study by incorporating *Continual Learning* in the Source-Free UDA. Our new method comprises of two stages: a Source-Free UDA pipeline based on pseudo-label refinement, and a procedure for extracting class-conditioned source-style images by leveraging the pre-trained source model. While stage 1 holds the same collaborative peculiarities, in stage 2, the collaboration exists in an indirect manner *i.e.,* it is the source model that provides the *only* possibility to generate

source-style synthetic images which eventually helps the final model in preserving good performance on both source and target domains.

In each study, we consider heterogeneous CV tasks. Nevertheless, with an extensive pool of experiments on various benchmarks carrying diverse complexities and challenges, we show that the collaborative learning framework outperforms the related state-of-the-art methods by a considerable margin.

Contents

List of Figures

List of Tables

Common Abbreviations

CV	Computer Vision
CL	Collaborative Learning
DNN	Deep Neural Network
CNN	Convolutional Neural Network
PI	Privileged Information
NL	Negative Learning
NEL	Negative Ensemble Learning
UDA	Unsupervised Domain Adaptation

Chapter 1

Introduction

Computer Vision (CV) typical tasks, such as image classification and object detection, have been traditionally addressed employing hand-crafted features, such as SIFT [7] and HOG [8], usually followed by learning algorithms like Support Vector Machines (SVMs) [9]. Thus, the main processing pipeline was mainly composed of two steps, namely, feature design and learning algorithm design, both of which were mostly independent.

More recently, deep learning algorithms have provided an appealing alternative: automatically learning task-specific features. With this new paradigm, every problem in computer vision is now being re-investigated from a deep learning perspective. Among different types of Deep Neural Networks (DNNs), Convolutional Neural Network (CNN) has been widely adopted by the vision community [10]. Such architectures are analogous to the connectivity pattern of the *neurons* in the human brain which have advanced tremendously in the last decade, owing to the GPU-accelerated computation, the development of high-capacity models, and the availability of large datasets.

In this dissertation, we focus on the concept of machines-teaching-machines framework, which, roughly speaking, aims at developing deep learning systems in which multiple models help each other during training to boost the generalization ability of DNNs carrying different learning capacities. We call this paradigm as *Collaborative Learning* [11]. A typical approach is known as a "teacher-student" learning, in which predictions (*e.g.,* soft labels) produced by a bigger and better *teacher* are used as structured knowledge to supervise the training of the smaller *student* model, helping the student network to generalize better than that trained from scratch. In the case of model ensemble (*e.g.,* more than 2 DNN models), the acquired experience is shared among arbitrary ensemble members, enabling them to gain extra knowledge from each other.

Among existing works, the framework presented in [12], introduces a unifying perspective on two distinct theories: Privileged Information [13] and Knowledge Distillation [14, 15]. The former, also known as Learning Using Privileged Information (LUPI), introduces to the learning process leveraging a "teacher" model that, in addition to the label supervision, provides additional information to a "student" model. The intuition is that the additional rationalisations provided by the teacher enable the student model to learn better than if it were

only trained using label supervision. Importantly, the teacher's additional information is only available to the student during training time, thus the term is called "privileged information".

On the other hand, Knowledge Distillation (KD) is a training procedure that distills information from a larger to a smaller model by transferring knowledge from a previously trained large model or ensemble of models. This concept stems from the fact that the training and testing phases have very different speed and computation requirements. These ideas have in common the concept of machines-teaching-machines: the inference model learns from a model that was previously trained in a more advantageous condition, such as with more information or better data, or is simply an ensemble of several large models.

The works presented in this dissertation are related to both, the privileged information theory and to knowledge distillation, and address these from a Collaborative Learning perspective. Precisely, the question we address is the following: how can we take advantage of PI for training the target model, knowing that, at test time, such PI might be missing? In this context, we introduce new methods to address four different challenging real-world computer vision problems: 1) Deep Neural Network Model Compression, 2) Multi-Model Learning for Action Recognition, 3) Cleaning Noisy Labels, and 4) Continual Learning for Source-Free Unsupervised Domain Adaptation.

1.1 Contributions and Outline

This thesis discusses several approaches we propose to address computer vision problems of diverse challenges demonstrating that Collaborative Learning is a reliable and effective deep learning system design choice.

Chapter 2 describes our proposed method for *model compression*. The concept of compressing deep neural networks (*e.g.,* CNN) is essential to use limited computation, power, and memory resources, especially on low-compute embedded devices. However, existing methods achieve this objective at the cost of a drop in inference accuracy in computer vision tasks. To address such a drawback, we propose a framework that leverages privileged information in a *teacher-student* framework along with customizable block-wise optimization to *learn* lightweight neural network structure while preserving better control over the compression-performance trade-off. Considering specific resource constraints, *e.g.,* floating-point operations per inference (FLOPs) or model parameters, our method results in a state-of-the-art network compression while being capable of achieving better inference accuracy.

Chapter 3 describes our proposed approach for *action recognition*, pursuing a challenging direction of showing that Electroencephalography (EEG) is a richer and complementary data

modality with respect to video, useful to classify what visual stimuli *implicitly mean* behind their visual appearance. We consider a challenging computer vision benchmark *i.e.,* Moments in Time dataset (MiT) [16], designed for action recognition in which video footages do not explicitly visualize the action to be recognized, only implicitly referring to it (*e.g.,* the first-person view of the landscape seen from an airplane window or extreme and vague cases, such as exploding fireworks symbolize the action as "flying"). We employ such videos as stimuli and involve a large sample of subjects to collect a high-definition, multi-modal EEG and video data, designed for understanding action concepts. We discover an agreement among brain activities of different subjects stimulated by the same video footage. We call this *subjects consensus*, and we design a computational pipeline, based on privileged information, to transfer knowledge from EEG to video, sharply boosting the recognition performance.

Chapter 4 describes our proposed method for *Source-Free Unsupervised Domain Adaptation (UDA)*. Conventional UDA methods presume source and target domain data to be simultaneously available during training. Such an assumption may not hold in practice, as source data is often inaccessible (*e.g.,* due to privacy reasons). On the contrary, a pre-trained source model is usually available, which performs poorly on target due to the well-known *domain shift* problem. This translates into a significant amount of misclassifications, which can be interpreted as structured noise affecting the inferred target pseudo-labels. We cast UDA as a pseudo-label refinery problem in the challenging source-free scenario. We propose Negative Ensemble Learning (NEL) technique, a unified method for adaptive noise filtering and progressive pseudo-label refinement. NEL is devised to tackle noisy pseudo-labels by enhancing diversity in ensemble members with different stochastic (i) input augmentation and (ii) feedback. The latter is achieved by leveraging the novel concept of Disjoint Residual Labels, which allow propagating diverse information to the different members, leading to a superior noise resilience and a stronger consensus. Such consensus on a new, possibly cleaner, pseudo-label enables ensemble members to gain extra knowledge (in terms of better subsequent feedback) from each other. Eventually, a single model is trained with the refined pseudo-labels, which leads to a robust performance on the target domain.

Chapter 5 describes our method extending the previous work by incorporating *Continual Learning* in Source-Free UDA. Source-Free UDA methods work under the constraining, yet very realistic assumption, *i.e.,* they assume the availability of a pre-trained source model along with unlabeled target data, but no access to source samples. Such approaches inherently exhibit catastrophic forgetting as they boost the performance on target while completely disregarding the (inaccessible) source. In this context, we address the challenging task of adapting a source model to a target domain without forgetting the source, yet assuming no access whatsoever to it. We propose a novel Continual Source-Free UDA (CSF-UDA) framework comprising two main stages: i) a Source-Free UDA pipeline based on pseudo-label

refinement, which produces cleaner target pseudo-labels (and thus guarantees good target performance); ii) a procedure for extracting class-conditioned source-style images by leveraging source model, with target data and its refined pseudo-labels as a prior. Eventually, a single model trained with synthetic-source and real-target images ensures good performance on both domains. The collaboration is indirect in this case, as the source model is the only means that can be used to generate source-style synthetic images, which ultimately assists the final model in maintaining good performance on both domains.

Finally, Chapter 6 draws conclusions and discusses the future directions.

1.2 List of Publications

The work presented in this thesis has produced the following publications:

Publications:

- Waqar Ahmed, Andrea Zunino, Pietro Morerio, and Vittorio Murino. Compact CNN structure learning by knowledge distillation. In 2020 25th International Conference on Pattern Recognition (ICPR), pages 6554–6561. IEEE, 2021.
- Waqar Ahmed, Pietro Morerio, and Vittorio Murino. Cleaning noisy labels by negative ensemble learning for source-free unsupervised domain adaptation. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 1616-1625, January 2022.

Submissions:

- Jacopo Cavazza, Waqar Ahmed, Riccardo Volpi, Pietro Morerio, Francesco Bossi, Cesco Willemse, Agnieszka Wykowska, Vittorio Murino. Understanding action concepts from videos and brain activity through subjects consensus. Submitted at Nature Machine Intelligence Journal, January 2022.
- Waqar Ahmed, Pietro Morerio, and Vittorio Murino. Continual source-free unsupervised domain adaptation. Submitted at Proceedings of the IEEE/CVF Conference 2022.

Chapter 2

Compact Convolutional Neural Network Structure Learning by Knowledge Distillation

2.1 Introduction

Recent years have seen remarkable performance breakthroughs achieved in machine learning [17] and computer vision applications using deep Convolutional Neural Networks (CNNs) [18, 19, 20]. However, the CNNs are computationally expensive, memory-intensive, and power hungry. Therefore, extraordinary inference speed, throughput, and energy efficiency are required to meet the real-time application's demands running on resource-constrained devices such as drones, robots, smartphones, and wearable devices [21].

Researchers have demonstrated the possibility of using an automated architecture search approach to discover an optimal CNN structure for the task of interest. Yet, it is an impractical method that requires a huge architecture searching time in finding a reasonable solution due to the combinatorially large search space [22, 23]. Another possible proposed direction is to design lightweight CNN architectures that typically requires expensive, frequently manual, trial-and-error exploration to find a good solution. However, CNN customized for a particular task fails to maintain the required performance in other tasks: thus, a similar exercise is needed to target every new problem. Nevertheless, former methods do not consider resource constraints (*i.e.* Floating Point Operations per Inferrene (FLOPs) and Model-Parameters) and are not scalable for growing task complexity.

To overcome these challenges, we propose a powerful and adaptive method for learning an optimal network structure for the task of interest. Our approach advances the spirit of recently proposed method *MorphNet* [24] which has the advantage of being fast, scalable and adaptive to specific resource constraints (*e.g.,* FLOPs or Model-Parameters). Most importantly, it *learns* network structure during training. However, the current optimization technique has an intrinsically *biased concentration* that either pushes the optimizer to focus on high-resolution layers (towards network input) or focus more on low-resolution layers (towards network output) when optimized for FLOPs or Model-Parameters, respectively. Consequently, it leads to an sub-optimal network structure and reduced model performance. Thus, [24] employs a

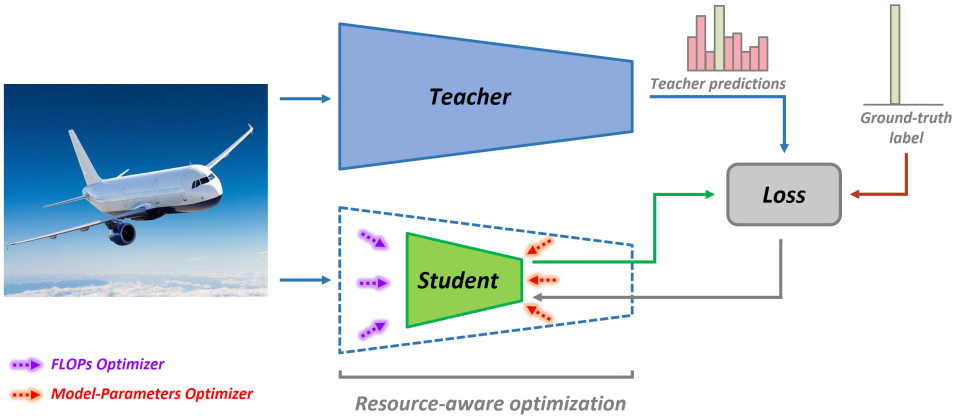


Figure 2.1: **Overview of the proposed method.** Our method assumes a teacher network which is usually the network to be compressed itself. The resource-aware optimization employs FLOPs and model-parameters optimizers on suitable parts of the student network with respective budget constraints. While it relaxes the task complexity, privileged information imposes control over predictions to preserve superior model performance during network structure learning.

width multiplier to uniformly expand all layers to improve model performance which eventually results in a similar resource-intensive network structure.

To mitigate these shortcomings, our method employs *Resource-Aware Optimization* augmented with the *Privileged Information (PI)* technique in a student-teacher scheme (see Fig. 2.1). The proposed resource-aware optimization breaks down the seed network in smaller instances which curtails task complexity to learn better end-to-end network structure. Eventually, it enables customized optimization of each stage of the network with specific budget constraints. As several works have already proved PI's potential in improving model performance [25, 26, 12], in our case, it augments our method's capability by imposing control over model performance during optimization considering the teacher network performance as a target. This facilitates the optimizer to maintain high model performance while learning the lightweight network structure. Note that our method does not apply network expansion at all, and the student network to be compressed utilizes PI extracted almost for free from the uncompressed network itself.

The hybrid of the above-mentioned strategies leads to a superior and consistent compression results over a variety of network architectures (*e.g.,* ResNet101 and MobileNet_v2) and datasets (*e.g.,* CIFAR-10, CIFAR-100, and ImageNet). The proposed method is novel, effective and carefully devised to target specific limitations of the existing method. As a result, an optimal network structure is discovered which is also capable of delivering better model performance. In particular, for image recognition task using the already compact

network MobileNet_v2 on CIFAR-10 benchmark, our method achieves 2× and 5.2× better model compression than [24], in terms of FLOPs and model-parameters, respectively (see Fig. 2.3a). Especially, the resultant network delivers 80.38% classification accuracy which is 1.05% better than the baseline teacher network.

2.2 Related Work

Design of computation, memory, and power-efficient CNNs are needed to deploy deep learning applications on embedded devices such as drones, robots, and smartphones [21]. One way of achieving this objective is to transfer knowledge from a deep, wide and complex *teacher* network to a shallow *student* network [14, 12, 27, 28]. In principle, a teacher network is trained in advance. Afterward, a lightweight student network is trained to mimic the behavior of the teacher network in an equally effective manner. Also, a quantized distillation approach was proposed in [29], whereas [30] suggests an adversarial learning process for model compression. However, in existing works, distilling the generalization ability of complex teacher into a smaller network cost superfluous FLOPs and model-parameters depending on *predefined fixed structure* of the student. Differently, we propose an approach leveraging privileged information to dynamically learn an optimal network structure of a student while respecting the given resource constraints for the task of interest.

Some recent works achieve model compression by pruning redundant connections [31, 32] or using low-precision/quantized weights [29, 33]. While others propose a precise design of efficient CNN architectures by inverting residual connections between the thin bottleneck layers [34], grouping point-wise and depth-wise dilated separable convolutions [35], or utilizing pointwise group convolution and channel shuffle [36]. However, the designing of efficient CNNs approaches is not scalable and requires extensive human efforts to target every new problem, dataset or platform. On the contrary, our proposed method automatically *learns* a lightweight network (student) structure, sufficient to deliver a comparable performance of a given large and complex (teacher) network.

Some related methods exist, such as [37] which progressively simplifies a pre-trained CNN by generating network proposals during training until the resource budget is met. Auto-Grow [38] automates the process of depth discovery in CNNs by adding new layers in shallow seed networks until the required accuracy is observed. However, our proposed approach is inspired by MorphNet [24], an open-source tool for learning network structure based using resource weighted sparsifying regularizer. Among all state-of-the-art works, this method is fast, scalable and adaptable to specific resource constraints (*e.g.,* FLOPs or model-parameters). However, the optimization comes with a drawback, *i.e.* the more you iterate, the more you observe a drop in accuracy. To recover performance loss, all layers are uni-

formly expanded using a width multiplier which may lead to an improved but large resource-consuming network structure. To mitigate these limitations, our method utilizes *Privileged Information* along with *Resource-aware Optimization* to improve the network structure learning process. Without any network expansion, the proposed method offers superior FLOPs and model-parameters reduction along with better model performance.

2.3 Method

We propose a framework that learns an optimal CNN structure to efficiently target the task of interest, considering allowed resource constraints *e.g.,* FLOPs and model-parameters while preserving high model performance. We advance the spirit of MorphNet which is based on a training procedure to optimize CNN structure. Since it does not represent a simple pruning or post-processing technique, the method is well suited for our task. The model compression method of MorphNet [24] relies on a regularizer $\boldsymbol{\mathcal{R}}$. It induces sparsity in activations by putting greater cost $\boldsymbol{\mathcal{C}}$ on neurons contributing to either FLOPs or the model-parameters. The network sparsity is measured on the basis of batch normalization scaling factor γ associated with each neuron *i.e,* if γ lies below than the user-defined threshold, the corresponding neuron is considered as dead and can be discarded (since its scale is negligible). Both the FLOPs and model-parameters are influenced by the particular layer associated with matrix multiplications - *i.e.* convolutions. This makes sense, as the lower layers of the neural network are applied to a high-resolution image, and thus consume a large number of the total FLOPs. Whereas, the upper layers typically comprises of larger number of channels and thus contain abundant weight matrices. We can define separate cost functions as follows:

$$\boldsymbol{\mathcal{C}}_{FLOP} = \sum_{k=1}^{K} [C_{in}^{k} * (w^{k})^{2} * C_{out}^{k} * S_{out}^{k}] \tag{2.1}$$

$$\boldsymbol{\mathcal{C}}_{PARAM} = \sum_{k=1}^{K} [C_{in}^{k} * (w^{k})^{2} * C_{out}^{k}] \tag{2.2}$$

where K is total number of layers and k is the layer index, w^2 is the kernel size, C_{in} is the number of input channels, C_{out} is the number of output channels, and S_{out} is the size of the output layer (*i.e.* the number of times the kernel is applied). In this study, we propose to use two different regularizers $\boldsymbol{\mathcal{R}}_{FLOP}$ and $\boldsymbol{\mathcal{R}}_{PARAM}$, depending on the resources being optimized in a particular stage of the network. So the optimization problem is equivalent to applying a penalty on the loss as follows:

$$\min_{\theta} \boldsymbol{\mathcal{L}}(\theta) + \alpha \boldsymbol{\mathcal{C}}_{j}(\theta), \; j = \{FLOP, PARAM\}. \tag{2.3}$$

where $\mathcal{C}$ is a function of the model-parameters θ and the hyperparameter α regulates the resource optimization intensity. For comparison, we refer to MNF and MNP as the original MorphNet method which optimizes the entire network for FLOPs and model-parameters, respectively.

The network structure obtained using stand-alone MorphNet costs a significant drop in model performance. This happens because its optimization is either based on Eq. (2.1) or (2.2) that leads to a biased concentration that either forces the optimizer to focus on high-resolution layers (towards network input) or focus on low-resolution layers (towards network output) when optimized for FLOPs or model-parameters, respectively (see Fig. 2.2). Consequently, learning structure of the entire complex network with such a biased method leads to a sub-optimal solution and a reduced model performance. To recover performance loss, the existing method uniformly expands all layer sizes using a width multiplier which may lead to a better but large resource-consuming network structure.

To overcome these challenges, our method employs *Resource-aware Optimization* augmented with the *Privileged Information (PI)* technique in a student-teacher scheme (see Fig. 2.1). The resource-aware optimization breaks the complex task of learning the entire CNN structure into comparatively simpler sub-tasks. Subsequently, appropriate optimization is performed on each stage of the network with specific budget constraints (e.g., either FLOPs or model-parameters). Eventually, our approach discovers a global network structure that is lighter than the original end-to-end solution.

In addition to that, the privileged information framework [12], augments our method's capability by imposing control over model performance during optimization considering the teacher network performance as a target. This facilitates the optimizer to maintain high model performance while learning the optimal network structure. Note that our method does not apply network expansion at all, and the student network to be compressed utilizes PI extracted almost for free from the uncompressed network itself. In this way, the impact on performance is also accounted along with the existing sparsity measure that helps the optimizer to remove only the least significant neurons from the network.

2.3.1 Leveraging Privileged Information

A pre-trained teacher network f_t is cloned to serve as a student network f_s (a network to be compressed). Although the architecture of the teacher can be different, the choice of the same architecture eliminates the requirement of training an additional teacher network. As depicted in Fig. 2.1, the structure of the student network is optimized to meet the required resource budget while taking advantage of the soft predictions (from the teacher) along with the ground-truth labels. This forces the student network to keep mimicking baseline predic-

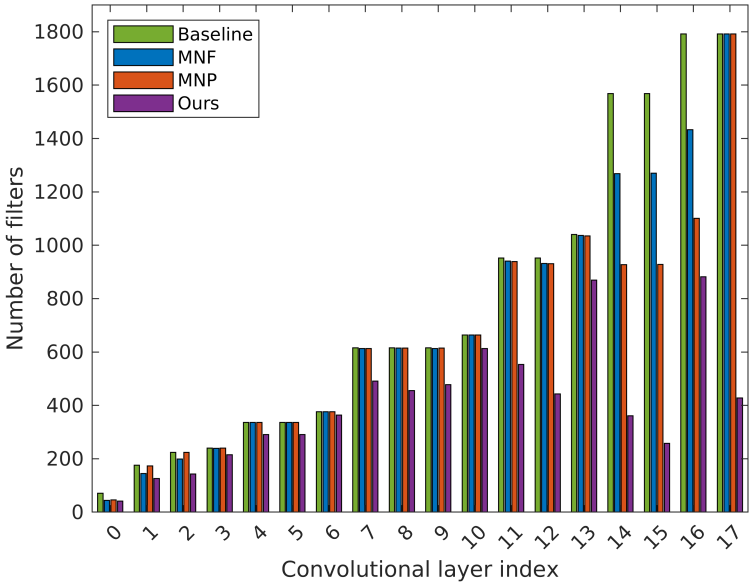


Figure 2.2: **MobileNet_v2 structure optimized on CIFAR-100.** The percentage of dropped filters from each convolution stage of MobileNet_v2 is presented. Comparing with baseline (outright number of filters), it can be observed that FLOP regularizer (MNF) of the existing method tends to remove filters from the lower layers near the input, whereas the model-parameters regularizer (MNP) tends to remove more filters from upper layers near the output. On the contrary, in terms of model compression, our proposed method clearly outperforms the existing method by a large margin over all stages of the network. Please refer to Fig. 2.3b for the accuracy comparison.

tions during optimization. In principle, both networks are trained to achieve the same task - *i.e.* infer the identical class of input image x^i. The training is accomplished by minimizing the following cross-entropy loss:

$$\boldsymbol{\mathcal{L}}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N} l\big(y^i, f(\mathbf{x}^i, \boldsymbol{\theta})\big) \tag{2.4}$$

where $\boldsymbol{\theta}$ are parameters of the model, y^i is the ground-truth label of sample i, $f(\cdot)$ symbolizes the activation function, $\mathbf{x}^i = \{x_1^i, x_2^i, \ldots, x_m^i\} \in \mathbb{R}^m$ denotes a training sample, and l is a loss to measure the prediction error. During the inference of y^i given x^i, $i = 1, \ldots, N$, the student network leverages privileged information z^i about the sample (x^i, y^i). Such additional information is derived from the teacher model's prediction:

$$z^i = \sigma(f_t(x^i)/T) \tag{2.5}$$

where σ symbolizes the softmax operator and $f_t(x^i)$ refers to the teacher logits. Thus, the student network is trained according to the following optimization problem:

$$f_s = \arg\min_{f \in F_s} \frac{1}{N} \sum_{i=1}^{N} [(1-\lambda) l(y^i, \sigma(f(x^i))) + \lambda l(z^i, \sigma(f_t(x^i)/T))]. \tag{2.6}$$

The parameter T regulates the amount of smoothness applied to logits. This not only reveals commonalities and differences between classes to be discriminated but also exploits the true potential of the soft labels [12]. F_s is the student function hypothesis space and the parameter $\lambda \in [0, 1]$ is the imitation factor, controlling the student to mimic the teacher vs. to predict the ground-truth label. Therefore, the proposed approach is modeled by incorporating Eq. (2.6) into MorphNet's minimization equation (in Eq. (2.3)). Thus we get the following optimization problem:

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} [(1-\lambda)\, l(y^i, \sigma(f(x^i, \theta)/T)) + \lambda\, l(z^i, \sigma(f_t(x^i, \theta)/T)) + \alpha\, \boldsymbol{\mathcal{C}}_j(\theta)], \quad j = \{FLOP, PARAM\}, \tag{2.7}$$

where the standard cross-entropy loss incorporates both, the privileged information and the MorphNet optimizations. In principle, incorporating privileged information from the uncompressed method is sufficiently general to be applied to any compression algorithm which can be expressed in the form of Eq. (2.3).

2.3.2 Resource-aware optimization

In CNNs, the input image is processed by progressively reducing the feature map's resolution while increasing the number of filters to be applied along with the network. Therefore, lower layers carry higher *FLOPs Cost*, while higher layers account for huge model-parameters (see Eq. (2.1)&(2.2)).

However, Fig. 2.2 shows that layers of MobileNet_v2 are reduced differently according to the type of optimization (MNF or MNP) performed during training. This suggests that different optimizers on lower and upper layers are needed in order to be more effective in network structure learning. For this reason, we propose a resource-aware optimization scheme to optimize different layers of the network using suitable optimizer which leads to following

modification to Eq. (2.7):

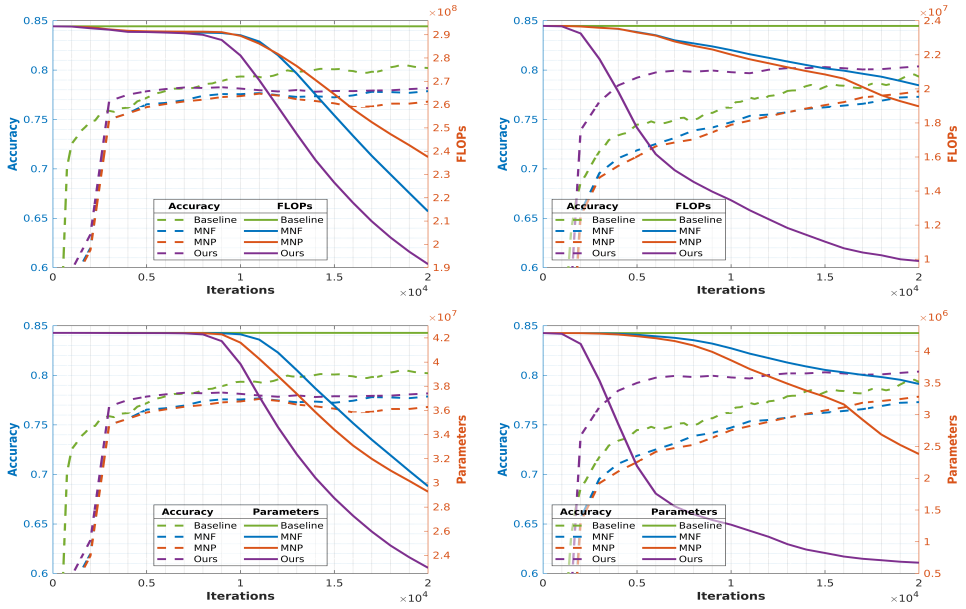
$$\min_{\theta_1}\min_{\theta_2}\frac{1}{N}\sum_{i=1}^{N}[(1-\lambda)\,l(y^i,\sigma(f(x^i,\theta_1,\theta_2)/T)) \\ +\lambda\,l(z^i,\sigma(f_t(x^i,\theta_1,\theta_2)/T)) \\ +\alpha\,(\boldsymbol{\mathcal{C}}_{FLOP}(\theta_1)+\boldsymbol{\mathcal{C}}_{PARAM}(\theta_2))], \tag{2.8}$$

where $\theta_1 \cup \theta_2 = \theta,\ \theta_1 \cap \theta_2 = \varnothing$ is a partition of the weights parametrizing the lower (*i.e.,* block 1&2 of ResNet101) and upper (*i.e.,* block 3&4 of ResNet101) layers of the network, respectively. Specifically, we propose a configuration in which the first half of the network is optimized for FLOPs and the second half is optimized for model-parameters.

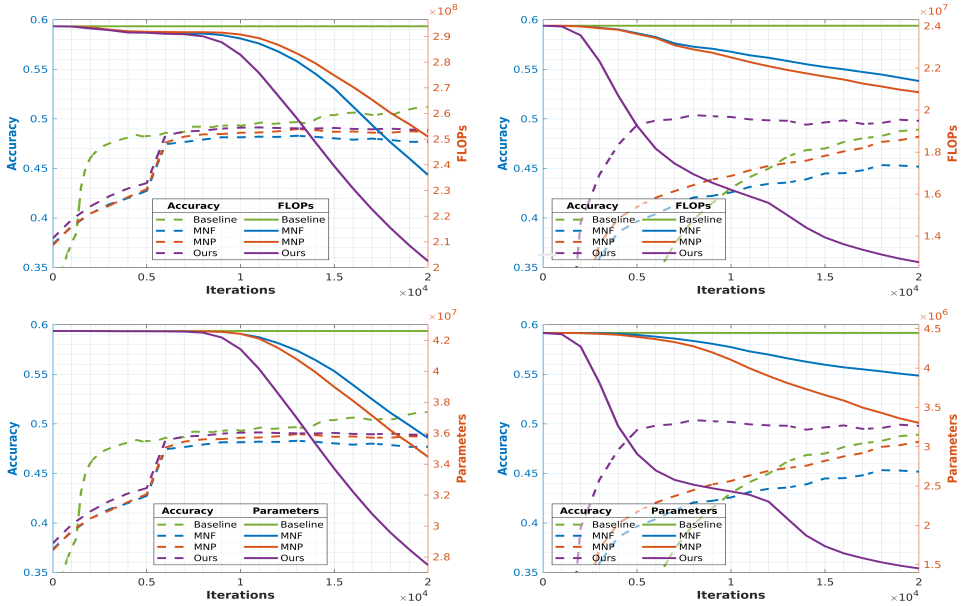
2.4 Experiments

We choose two widely used networks namely ResNet101 [18] and MobileNet_v2 [34] to examine the generalizability of proposed method on varied architecture designs. These networks were hand-crafted to achieve two distinct goals: the former was designed to obtain high accuracy while the latter was designed to yield low computation expense on mobile devices. Thus, we perform an extensive evaluation with these networks to show the effectiveness and adaptability of the proposed method (see Tab. 2.1). We evaluate our proposed method on three standard classification datasets: CIFAR-10 [39], CIFAR-100 [39] and Imagenet [40]. CIFAR-10 and CIFAR-100 datasets consist of 60000 tiny images of dimensions 32×32, categorized into 10 and 100 distinct classes respectively. Both datasets comprise $50K$ images in the trainset and $10K$ images in the testset. The ImageNet dataset consists of $1M$ images in train and $50K$ images in test split of dimensions 224×224, categorized into 1000 classes. In experiments, the teacher network is trained on target datasets in advance and kept fixed throughout the training process. Subsequently, an identical network architecture is considered as a student to discover its compact structure form capable of delivering competitive performance. For resource-aware optimization, we empirically divide the end-to-end optimization task into two sub-tasks by partitioning the network weights into two groups – lower half and upper half, although our method is generalizable to a higher number of sub-tasks.

Our implementation is based on TensorFlow and open-source tool MorphNet [24]. We use Adam optimizer with a fixed learning rate of 10^{-4} for CIFAR-10 and CIFAR-100 and RMSProp optimizer for ImageNet with an initial learning rate of 10^{-4} that decays by a factor of 0.98 every 2.5 epochs. In each experiments, the student network is trained for $20K$ iterations with a mini-batch size of 100 for CIFAR-10 and CIFAR-100. For ImageNet, ResNet101



(a) Results on CIFAR-10



(b) Results on CIFAR-100

Figure 2.3: We compare FLOPs and model-parameters reduction trend for CIFAR-10 and CIFAR-100 benchmarks considering ResNet101 (left) and MobileNet_v2 (right) as backbone networks. **MNF** and **MNP** are variants of existing method to exclusively optimize network structure for FLOPs and model-parameters, respectively. For ResNet101, our method outperforms the existing method in FLOPs and model-parameters reduction with slightly better model performance. Especially, for already compact network MobileNet_v2, our method brings superior network compression even with accuracy higher than the baseline.

Table 2.1: The results are reported on CIFAR-10 and CIFAR-100 with two different backbone CNNs. α=regularization-strength, **ACC**=accuracy on testset (%), **RED**=reduction achieved (%), **MNF**,**MNP**=existing methods. Our proposed method offers an optimal solution for both FLOPs and model-parameters reductions, so it presents two RED columns, accordingly to the optimization considered. Our method outperforms existing one in terms of compression, with comparable or marginally lower accuracy (cases in **bold**) or even with higher accuracy (cases in **bold**). Results show proposed framework's consistency, robustness, and generalizability.

(a) **Results on MobileNet_v2.**

CIFAR-10 Baseline - ACC:84.9, FLOPs:2.37×10^7, Model-Parameters:4.29×10^6.
CIFAR-100 Baseline - ACC:55.1, FLOPs:2.40×10^7, Model-Parameters:4.45×10^6.

		Optimization for FLOPs				**Optimization for model-parameters**		
	α	MNF		**Ours**			MNP	
		ACC	RED	ACC	RED	RED	ACC	RED
CIFAR-10	1	77.8±0.8	6.0±0.1	82.8±0.2	**23.9**±0.2	**55.6**±0.4	77.6±0.1	9.4±0.3
	5	77.8±0.9	7.6±0.2	82.1±0.3	**39.6**±0.2	**73.0**±0.2	77.6±0.5	16.0±0.6
	10	77.2±0.4	9.5±0.2	81.4±0.2	**48.9**±0.9	**78.8**±0.7	78.0±0.8	31.0±1.9
	15	77.5±0.4	11.8±0.3	80.2±0.4	**53.1**±1.6	**81.4**±1.2	78.2±0.2	39.9±0.3
	20	77.4±0.1	14.7±0.1	80.3±0.2	**58.3**±0.5	**84.3**±0.1	77.8±0.5	44.4±0.6
	25	76.9±0.3	17.3±0.3	79.5±1.2	**61.5**±0.2	**86.2**±0.1	77.9±0.4	48.2±0.4
CIFAR-100	1	46.8±0.4	5.4±0.0	55.6±0.2	**10.5**±0.3	**18.7**±0.9	47.2±0.7	6.4±0.4
	5	46.5±0.5	6.1±0.1	53.6±0.4	**20.8**±0.4	**36.0**±0.4	47.3±0.9	10.0±0.6
	10	46.7±0.4	6.9±0.1	52.1±0.4	**29.1**±0.3	**48.8**±0.3	47.4±0.2	14.6±0.2
	15	46.5±0.6	7.8±0.5	51.0±0.5	**35.4**±0.7	**56.5**±0.9	47.1±0.5	18.3±1.3
	20	45.1±0.8	9.1±0.3	50.5±0.7	**39.9**±1.1	**61.4**±0.9	48.0±1.0	21.7±1.6
	25	45.3±1.1	11.0±0.6	49.6±0.6	**46.9**±0.3	**67.3**±0.5	48.6±0.0	25.7±0.8

(b) **Results on ResNet101.**

CIFAR-10 Baseline - ACC:80.3, FLOPs:2.94×10^8, Model-Parameters:4.24×10^7.
CIFAR-100 Baseline - ACC:52.2, FLOPs:2.94×10^8, Model-Parameters:4.26×10^7.

		Optimization for FLOPs				**Optimization for model-parameters**		
	α	MNF		**Ours**			MNP	
		ACC	RED	ACC	RED	RED	ACC	RED
CIFAR-10	1	79.9±0.6	2.0±0.4	79.8±2.0	**4.4**±0.5	**7.4**±0.7	80.2±0.4	5.9±0.7
	5	78.7±0.6	14.1±3.8	77.3±0.6	**18.3**±0.4	**27.9**±0.6	79.8±0.9	26.7±1.2
	10	77.4±0.9	27.1±4.9	78.6±1.3	**34.8**±4.8	**45.8**±4.5	77.2±1.7	31.0±6.7
	15	76.7±0.5	35.4±0.8	77.1±0.5	**41.8**±2.1	**53.9**±1.8	77.6±2.5	48.1±6.8
	20	76.0±3.1	43.5±7.5	77.5±0.7	**50.0**±6.7	**61.6**±5.3	78.6±1.3	56.1±2.5
	25	76.3±0.9	48.7±1.6	76.3±1.6	**55.3**±2.6	**66.9**±1.7	76.0±2.2	58.5±1.3
CIFAR-100	1	50.5±2.4	1.0±0.1	50.3±0.8	1.0±0.0	0.1±0.0	50.0±0.9	0.1±0.0
	5	50.8±2.0	4.2±1.7	50.7±1.1	**7.8**±3.2	**8.1**±3.6	49.9±0.5	2.0±0.4
	10	48.6±1.0	8.1±1.3	48.4±0.4	**17.7**±2.0	**20.5**±2.0	47.9±1.1	10.3±2.9
	15	46.7±0.4	19.7±4.6	48.1±0.4	**31.1**±3.5	**35.5**±3.2	48.5±0.6	19.1±5.3
	20	47.2±1.2	22.2±1.7	46.2±1.0	**32.1**±1.4	**38.7**±1.3	48.0±1.7	26.0±4.6
	25	44.5±2.3	29.7±2.8	47.5±1.3	**43.8**±3.4	**51.1**±3.1	45.9±2.2	34.9±1.4

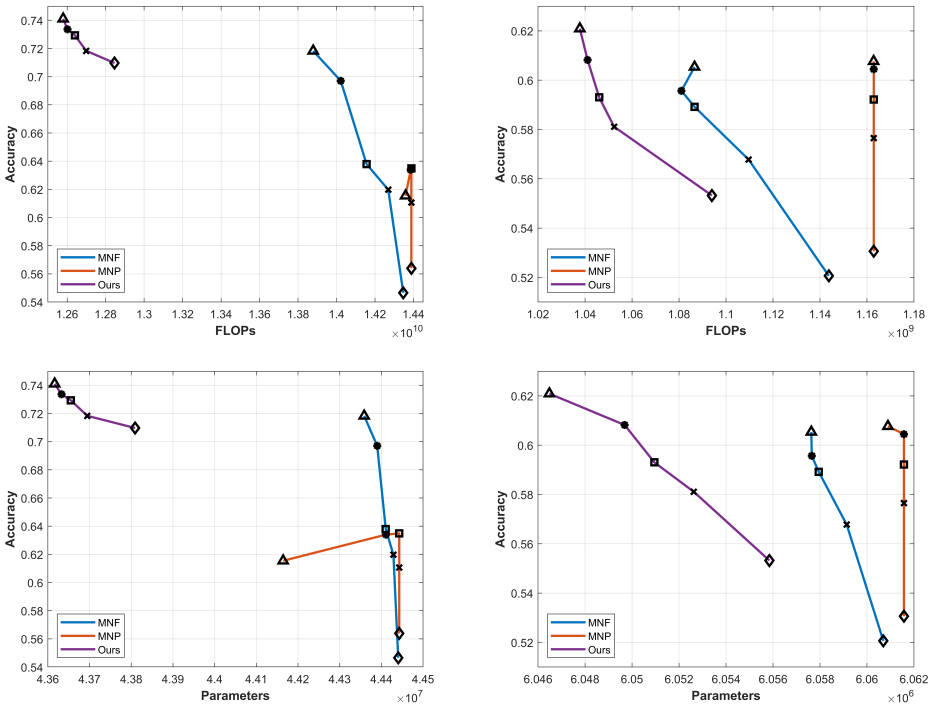


Figure 2.4: **Results on ImageNet.** We compare FLOPs and model-parameters reduction trend for ResNet101 (left) and MobileNet_v2 (right). **MNF** and **MNP** are variants of existing method to optimize network structure for FLOPs and model-parameters, respectively. We present accuracy vs. FLOPs/parameters results at $100, 200, 300, 400$ and 500 thousand iterations marked as $\diamond, \times, \square, *$ and $\triangle$, respectively. For ResNet101, our method outperforms the existing method by a large margin in terms of FLOPs and model-parameters reduction with superior model performance.

and MobileNet_v2 models are trained for $500K$ iterations with mini-batch size of 32. In the network structure learning process, fixed number of iterations is essential for comparison - *i.e.* more or less iterations can lead to different network structure and performance. We used different values of regularization-strength α with fixed $\gamma = 0.01$ to guarantee fair comparison with [24]. For distilling knowledge, we use $T = 10$ and $\lambda = 0.5$. After training, the performance of the student network is evaluated over the entire test-set to observe the effectiveness of learned lightweight network structure. We compare our proposed method with the stand-alone variants of MorphNet, namely MNF (entire network optimized for FLOPs) and MNP (entire network optimized for model-parameters). The results are reported in terms of Accuracy (ACC) and the FLOPs/model-parameters Reduction (RED) metrics - *i.e.* the percentage reduction with respect to the *FLOPs* or *model-parameters* cost of the teacher network. All parameters were kept consistent for entire experiments and results are averaged over three runs.

2.5 Results

In this Section we demonstrate that our method consistently outperforms the existing method by substantial margins, both, in terms of FLOPs and model-parameters reduction while offering better model performance. Targeting networks of different capacities, we present results for the two image recognition datasets CIFAR-10 and CIFAR-100 in Tab. 2.1 with varying compression intensity steered by the parameter α. In each sub-caption, we report the performance achieved by the teacher network in terms of accuracy and original *FLOPs/model-parameters* cost after been trained for the same image recognition task. We compare the proposed method with the stand-alone MorphNet MNF (entire network optimized for FLOPs) and MNP (entire network optimized for model-parameters). We demonstrate that our method brings better compression-performance tradeoff over all regularization strength α considered.

In particular, evolution trend during training for CIFAR-10 in Fig. 2.3a (right) shows that our method is relatively more effective for an already compact network (MobileNet_v2) with $2\times$ better FLOPs reduction than MNF and $5.2\times$ better model-parameters reduction than MNP. Such substantial gain in compression is achieved with up to 1.05% better recognition accuracy than the baseline teacher network. The most notable difference in trends is that our method brings superior model compression and performance right from the beginning and keeps on fine-tuning over successive iterations. Similarly, for ResNet101 in Fig. 2.3a (left), our method learns network structure capable of delivering slightly better performance while being $1.1\times$ and $1.3\times$ more compressed in terms of FLOPs and model-parameters, respectively, than the existing method.

Also, Fig. 2.3b shows the same consistent trend for CIFAR-100 on both networks. For MobileNet_v2 in Fig. 2.3b (right) our method brings $1.7\times$ and $2.7\times$ better compression in terms of FLOPs and model-parameters, respectively with 0.88% better recognition accuracy than the baseline. For ResNet101 in Fig. 2.3b (left), our method learns network structure capable of delivering slightly better performance while being $1.2\times$ and $1.3\times$ more compressed in terms of FLOPs and model-parameters, respectively, in comparison with the existing method.

Finally, we report experiments considering the popular large scale ImageNet dataset on both networks. Fig. 2.4 shows the accuracy versus FLOPs (top row) and model-parameters (bottom row) reduction. We present results achieved after $500K$ iterations with $\triangle$ along with intermediate results (*i.e.* results after $100, 200, 300$ and 400 thousand iterations marked as $\diamond, \times, \square$ and $*$, respectively) obtained during network structure learning. As expected from the insights discussed in Sec. 2.3.2, the proposed method works best in terms of learning optimum network structure. Since lower layers are optimized for FLOPs and higher layers for

model-parameters, the structure of each block is optimized accordingly to the most suitable resource constraint. For MobileNet_v2 in Fig. 2.4 (right), our method brings outstanding model compression along with higher classification accuracy even in the cases where MNP has not started the optimization yet. A similar trend is also confirmed for ResNet101 in Fig. 2.4 (left) in which our method obtains superior model compression after only $100K$ iterations that is way higher than what is achieved after $500K$ iterations using the existing method.

2.6 Summary

In this study, we present a resource-aware network structure learning method, which enables suitable optimization in different sections of the seed network considering FLOPs and model-parameters constraints - *i.e.* lower layers are optimized for FLOPs and higher layers for model-parameters. Furthermore, Our method leverages privileged information to impose control over predictions to preserve high-quality model performance. In an extensive evaluation of various network architectures and datasets, our method brings state of the art network compression that outperforms the existing method by a large margin while maintaining better control over the compression-performance tradeoff.

Chapter 3

Understanding Action Concepts from Videos and Brain Activity through Subjects Consensus

3.1 Introduction

Electroencephalography (EEG) measures the electrical activity patterns induced by the aggregation of excitatory/inhibitory post-synaptic potentials generated in cerebral cortex. Such data modality is helpful in registering and monitoring the brain activity, which can then be decoded and used in a variety of scientific, medical and other application domains.

Typically, EEG is the basic recording instrument to support brain-computer interfaces (BCIs), aimed at mediating between brain signals and downstream applications. For example, from a clinical perspective, there is a well established research direction towards decoding motor imagery (refer to [41] for a survey), so that motor neural impulses can be mapped and controlled with the ultimate goal of assisting, augmenting, or repairing human cognitive or sensory-motor functions. Industry is also making efforts in the design of "mind reading" devices in order to let users monitor their well-being [42], for example, to support meditation or facilitate the execution of daily activities [43, 44, 45, 46, 47, 48, 49, 50, 51].

In the literature, there is a substantial body of works where EEG is utilized in tandem with machine learning and computer vision to recognize useful patterns to face the task of interest. For instance, the recognition of emotions can effectively be addressed through EEG data: stimuli such as natural images or videos can be analyzed in order to figure out the induced emotional state, and measure valence, arousal and dominance factors (as in [52]), while also allowing a finer prediction of happiness, fear or disgust reactions (as in [53]). Furthermore, EEG data are also useful for the general purpose of object categorization, for instance, in the case of the recognition of characters displayed on a screen [54] or the classification of synthetic/natural images [46, 47].

We differ from previous works mainly devoted to read emotions or to decode mental processes related to the classification of what is explicitly visualized in the typical static

Figure 3.1: Example footages related to some of the selected actions for the task of concept understanding. The "cooking" action is represented by the smoke coming out from a pan, the creation of a dough, the mixing of a chocolate cream or the garnishing of a dessert (*top-left*). The action of "flying" is represented from footages in which pigs fly, fireworks are shot in the sky. Alternatively, a scene in which the panorama is filmed from an airplane window and Santa Claus is delivering presents (*top-right*). For "hugging", unusual cases are considered such as a man hugging his dog, a girl hugging a tree, two gibbons hugging each others and a baby hugging his toy (*bottom-left*). Similarly, for "throwing", elephants can throw sand on their backs, axes can be thrown, a weight can be thrown during a fitness sessions and eventually money can be thrown (*bottom-right*). While capturing EEG data out of this footage, we can register the mental processes which try to categorize videos which refer to a given action class, without trivially representing it in its most conventional case. Differently to the explicit visualization of an action, we can investigate what happens in the brain when trying to understand those videos and the action they imply.

stimuli (such as characters, digits or objects). In this study, we move a step forward in the comprehension of the potential associated to EEG data. In particular, we attempt to classify EEG signals related to higher levels of reasoning, associated to dynamic stimuli, and specifically devoted to the problem of *understanding the concepts behind an action*.

To this end, we built a dataset of EEG recordings acquired from 50 different subjects visually stimulated by videos from the recently designed Moments in Time (MiT) dataset [16], referring to the following 10 categories of actions: cooking, fighting, flying, hugging, kissing, running, shooting, surfing, throwing and walking. The peculiarity of MiT dataset

is the huge intra-class variability, which makes it one of the most challenging datasets that are currently available in computer vision [16]. For instance, prototypical videos explicitly visualizing a "flying" action might represent a bird or an airplane in the sky, but we are also considering videos representing such class in an indirect and implicit manner (*e.g.*, the ego-vision of the panorama seen from an airplane window), and even extreme and vague cases, such as exploding fireworks (see Fig. 3.1). In other words, the problem of understanding action concepts can be defined as classifying an action even when it is not explicitly displayed in a video footage, but, rather, only implied or represented in an abstract sense.

Since the video sequences associated with a same action can be very different in appearance and dynamics, it can be very difficult to classify by looking at the visual data only. Nevertheless, they have in common the same concept associated with the action which is hopefully captured by the brain activity and recorded in EEG data. In these terms, EEG naturally arises as an extremely convenient data modality to pair with video while attempting to solve the problem of actions' concept understanding.

This study explores the possibility of understanding action concepts in the brain activity stimulated by video footages and formulate the problem as a (multi-modal) classification framework. In fact, we posit that EEG is capable of capturing (some of) the mental processes responsible for action recognition and, to the best of our knowledge, our work is the first to demonstrate that understanding action concepts from EEG is a solvable and viable problem.

In former computer vision studies in which EEG data is used as an additional modality [46, 47, 48, 49, 50, 51], the selected tasks and corresponding benchmarks are relatively easier problems, like character [54] or object recognition [51]. In principle, both tasks could have been reliably solved without the help of EEG in the sense that, in these works, the performance obtained from EEG data is inferior to that obtained by directly processing the stimuli (*e.g.*, digits or images).

Differently, we claim that EEG is fundamental in our case because visual information is not always reliable for the sake of recognizing action concepts, which can be better understood from brain activity. Through a broad experimental validation considering state-of-the-art methods for video and EEG data processing, we obtained the experimental evidence that the performance achieved by leveraging EEG is superior to the one of video-only framework. In some sense, in the EEG data modality, there are humans in the loop *i.e.,* humans are involved indirectly in the sense that their brain activity is recorded while they watch the same video. In particular, the two sources of information are complementary, as we proved through a baseline of fusion methods. Therefore, we design a multi-modal computational method to take advantage of EEG data modality in order to boost the performance in classifying action

concepts from video data.

Our approach is rooted in the consideration that different subjects should share some common agreements about which video is prototypical for which actions, although, we should also account for personal differences. Therefore, a combination of EEG signals associated with a plurality of subjects translates into a peculiar inductive bias which has a regularizing effect in filtering out subject-specific nuances. As a consequence, we show that subjects' consensus boosts the generalization capabilities and favors the development of more accurate video action recognition algorithms, similarly to what happens when adopting an ensemble of models [55, 56, 57].

Furthermore, when averaging together the predictions of subject-specific action classifiers, we register a sharp gain in performance. Such a positive effect suggests that there exists an agreement across subjects while predicting actions from EEG data. We exploit this finding in a privileged information framework [12], where the consensus among the subjects yields to a teacher model supported by EEG data. A student model is then distilled by training it with ground-truth action labels as well as soft labels extracted from the teacher: the consensus that EEG data shown when pooling the acquisitions from different subjects is capable of boosting the performance of two state-of-the-art computer vision models, Temporal Relation Networks [1] and Temporal Shift Models [2], conventionally trained with videos only.

To recap, the main contributions of this work are the following.

- We introduce *Action Concepts*, a novel dataset of EEG recordings collected from 50 subjects stimulated by complex action videos taken from the MiT dataset [16]. In such footage, a given action is not always explicitly visualized but implied only. To perform action classification from such data, we posit that EEG is useful to capture brain activity for the sake of understanding what visual stimuli implicitly mean behind their visual appearance.

- We provide a broad experimental analysis of state-of-the-art algorithms in machine learning and computer vision, to demonstrate that the problem of understanding action concepts, originally tackled using visual data, is solvable by EEG data as well. Moreover, in terms of pure classification performance, we prove that EEG is a superior, yet, complementary data modality with respect to video.

- We explore to which extent different human subjects have commonalities in their high-level reasoning related to the process of understanding action concepts. We computationally demonstrate that the decision scores of action classifiers trained over EEG data across different subjects reinforces each other in removing subject-specific nu-

ances. Such ensemble provides a regularizing effect, capable of sharply raising the recognition accuracy using EEG data.

- We leverage subjects' consensus to distill from EEG a data-driven supervision which can be transferred to video sequences. Adopting a privileged information framework trained on the multi-modal combination of video + EEG data (and tested on video only), we show that the subjects' consensus enhances the performance of two state-of-the-art computer vision models: TRN [1] and TSM [2].

3.2 Related Work

The problem of "mind reading" has been considered for a variety of different applications by using a broad spectrum of sensory devices to register brain activity.

By means of blood oxygen level dependent (BOLD) signals measured through functional Magnetic Resonance Imaging (fMRI), recent studies [58, 59, 60, 43, 61, 62, 63, 64, 65, 66] have demonstrated the effectiveness of decoding the mental processes which are implied from a dynamical stimulus (like a video). In [62], authors exploit a Bayesian approach to reconstruct the video footage from the fMRI signal, while, in [63], fMRI was proved to be advantageous for the sake of action recognition in videos from single subject, when compared with classical computer vision approaches. Other than using EEG data, we differ from this approach by considering high-level action concepts and by using a large number of subjects to investigate whether the task-based human reasoning mechanism can be generalized. Video clustering was also considered [64] through a composite pipeline in which features extracted from fMRI are combined with wavelet transform, Gaussian process regression and spectral clustering. The problem of deploying a more high level decoding of the fMRI signal originated from a video stream is addressed in [65, 66], to find those most effective regions of brain responsible for parsing the semantics or reacting to emotions. Pain can be decoded from this kind of signal as well [67].

Similarly, capturing brain activity through Magnetoencephalography (MEG) has been successfully applied to a variety of applications. The genre of a movie (*e.g.,* comedy, romantic, drama or horror) can be decoded from MEG signals [68]. It is also possible to predict emotion from MEG data when performing a single-trial classification task for valence, arousal and dominance in short music video segments [52]. The problem of decoding motor imagery and create responses to brain stimulus which are capable of controlling a brain-computer interface (BCI) is tackled in [69]. MEG is used to predict the modality with which a word was presented to a subject as stimulus, written on a screen versus spelled in audio stream [53].

Although fMRI and MEG were used as suitable sensory devices for a variety of tasks (mainly related to emotions), they were not used to address the problem we are addressing. fMRI results involving a more complex experimental setup (as compared to EEG), and both fMRI and MEG addressed tasks characterized by a lower statistics (in terms of number of subjects and number of classes). Overall, none of the above works addressed specifically the classification of action concepts.

Electroencephalography (EEG) is a powerful registration tool for a variety of application. For instance, the problem of recognizing the affective and emotional content of a stimulus was tackled by a variety of works. Event-related potentials can show the connection between a selective processing of emotional stimuli and the activation of motivational systems in the brain [70]. Using gathered data under psychological emotion stimulation experiments, one can successfully train a support vector machine to disambiguate between emotions [71]. In [72], EEG data is employed to assess valence and arousal in emotion recall conditions, while comparing different encodings. Facial expressions and EEG are combined for the purpose of affective tags' generation in a multi-modal approach [73]. EEG conveys patterns related to what makes a movie trailer appealing or not for the audience [74]. Through domain adaptation [75], one can better transfer across different subject while recognize emotions evoked by images [76, 77]. The shift from static to dynamical stimulus (videos) in emotion recognition was investigated in [78, 79, 80].

EEG effectiveness for creating brain-computer interfaces (BCIs) has also been extensively studied. In [43], a system for rapid image search is devised, whereas EEG and motion capture can be combined in a multi-modal BCI [44], with the optional usage of deep learning to better fuse the two modalities [45]. Classical computer vision problems, such as object classification in images can be boosted when having access to an ancillary data modality. For the latter purpose, EEG induced from images shown as stimuli to subjects is effective to boost recognition capabilities, as shown in [46, 47, 48, 49, 50, 51].

As outlined above, EEG was mainly utilized to deal with emotions and BCI applications, and seldom considered dynamic stimuli for the sake of action concept classification aimed at understanding brain mechanisms related to (generalized action) recognition. This makes our work as pretty unique in the panorama of the multi-modal learning framework.

3.3 The *Action Concepts* dataset

In this Section, we present the phase of stimuli selection and subsequent acquisition of EEG recordings from a pool of selected participants. The details of each single step are reported in a separated subsection.

Table 3.1: Comparison of existing public benchmarks for EEG data processing, finalized to diverse applications.

Dataset Name	Task	Year	Type of stimuli	No. of stimuli	No. of electrodes	No. of classes	No. of subjects	Reference
Sleep EDF	Sleep monitoring	2000	–	–	4	–	25	[81]
CAP sleep	Sleep monitoring	2001	–	–	12	8	16	[82]
UCDDB	Sleep monitoring	2011	–	–	2	–	25	https://physionet.org/pn3/ucddb/
EEGmmidb	Motor imagery	2004	–	–	64	4	109	[83]
BCI comptetion	Motor imagery	2008	–	–	3	2	9	[84]
TUH	Seizure prediction	2012	–	–	24-36	–	100	isip.piconepress.com/projects/tuh_eeg
BB-EEG-DB	Seizure prediction	2012	–	–	32	–	5	[85]
CHB-MIT	Seizure prediction	2013	–	–	24	–	22	https://physionet.org/pn6/chbmit
OpenMIIR	Sound classification	2015	sound	12	66	2	12	[86]
KARA-ONE	Speech classification	2015	text	–	62	11	14	[87]
MindBigData - MNIST	Object classification	2015	images	60K	14	10	1	mindbigdata.com/opendb/index.html
Learning Human Mind	Object classification	2017	images	2K	64	40	6	[51]
MindBigData - ImageNet	Object classification	2018	images	14K	5	569	1	mindbigdata.com/opendb/imagenet.html
eNTERFACE	Emotion recognition	2006	images	327	64	3	16	[88]
DEAP	Emotion recognition	2011	video	120	32	3	16	[89]
MAHNOB	Emotion recognition	2012	video	40	16	2	30	[90]
SEED	Emotion recognition	2018	video	40	64	4	15	[91]
Action Concepts (*ours*)	*Action recognition*	*2020*	*video*	*240*	*64*	*10*	*50*	

3.3.1 Original characteristics of the dataset

In this work, we exploit EEG to handle dynamical stimuli which consist of 3 seconds video footages extracted from Moments in Time dataset [16]. By design, such dataset was created to guarantee remarkable inter-class and intra-class variations among actions, representing dynamical events at different levels of abstraction (*i.e.,* "opening" doors, drawers, curtains, presents, eyes, mouths, and even flower petals). While using videos from Moments in Time as stimuli for EEG, we attempt to investigate to which extent EEG can convey patterns capable of distinguishing video sequences which, despite their visual diversity, subsume the same category. In fact, very few datasets combine EEG with dynamical stimuli (and, for the latter, fMRI is usually preferred [62, 63, 64, 65, 66]).

Specifically, in Tab. 3.1, we compare our Action Concepts dataset with those already present in the literature, categorizing the application task (sleep monitoring, motor imagery decoding for BCI, seizure prediction for epilepsy, sound/object classification and emotion recognition). As first peculiar aspect, our dataset ensures high resolution considering a large number of electrodes (64). Second, in terms of key statistical features, existing datasets are extremely unbalanced in terms of number of stimuli vs. number of classes vs. number of subjects considered in the acquisition. In our case, we have 1) a large number of stimuli, which ensures high variability within the data, and also 2) more classes, making the classification problem inherently harder. Moreover, accounting for several subjects is beneficial when using EEG data since this guarantees the reliability and the statistical significance of the study. Overall, unlike the most of existing studies, our proposed dataset is extremely balanced with respect to the discussed crucial indicators.

The distinctive feature of our datasets is the targeted application. We do not just to recognize what is explicitly visualized in the stimuli adopted in the EEG data acquisition, but, rather, we attempt to recognize what is implicitly visualized in the stimuli itselves. For instance, consider the case of the *cooking* action in Fig. 3.1: there are scene statistics (a smoky pan, a dough, a chocolate cream or a even a chef) which are referring to the action of interest. Using EEG to capture the decoding process related to recognize that “a smoky pan/a dough/chocolate cream/chef is visualized in that frame” is clearly not enough to understand actions. Rather, the only chance to recognize the *cooking* action is to capitalize from the mental decoding process related to the fact that the aforementioned scene statistics implicitly refer to the action we seek to recognize.

To the best of our knowledge, our dataset is the first one designed for this application and it will be publicly released upon paper acceptance (hopefully at Nature Machine Intelligence).

3.3.2 Collecting stimuli: manual class selection

Among the 339 classes that compose Moments in Time dataset [16], we selected the following 10 classes: cooking, fighting, flying, hugging, kissing, running, shooting, surfing, throwing, walking. The decision was aimed at encompassing classical classes that are usually included among action recognition databases, spanning sports (walking, running, surfing), daily contexts (cooking) and human-to-human interactions (hugging and kissing). At the same time, we chose classes which have a similar execution, being only different in a specific details: the discriminant in between a walking and running action often lies in the speed with which the action is executed. Similarly, both throwing and shooting involve the fluctuation in the air of a physical entity (an object versus a bullet), with the crucial difference that shooting necessary requires a weapon whereas throwing doesn't. Finally, for either hugging or kissing, two people are closely interacting with each others. But, hugging consists of wrapping the hands around another person's neck, waist or back. Differently, kissing implies instead a face-to-face interaction. When using EEG to encode those actions, by successfully decoding such signals into discriminative classification patterns we can actually demonstrate to which extent EEG is sensible towards differences among classes, even when the differences are subtle.

3.3.3 Collecting stimuli: automatic video selection

Once the manual selection for the classes was completed, we exploited an automatic machine learning algorithm to decide which video to include within our analysis. We adopted a ResNet-50 convolutional neural network pre-trained on ImageNet [92], to process videos instead of images as done in [16]. From each video of the training set, 5 random frames are subsampled to fine-tune the ResNet-50's weights. During inference, a video is classified into one of the 10 classes considering 5 randomly sampled frames followed by a majority voting over the predicted labels. We selected 24 videos per class (240 videos overall) from the validation set of Moments in Time, by considering only the videos which were correctly classified by the model. Among them, we selected the 12 videos classified with maximal confidence (*i.e.,* sharpest softmax peak) and the 12 correctly classified with the lowest confidence. In this way, we account for the videos which can be clearly classified by the model as well as the more difficult footages for which the automatic selection stage is less confident (although still managing to achieve a correct inference). We believe that it is interesting to assess "what is easy/hard to understand" in a comparative scenario between an algorithm (in this case, a neural network) and a pool of human beings.

3.3.4 Selection of the participants

Fifty-three healthy participants (out of which 25 males) were recruited for the experiment (mean age: 23.8 $\pm$ 3.71 years). One participant was excluded from the analyses due to technical problems related to data quality. All participants provided written consent before enrolment in this study and were screened for contraindications to EEG. The exclusion criteria included the presence of a history of any neurological or psychiatric disease, use of active drugs, abuse of any drugs (including nicotine and alcohol) as well as any skin condition that could be worsened by the use of the EEG cap. The study was approved by the local Ethics Committee (Comitato Etico Regione Liguria) and was conducted in accordance with the ethical standards laid out in the 1964 Declaration of Helsinki. All participants had normal or corrected-to-normal vision and were right-handed.

3.3.5 Acquisition procedure

The participants were asked to sit in a dimly illuminated room, maintaining one meter distance from the screen. There, the EEG cap and EOGs were put on the head (see next section for details) and connected to the EEG amplifier. All the sections of the experiment were run using PsychoPy software [93]. First of all, participants' resting state activity (with open and closed eyes) was recorded. Subsequently, participants took part in another brief experiment (lasting approximately 15 mins), not relevant for this study. Before starting with the experiment, participants read the experimental instructions on screen and the experimenter asked for any possible questions or uncertainties. Participants were then presented with a practice part, during which they responded to videos belonging to the category "eat". Block and trial structures were identical to the experimental part. Participants then moved to the experiment, consisting of 5 blocks, one for each category, presented in a random order. Each participant responded to only 5 categories to avoid effects related to the long experiment duration. The categories were counterbalanced across participants, *i.e.,* half of the participants responded to cooking, fighting, flying, hugging, kissing categories, while the other half responded to running, shooting, surfing, throwing, walking categories. Each block consisted of the presentation of 24 videos per category in a random order. In the beginning of each block, participants were presented with the action category.

Trials were self-paced, *i.e.,* each trial was started by the participant by pressing the spacebar key on the keyboard: between two consecutive videos, an intermediate grey screen is shown to the participant. The grey screen displays the action category name and the line "press the spacebar to start the next video": this screen lasts until the participant pressed the spacebar. Each trial starts with 1 second inter-stimulus interval (ISI), presenting only a white fixation cross in the centre of the screen. Then, the video is presented in the centre of

the screen, superimposed by the white fixation cross. All videos lasted 3 seconds. In order to maintain the focus of the participants on the videos, an oddball-like [94] task was added during the video presentation. That is, in a random order between the streaming of the videos related to one of the classes of interest, 3 dummy videos where displayed. During the presentation of the dummy videos (3 for each category) the white cross turned red for 250 ms. This colour change was happening at a random moment between 1500 and 3000 ms after the beginning of the video. Participants were asked to press the spacebar as fast as possible when they noticed the colour change. These dummy trials were subsequently removed during the EEG analysis phase. The reader can refer to Appendix 3.7.2 for further details.

3.3.6 EEG data recording and pre-processing

EEG data were recorded using 64 Ag-AgCl electrodes of an active electrode system (ActiCap, Brain Products, GmbH, Munich, Germany) referenced to FCz. Horizontal and vertical EOG were recorded from the outer canthi of the eyes and from above and below the observer's right eye, respectively. The EEG signal was amplified with a BrainAmp amplifiers (Brain Products, GmbH), digitised at a 5000 Hz sampling rate for recording. No filters were applied during signal recording. Electrode impedances were kept below 10 kΩ throughout the experimental procedure. EEG data were analysed using MATLAB™ version R2018a and FieldTrip toolboxes [95]. Data were downsampled to 250 Hz and a band-pass filter (0.5–100 Hz) and a notch filter (50 Hz) were applied to extract the signal of interest and remove power line noise. Subsequently, data was segmented into epochs (*i.e.,* trials) from 0 to 5000 ms after the start of each trial. With this segmentation, data from one second before (ISI) and one second after each video were taken into account. Each trial was baseline corrected by removing the values averaged over a period of 1000 ms (from 0 to 1000 ms after the trial started, *i.e.,* the ISI). After visual inspection, trials affected by prominent artifacts (*i.e.,* major muscle movement and electric artifacts) were removed, and bad channels were deleted, (however its values are spherically interpolated using ICA so that, effectively, the number of electrodes is always the same across all the different participants). The signal was referenced to the common average of all electrodes [96], and independent component analysis (ICA) was applied to remove the remaining artifacts related to eye-blinks, eye movements, and heartbeat. After removing the remaining artifacts, noisy channels were spatially interpolated: at the end of this stage, a total number of 5339 EEG recordings was obtained.

3.4 Baseline experiments for the different data modalities

In this Section, we provide the baseline experiments for action classification either using EEG or video data modality (or a multi-modal fusion of the two). Additional technical and implementations details are provided in Appendix 3.7.1.

3.4.1 Baseline methods for EEG sequences

We adopted several alternative descriptors to encode EEG data for the sake of action recognition. First, we exploited two descriptors for a frequency-response analysis, based on Fast Fourier Transform (FFT) or Differential Entropy (DE) [97], respectively. For FFT, we compute the magnitude of real and imaginary part of Fourier coefficients. Then, we averaged over 4 classical frequency bands of interest: theta (5–7 Hz), alpha (8–13 Hz), beta (14–30 Hz) and gamma (31–60 Hz). For DE, we applied a band-pass filter in correspondence of any of the prior frequency bands. Subsequently, we inversely map the filtered signal in the temporal domain. As shown in [97], the DE of the resulting signal can be estimated by the logarithm of its temporal variance.

We also adopted a time-frequency response analysis by computing Morlet Wavelet coefficients on different frequency bands (theta, alpha, beta and gamma), where the temporal window is the one in which the video is shown to the subject. For either FFT, DE or Wavelet encoding, we performed a preliminary baseline removal stage in which we computed the temporal average of the representation between -700ms and -100ms before the actual start of the video. Such temporal average is then subtracted to normalize the descriptors used for the classification stage based on a linear support vector machine.

As an alternative class of feature encodings, we also explored the usage of feature learning algorithms, to directly learn representations from the data to be used for classification. In order to capture temporal dependencies within EEG data, we took advantage of a recurrent neural network with long-short term memory units (LSTM) [98]. In addition to a vanilla LSTM model, we also explored two variants. In the first one, a two branched network is used: one branch is composed by a (vanilla) LSTM, the other one is designed as a deep network performing stacked temporal convolutions. At the end, the decisions of the two branches are merged [99]. In the second variant, the previous two-branched LSTM model is added with an attention module [99]. Refere to Fig. 3.8 in Appendix 3.7.1 for further details.

As an alternative to recurrent network, we also exploited temporal convolutions in a standalone fashion. As a vanilla convolutional Neural Network (CNN), we employed the architecture inspired by [100]. Fed with raw data, the model performs convolutions across channels followed by convolution in time before feeding a softmax classifier. In addition, we also explored a variant in which raw EEG data are reshaped into a RGB image using theta, alpha and beta frequency bands as red, green and blue color channels, respectively [49]. On top of such representation, we fine-tuned a ResNet-50 model (pre-trained on ImageNet) for the final classification stage. Along the line of feeding an artificial neural network with precomputed EEG features, we also report the performance of a multi-layer perceptron (MLP)

Table 3.2: Performance of hand-crafted features for EEG classification.

	theta	*alpha*	*beta*	*gamma*
FFT	12.06%	13.03%	**12.81%**	12.66%
DE	13.78%	13.93%	22.62%	**28.16%**
Wavelet	10.94%	11.91%	**15.13%**	11.16%

Table 3.3: Performance of learnable features for EEG classification.

vanilla LSTM	16.78%
two-branched LSTM	27.04%
two-branched LSTM + attention	28.01%
vanilla CNN	21.80%
EEG images + ResNet-50	33.56%
DE + MLP	**39.78%**

fed with DE features [97]. The classification performance of this baseline EEG model are reported in Tab. 3.2 and 3.3.

Discussion. When considering hand-crafted features for action recognition, the performance of FFT, DE and Wavelet increases towards higher frequency bands (that is, when ranging from theta to gamma). On alpha, beta and gamma bands, frequency-response descriptors (FFT and DE) improve the time-frequency-response (Wavelet). On beta, DE is still the best descriptor, while Wavelet improves FFT. In all cases, for each frequency band and descriptor, random chance (10%) is significantly improved, and the best classification performance achieved is given by DE features on gamma bands: 28.16%.

An analysis of the performance of EEG images. As proposed in [49], EEG images are an effective strategy to cast EEG input data into colored images (by converting theta, alpha and beta frequency band into the R, G and B color channels of an image - see Appendix 3.7.1 for additional implementation and technical details). Once EEG data is casted into image-like input stream, convolutional neural networks can be adopted, such as ResNet-50. In this paragraph, we are interested in analyzing the performance of this model in terms of confusability of similar action classes among each others.

We provide the Receiving Operator Characteristic (ROC) curves by computing the related area under it (AUC). To do so, we extract the softmax scores from our model trained on EEG images, we compare the scores with which any of the test videos from our dataset is associated by the model to each of the categories, while also having access to the ground truth label. The ROC curves, for each of the 10 classes of our dataset, and the relative AUC values are reported in Fig. 3.2. These indicators are useful in spotting which actions are easier/harder to recognize in absolute terms: *fighting* seems the easier one (AUC = 85.49%), together with *kissing* (83.02%). Actions such as *flying, hugging, running, shooting, surfing* or *throwing* are "intermediate" since their respective AUC is above 70%. Even, for the most

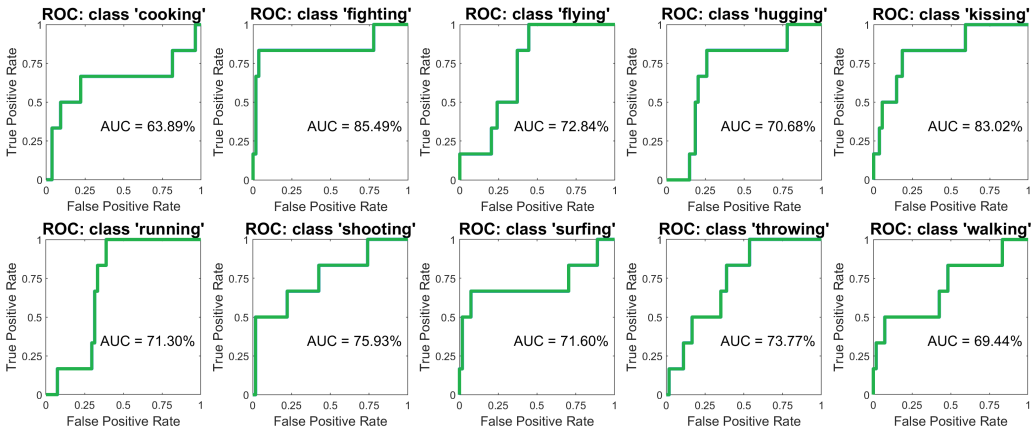


Figure 3.2: Receiving Operator Characteristic (ROC) curve and the relative area under it (AUC), relative to the ResNet-50 model fed with EEG images (see Tab. 3.3).

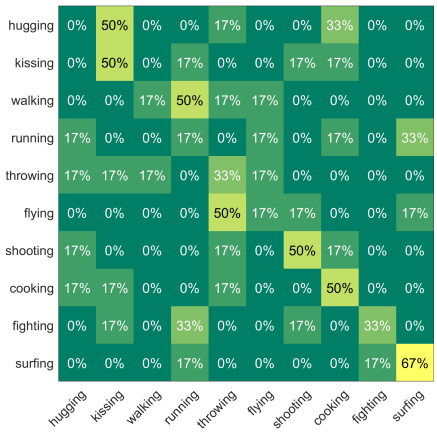


Figure 3.3: Confusion matrix related to the ResNet-50 model trained on EEG images (EEG images + ResNet-50, as in Tab. 3.3). Actual classes are listed by rows, while predictions are displayed by columns.

difficult actions (walking - AUC = 69.44% and cooking - AUC = 63.89%), the classification scores are still reliable enough to certify that the task of recognizing implicit actions from video can be tackled and solved with a sufficient degree of success.

Global statistics of the classification performance can be found in Fig. 3.3 to compare the predictions made by the EEG images + ResNet50 models with the ground truth. As expectable, actions such as *kissing* and *hugging* have a high chance to be confused since both of them imply close physical interaction between two human agents and therefore the visual cues which will help in implicitly referring to these two actions are highly overlapping between each others. Similarly, *walking* and *running* are confused for the very same reason: the most likely visual cue that helps in disambiguating this couple of actions is the execution

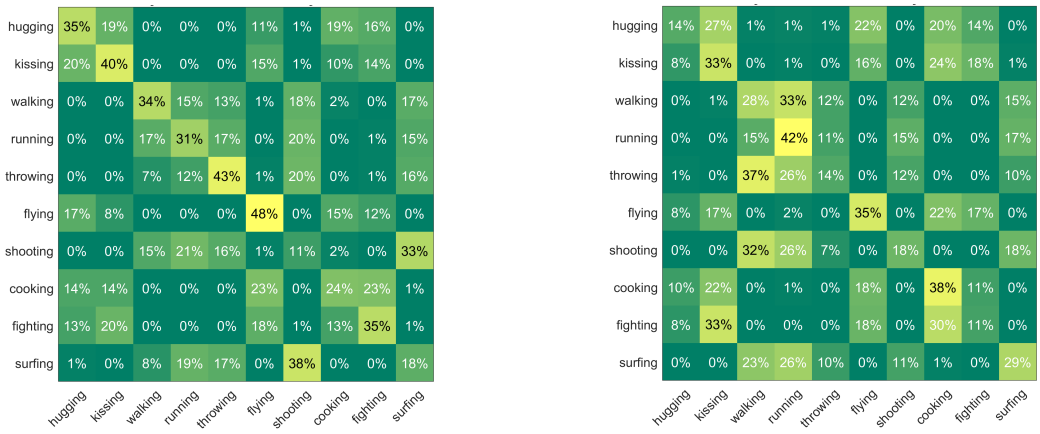


Figure 3.4: Comparison of confusion matrices obtained using two different computational approaches. *Left*: DE+MLP, a multi-layer perceptron (MLP) fed with differential entropy (DE) features (*left*). *Right:* the attention-based LSTM neural network proposed by [Karim et al. 2018]. In both cases (*left* and *right*), we list ground truth (actual) classes by row and predictions by columns.

speed and it may be actually subjective in the way a running/walking action is implicitly referred in a video. *Flying* and *throwing* are sometimes confused between each others and this is understandable from the fact that they both refer to the case in which something is displacing "in the air". In all other cases, the remaining actions are quite well classified, as another evidence for the fact that EEG is a reliable data modality for the sake of recognizing actions, even when they are only implicitly referred in a video and not explicitly visualized.

A comparison between MLP and LSTM. For the sake of comparing different architectural design applied to EEG data, we provide an in-depth analysis on the classifiication performance of two different models: the multi-layer perceptron (MLP) fed with differential entropy features (DE) and the recurrent neural network with Long-Short Term Memory (LSTM) units and attention mechanism. With respect to the aforementioned models, we provide the respective confusion matrices in Fig. 3.4. While comparing MLP and LSTM, we cannot see a clear difference in the way actions are confused by the two alternative models: action pairs such as hugging versus kissing or walking versus running are highly confusable (and this is quite understandable). Obviously, the confusion matrix relative to the MLP is more "diagonal": actions such as throwing or flying are better classified if compared to the LSTM counterpart. Overall this translates into a superior classification score achieved by the MLP: 39.78% versus 28.01%, which is achieved by the (see Tab. 3.3).

Although the LSTM model, by design, is suited to capture the dynamics of an action, the MLP has been endowed by this capability after the usage of differential entropy (DE) features. In fact, DE features are accounting for second order residuals of entropy in time and

are therefore encoding the kinematics in an alternative manner to recurrent networks. Among these two alternative strategies of encoding the temporal extent of an action, the MLP seems more effective and this can be explained in technical reasons: the adopted LSTM model, due to the plug-in of the attention mechanism, has a bigger number of parameters if compared to the simpler MLP classifier which seems to show superior generalization capabilities by better preventing overfitting.

3.4.2 Baseline methods for video footages

We took advantage of dense trajectories [101] to extract local spatio-temporal histogram features (HOG, HOF and MBHx/y). These features were pooled by means of bag of features (1000 codewords), ultimately producing a vectorial representation of the video footage for a subsequent SVM classification.

We also took advantage of deep learning-based action recognition methods using video. Precisely, we sampled 5 frames from each video, feeding them to a CNN designed for image classification (ResNet-50): the final prediction on video is done by averaging the prediction on the sampled frames [16].

Further, we exploited *Temporal Relation Network* (TRN) [1], which is a recent action recognition method devised to simultaneously model several short and long range temporal relations between sparsely sampled frames. Given a video V, composed of n selected ordered frames $f_1, f_2, ..., f_n$, 2-frame temporal relations $T_2(V)$ are defined as

$$T_2(V) = h_\phi^{(2)}(\sum_{i<j} g_\theta^{(2)}(f_i, f_j))$$

and 3-frame temporal relations as

$$T_3(V) = h_\phi^{(2)}(\sum_{i<j<3} g_\theta^{(3)}(f_i, f_j, f_k)).$$

Analogous definition can be expressed for longer-term temporal relationships $T_4(V)$, ..., $T_N(V)$. In the previous formulæ, f_i is extracted features of i^{th} frame and $h_\phi^{(d)}$ and $g_\theta^{(d)}$ are a single-hidden layer neural network. The overall optimization objective $\mathcal{L}$ for the video V is $\mathcal{L}(V) = T_2(V) + T_3(V)... + T_N(V)$ which is optimized via gradient descent with respect to the parameters of the networks $h_\phi^{(d)}$ and $g_\theta^{(d)}$, $d = 1, \ldots, N$. In our experiments using TRN model, we adopted the BN-Inception model [102] pre-trained on ImageNet to extract frame-level feature f_i. Also, the hyper-parameter N in equation $\mathcal{L}$ is selected using prescribed valused in [1], alternatively fixing to $N = 4$ and $N = 8$ to capture medium and long term dependencies in time. Default training strategies of batch normalization and dropout after

Table 3.4: Performance for video classification: $\hbar$ denotes hand-crafted features, while ℓ refers to methods based on feature learning.

$\hbar$	Dense Trajectories: HOG	18.64%
	Dense Trajectories: HOF	16.95%
	Dense Trajectories: MBHx	23.33%
	Dense Trajectories: MBHy	26.67%
ℓ	ResNet-50	29.33%
	TRN ($N = 4$)	28.30%
	TRN ($N = 8$)	31.70%
	TSM	**42.33%**

global pooling are used.

We also adopted *temporal shift models* (TSM) [2] in which standard convolutional neural network baseline architectures (here, ResNet-50) are extended to handle temporal data. This is done by considering frame-wise 2D convolutions along with 1D temporal convolution among temporal shifted version of the input video across time frames. For instance, given an input video of frames I_t indexed over a timestamp t, in addition to 2D convolutions acting on I_t for each t in parallel, temporal shift models also compute a 1D temporal shifted convolutions according to the formula $w_1 I_{t-1} + w_2 I_t + w_3 I_{t+1}$ in the case of a temporal kernel of length 3. Note that the weights of the temporal kernel for shifted convolutions are shared across different shifted version of the input video.

Discussion. Among the hand-crafted histogram features that we considered, MBHx and MBHy resulted in a superior performance with respect to HOG and HOF: this is understandable for the fact that, by design MBHx and MBHy are more robust towards camera motion and are therefore more effective in handling real-world videos as the ones from Moments in Time database. The performance of hand-crafted representation (extracted through dense trajectories) is inferior with the one provided by methods that rely on deep learning. And, among deep-learning based methods, TRN and, especially, TSM provided a better performance with respect to ResNet-50 processing frames, with or without the additional fine-tuning stage.

3.4.3 Baseline for Fusion Methods

Different data modalities usually carry complementary information, which can be exploited with various fusion frameworks. *Early fusion* approaches combine the two modalities at the level of the feature representations and embeddings, separately computed out of each modality. However, given the very different nature of raw EEG and Video data, *late fusion* found to be a more reasonable choice, as it combines higher-level refined information.

By design, our dataset couples *one* video instance with the EEG signals gathered from the *many* subjects who watched the action video. This means that, depending on the task, we can

Table 3.5: Performance of the fusion methods

Kernel Fusion	**46.14%**
Fusion of Logits	45.47%

have two different fusion setups: if the task is the classification of EEG signals, than we can fuse the corresponding video information for each EEG instance. On the other hand, if the task is video classification, we have multiple EEG instances that can be fused with each video instance. We focus here on the former scheme, while dedicating section 3.5 to the latter.

First, inspired by [103], we adopted a kernel fusion approach in which we considered MBHx and MBHy features (encoded with Bag of Features) extracted with dense trajectories, the hidden representation of the MLP fed with DE features and the feature vector produced by the last average pooling layer of ResNet 50 fed with EEG images. For each feature, we computed a linear kernel and the resulting Gram matrices are averaged and fed to a support vector machine for classification.

We also explored a late fusion of logits [104]: we selected the best video model (TRN) and the best model for EEG (MLP fed with DE features). In each model, the input vector to a softmax operator is extracted, averaged together and the final classification performance is computed by arg-maxing over it. The performance of the fusion baseline is reported in Tab. 3.5.

Discussion. When using kernel fusion, hand crafted features perform almost on par to the feature learning scheme in which high level decision functions of neural network are combined (through the averaging of logits of TRN and the MLP fed with DE features). With respect to the best performance achieved when processing video data (which is the fine-tuned TRN model), kernel fusion improves by +11.14% and the averaging of logits by +10.47%. Also, with respect to the best performance scored when processing EEG data (Tab. 3.3, DE + MLP), kernel fusion improves by +6.36% and the averaging of logits by +5.69%.

3.5 Subjects' consensus

We want to go beyond the simple perception of action and we are now considering a higher-level task, attempting to investigate the process of recognizing actions that are not explicitly visualized, but implied only. In this Section, we propose a computational method to solve our targeted goal.

Let us start from the following observation. Since we are tackling the problem of recognizing activities which are implied from video footages, and not explicitly displayed in them, we must assume that our EEGs will codify a certain degree of subjective interpretation. The

latter is related to the fact that, while processing a given footage, each subject will compare it with his/her mental concept for that action. So, it may happen that, for a given subject, his/her mental idea of "`flying`" will be closer to the one of a bird/airplane flying, whereas, for another one, it can be that he/she is closer in reasoning towards a centric-view of the panorama visible from an airplane window. This will translate into EEG recordings which codify for that subjective traits, ultimately hiding the class-related patterns which are directly related to the activities that we are interested to recognize.

In other words, each subject has his/her own biases in understanding activities and, by design, such biases will be captured in EEG signals captured when showing Moments in Time videos to the participants. The fact that, in those video, the action is not explicitly displayed, but simply implied, will cause EEG signals not just to reflect the perception task of visually parsing the video which has been given as stimulus. Differently, in those EEG signals, we are going to capture the mental processes which lead to understanding to which extent the video matches the subjects' personal concept related to a given action. Clearly, the how much a footage is exemplifying a certain activities is expected change across subjects: fireworks ascending the sky may or may not be recognized as truly prototypical for flying. This depends upon the subject who is watching the video and reasoning on it. But, we can observe an important aspect. Despite a video of fireworks may not be strictly related to "flying" for some individuals, still, the same video will be definitely not categorized as an instance of a "cooking" activity from any subject. That is, although subjects may disagree on what is prototypical for a given action, we expect them to agree on what is not.

Leveraging this observation, we want to build a computational method which is capable of taking into account several "opinions" about actions' concepts, in order to achieve a more robust action recognition models. As a specific computational method to take advantage of such observation, we propose *subjects' consensus* in which we consider several action classifiers, each trained on a specific subject. We show that, when averaging the predictions of those classifiers, the combined decision score shows a form of consensus in which erroneous predictions are cancelled out as long as one increases the number of subjects utilized. Remarkably this happens for testing videos, which were never seen from any of the merged classifiers during training. As a consequence, errors' cancellation translates into sharper predictions for the correct class which, ultimately, yields to a better testing classification accuracy.

Implementation details. We implemented subjects' consensus by considering the baseline EEG model consisting in DE features fed into a multi-layer perceptron. In particular, we took advantage of the logits of that model (that is, the vectorial representation which is normalized into a probability density by applying a softmax operator). When a specific

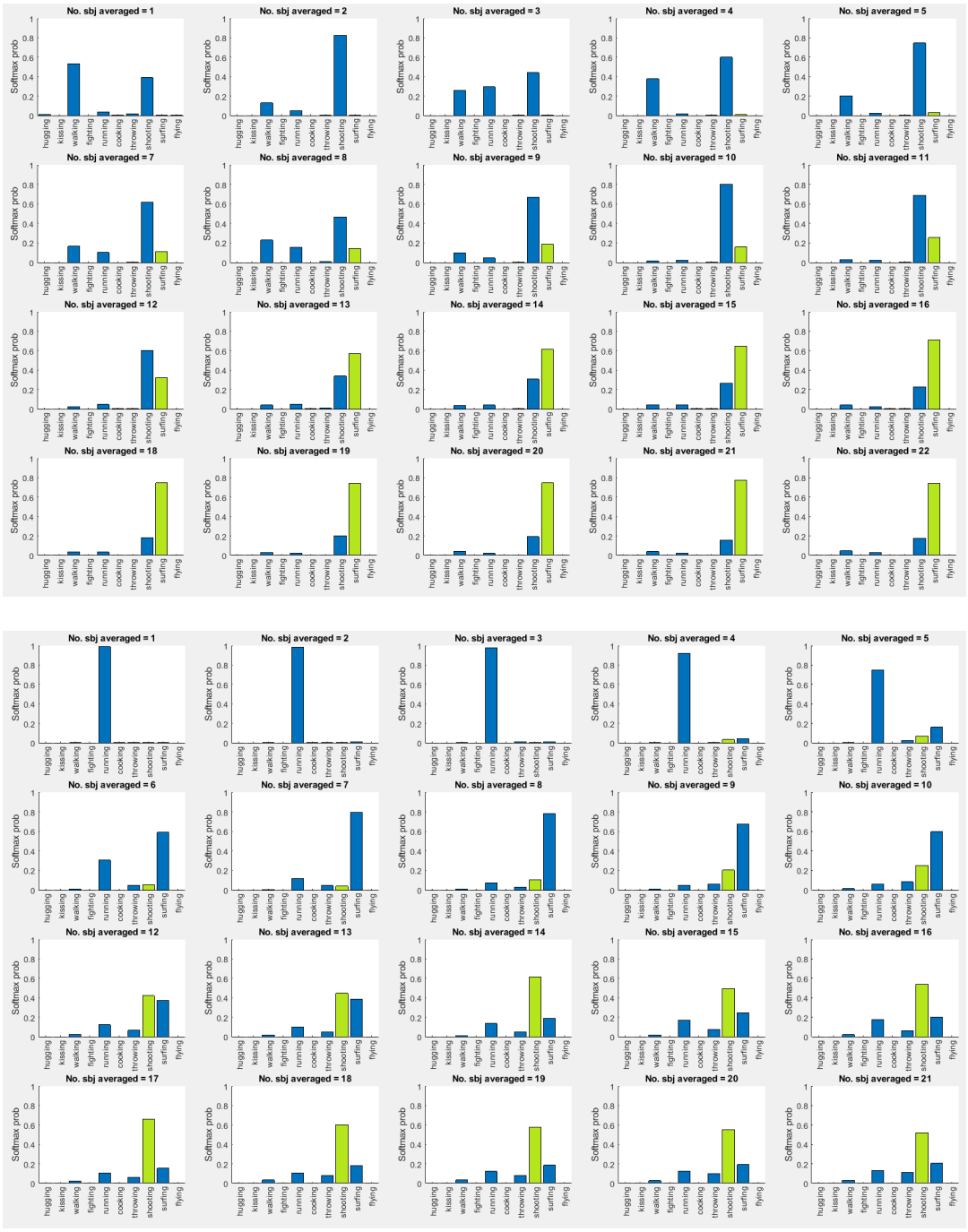


Figure 3.5: Consistency of averaging prediction on models trained on different subjects and tested on the same video footage. The probability of the ground truth class is highlighted in green.

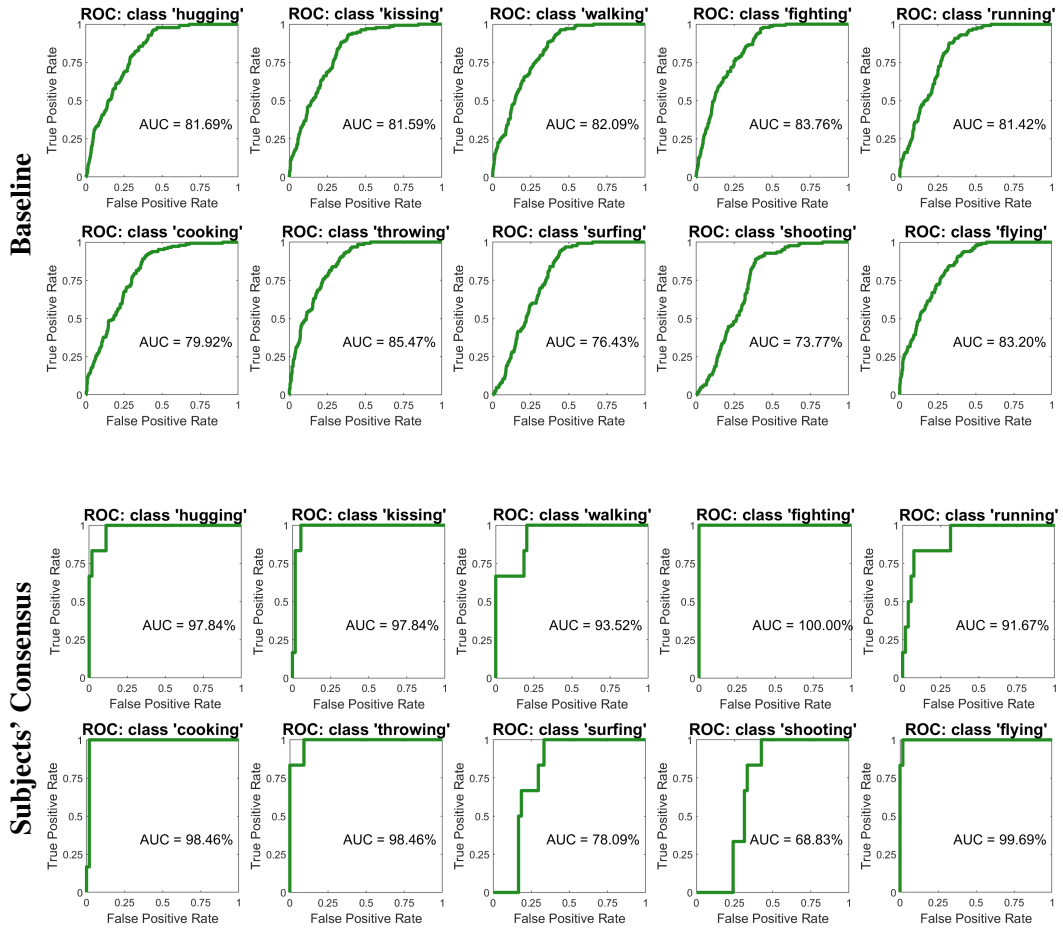


Figure 3.6: Receiver Operating Characteristic (ROC) curve related to the softmax scores of the multi-layer perceptron (MLP) trained with differential entropy features. We directly compare the performance, in EEG classification, of the model without model consensus (*top pane*) with the regularizing effect of subjects' consensus (*bottom pane*) which improves the action recognition for the video modality.

video footage needs to be classified, we considered all the subjects to which that footage was presented as stimulus during the database acquisition. In order to classify such footage we compute the logits of the DE + MLP which processes all the available EEG recordings (belonging to different subjects) corresponding to that footage. Afterwards, we average the logits and apply softmax for visualization purpose. In fact, such operation produces a probability density, indexed over the selected 10 classes, showing the most likely prediction according to the model. The effect of subjects' consensus are shown in Fig. 3.5 and Fig. 3.7.

Discussion. In Fig. 3.5, we consider two fixed testing footages, belonging to the ground truth class of surfing (top) or shooting (bottom). We then visualize the softmax vector which provides the normalize probability of a video belonging to the 10 classes considered in our

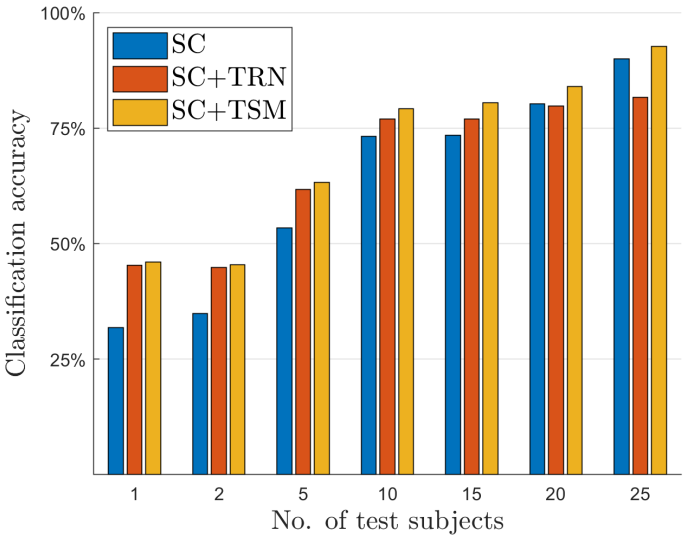


Figure 3.7: Quantitative performance of subjects' consensus (SC). Blue bars correspond to the performance of a DE+MLP (see Sec. 3.4) model trained on EEG data: the final classification is done by averaging the softmax prediction over a different number of subjects. Red bars report the performance of a fused approach in which the averaged prediction over EEG data are furthermore averaged with the prediction of the TRN [1] and TSM [2] architectures trained on video. In this figure, the selected video was not used during training, being therefore never seen before from any of the subject-specific predictions that are averaged. Best viewed in colors.

study (the probability corresponding to the ground truth class is highlighted in green). We ablate on what happens if we fix the testing video and we average the softmax predictions across a varying number of subjects who saw the video during the experimental acquisition. Therefore, when considering one subject only, we are exactly considering the prediction of the DE+MLP model reported in Tab. 3.3, where that model performs inference considering one subject at the time. Differently, when adding several subjects in the averaging of softmax predictions, we are tacking into account several different opinions. Computationally, this translates into sharper softmax predictions, which are more peaked on the ground truth class as long as we increase the number of subject.

The previous trend is quantitatively confirmed by Fig. 3.7 in which we evaluate the video testing accuracy of subjects' consensus, again with a variable number of subjects employed for averaging. With respect to the baseline methods reported in Sec. 3.4, let us stress a key difference: we are evaluating a video action recognition performance. But, to do so, we are using the predictions of models trained over EEG data and then averaged across all subjects who saw the specific test video considered. Therefore, strictly speaking, such performance is not directly comparable with the baseline models in Sec. 3.4. In any case, it is interesting to see that the consensus among subjects is effectively capable of sharply

improving the performance, so that the more subjects we consider, the better the performance. Therefore, the effect of the subjects' consensus keep raising while adding subjects up to 10. Afterwards, we register a sort of plateau in which the performance stabilizes, although, in absolute terms, the best performance is obtained by considering all the subjects. In Fig. 3.7, we also report the performance of the logits produced by subjects' consensus and fused with the predictions obtained from a TRN/TSM model trained with video data. The effect of such multi-modal fusion is represented by the red bars: as one can see, similarly to the case of subjects' consensus alone, accuracy improves when increasing the number of subjects considered for fusing logits. Also, when fixing the number of subjects whose predictions over EEG data are averaged, adding videos to EEG data is helpful to gain in performance. Interestingly, when considering all available subjects that watched a given footage, which is 25 at maximum[1], the subjects consensus leads to a superior classification accuracy when using EEG alone if compared to the combination of EEG and videos. Although this may appear counter-intuitive at a first glance, this seems a consequence of the difficulty of the selected video footages. In fact, since the videos do not explicitly display an action, but simply refer to it implicitly, when the subjects' consensus reaches its optimal performance in cancelling out subjects' biases, it is better to rely on EEG as opposed to videos. In fact, the concept related to the ground truth action to be classified has been unveiled through subjects' consensus in the EEG, while the same remains hidden in the videos.

To better understand the effect of subjects' consensus, we provide an evaluation of its effectiveness when combined to the DE + MLP model (Tab. 3.3) by means of the area under the curve for the Receiver Operating Characteristic (ROC) curve. As visible in Fig. 3.6, with respect to a baseline DE+MLP model trained to perform action recognition from video, the subject consensus is almost always able to raise the AUC since leveraging the consensus that different subjects seem to exhibit when visually stimulated using the very same video footage. In fact for the class hugging, we get a +16.1% absolute improvement in the value of the AUC of the ROC and similar improvements were observed for other classes as well: +16.25% for kissing, +11.43% for walking, +16.34% for fighting, +10.25% for running, +18.54% for cooking, +12.99% for throwing and +16.49%. In only two cases we get either a small improvement (+1.66% for surfing) and in the case of shooting we drop in performance (-4.94% for shooting): despite of this exceptions, the trend is that the subject consensus is able to improve the performance over a baseline EEG recognition pipeline without requiring computational changes to the model but simply aggregating different predictions corresponding to different subjects looking at the same video clip.

In shed of the previous considerations, we are interested in leveraging EEG as a source of

[1] As we explained in Sec. 3.3, we divided the available subjects (50) into two lists, showing to each of them half of the videos.

Table 3.6: Temporal Relation Networks (TRN) [1] with subjects' consensus as privileged information. Testing classification accuracies are reported with mean and standard deviation over 5 different runs.

	$\lambda = 0.25$		$\lambda = 0.50$		$\lambda = 0.75$		$\lambda = 1.00$	
T	Acc	Std	Acc	Std	Acc	Std	Acc	Std
1	30.00%	2.35%	32.67%	2.53%	32.34%	1.49%	34.33%	3.84%
2	31.67%	2.04%	33.33%	1.18%	33.33%	2.04%	31.67%	4.72%
5	33.33%	2.64%	34.33%	1.49%	34.67%	2.17%	35.33%	2.47%
10	33.00%	2.17%	**35.67%**	**1.49%**	34.67%	2.74%	31.67%	3.12%
20	33.33%	2.36%	31.67%	1.18%	32.67%	0.91%	22.00%	1.82%
50	33.33%	2.64%	32.33%	0.91%	34.33%	0.91%	10.33%	0.75%

Table 3.7: Temporal Shift Models (TSM) [2] with subjects' consensus as privileged information. Testing classification accuracies are reported with mean and standard deviation over 5 different runs.

	$\lambda = 0.25$		$\lambda = 0.50$		$\lambda = 0.75$		$\lambda = 1.00$	
T	Acc	Std	Acc	Std	Acc	Std	Acc	Std
1	43.33%	1.66%	43.00%	1.39%	42.33%	1.90%	42.66%	0.91%
2	**43.66%**	**1.39%**	43.00%	1.39%	40.66%	2.23%	42.99%	0.74%
5	40.66%	1.90%	36.33%	1.39%	33.00%	1.39%	23.66%	6.49%
10	38.99%	0.91%	32.33%	0.91%	30.33%	1.39%	22.66%	3.83%
20	38.00%	1.39%	31.33%	1.39%	27.00%	0.74%	19.66%	1.82%
50	37.66%	0.91%	31.00%	0.91%	27.66%	0.91%	15.66%	0.91%

privileged information [12] to boost video classification, as we will show in the next Section.

3.5.1 Subjects' consensus as privileged information

As data modality, EEG needs a specific acquisition setup to be acquired, and, if compared to videos, its portability is clearly inferior. Still, through the potentialities of EEG we are interested in learning an action recognition model (jointly trained on video and EEG) and then being able to deploy such a model in situations where EEG is not available within input data. So, our model will be trained on EEG+video, but tested on video data only.

The previous requirements can be framed in the context of learning with privileged information [12]. Specifically, within the task of predicting y_i given x_i, $i = 1, \dots, n$, privileged information leverages additional information x'_i about the example (x_i, y_i). In our case, x_i will correspond to a video footage, while x'_i will represent an EEG recording of a given subjects watching the same footage as stimulus.

In order to circumvent the usage of EEG data for inference, we exploit the generalized distillation framework [12, 56, 57] by first training a teacher model f_t to perform action classification on EEG data x'_i. Second, we compute the predictions $s_i = \sigma(f_t(x'_i)/T)$ using a temperature parameter T in order to have a smoothing effect and enhance commonalities and differences between classes to be discriminated - this is the true potentiality of soft labels

[12]. Then, we train a student model f_s *using video data only*, by minimizing the loss

$$\frac{1}{n}\sum_{i=1}^{n}[(1-\lambda)l(y_i,\sigma(f(x_i)))+\lambda l(s_i,\sigma(f(x_i))], \tag{3.1}$$

where the imitation factor $\lambda \in [0,1]$ controls the balance between predicted soft labels s_i and ground truth hard annotations y_i.

The results of using subjects' consensus as privileged information are reported in Tab. 3.6 and 3.7. Therein, we provide the test accuracy averaged over 5 different runs of the same model, also showing the corresponding standard deviation. We also ablate on different choices of T and λ. Results reveal that privileged with parameter $\lambda = 0.5$ and $T = 10$ offers an improved accuracy of 35.67% as compared to 31.70% accuracy with TRN alone (Tab. 3.4). Similarly, the baseline performance of TSM (42.33%) is improved by + 1.33% in the case $\lambda = 0.25$ and $T = 2$, reaching 43.66%.

3.6 Discussion, Summary & Future Work

In this study, we investigate the problem of understanding actions' concepts, a new application of EEG data, for which higher level cognitive task is investigated with respect to prior work [83, 84, 87, 86, 51, 88, 89, 90, 91]. In particular, we attempted to go beyond the analysis of mental processes which pertain to decoding what individuals see in a video stimulus. Rather, we aim at decoding what the video stimulus means for the subject. Experimentally, we did this by stimulating a pool of 50 subjects through a selection of videos from the Moments in Time dataset [16]. The peculiarity of the selected videos is that they are designed for action recognition in a setup where an action is not always explicitly visualized, being only implied (as illustrated in Fig. 3.1). We claim that EEG can go beyond the bare appearance of the stimulus, conveying useful discriminative patterns for the classification of some high level concepts (in this case, related to actions), even if the aforementioned concept is not explicitly visualized but only (vaguely) implied. Therefore, we can understand actions' concepts from EEG data which are captured from subjects visually stimulated using the aforementioned videos. In fact, an effective video classification is possible only if we are able to decode which are the mental processes that are elicited in a subject who is reasoning about how the selected footage is prototypical for the ground-truth class.

We presented a broad class of experimental baselines in which we employed state-of-the-art algorithms to assess the performance on action recognition when either using EEG/video footages separately, a fusion of the two, or the EEG as a privileged information to boost video classification. A summary of the best results achieved by means of different setups in avail-

Table 3.8: Highlights of the best achieved performance scored over EEG and video, when used as single data modality, while also including the combination of EEG and video (denoted by "fusion"). We also provide the boost in performance of TRN [1] and TSM [2] state-of-the-art computer vision models achieved from subjects' consensus over a video-only baseline.

EEG only	video only	fusion	subjects' consensus
39.78%	31.70%	46.14%	TRN: 31.70% $\rightarrow$ 35.67% TSM: 42.33% $\rightarrow$ 43.66%

able in Tab. 3.8. As one can notice, the performance on the separated modalities (EEG and video) considered alone is almost balanced. On the one hand, we have an evidence for the fact that concept understanding from EEG is feasible since we can significantly improve random chance performance (which is 10%) using EEG only. In absolute terms, the performance registered from video-based methods is inferior to the one of EEG-based methods: this is an evidence for the fact that EEG is a data modality that can go beyond the visual appearance of visual stimuli, showing that it is capable of supporting more elaborated mental processes related to decoding concepts hidden behind appearance. Furthermore, EEG and video contain complementary information, as proved by the increase in performance obtained through fusion. Finally, by using the EEG in the form of privileged information, we can furthermore boost video classification by around 4%.

As the most interesting consequence of our findings, we discovered that there exists an inter-subject agreement in how an implied action is decoded from a video footage. We proved that such subject consensus is capable of acting as a sort of model ensemble [55, 56, 57] at the level of classification scores derived from EEG data (using the described DE + MLP architecture - see Sec. 3.4). By averaging the classification scores obtained through increasingly adding the number of subjects, we obtain a sharper prediction in correspondence to the correct classes, while mis-classification errors are mitigated (Fig. 3.5). This suggests the intriguing perspective that, while taking into account the action representation of multiple subjects, there is a certain level of agreement on the mental processes related to concept understanding, which goes beyond the subjective evaluation of judging how much a certain footage is truly exemplifying a given action. Computationally, this translates into a privileged information which enhances the performance of computer vision model which rely on video only: we improved the action recognition performance of Temporal Relation Networks (TRN [1]) by +3.97% and the one of Temporal Shift Models (TSM [2]) by +1.33% by means of subjects' consensus.

The proposed dataset and current study open to a variety of future works. First of all, the design of novel and more advance pipelines to exploit the discovered subjects' consensus within a privileged information framework or the more general possibility of creating brain-inspired action recognition methods considering EEG as an ancillary data modality.

In addition to already discussed possibility of decoding actions' concept from brain signals captured by EEG, we can attempt to regress EEG data from video, providing a generative approach to reconstruct the mental imagery of a given participants when processing a stimulus. The availability of EEG data related to videos depicting action concepts would open towards novel research directions, aiming at comparing whether understanding an action's concept is equally easy for both human beings and computational models. Finally, the intrinsic multi-modal nature of the dataset opens to research towards in hallucinating one modality from the other [105], while also exploiting the pairing of EEG and video in self-supervised [106] learning frameworks.

3.7 Appendix

3.7.1 Technical and implementation details about the EEG, video and fusion baseline methods

1. *Methods for EEG classification, hand-crafted features (Tab. 3.2).*

Let us define $\mathbf{x}_t$ as a vector concatenating the registered electrical activity of the brain, at a given timestamp t in correspondence of all the electrodes ($x^{(n)}$) mounted on the scalp. Over a total of 64 channels, VEOG and HEOG are removed as it is usually performed in related studies to get rid of ocular artifacts. Therefore, $\mathbf{x}_t = [x_t^{(1)}, \ldots, x_t^{(n)}, \ldots x_t^{(62)}] \in \mathbb{R}^{62}$ for each $t = 1, \ldots, 750$ since 750 is the number of timestamps (250 timestamps per second are acquired, each video of MiT dataset is 3 seconds in temporal length). To perform data normalization, each acquired sequence is normalized performing a linear scaling of the range of variability of each channel into the range $[-1, 1]$.

We then compute the **Fast Fourier Transform (FFT)** $\{\mathbf{z}_t\}_t$ of the sequence $\{\mathbf{x}_t\}_t$ performing the following computation for each channel:

$$z_t^{(n)} = \sum_{s=1}^{750} x_s^{(n)} \exp\left(-s \cdot t \cdot \frac{2\pi i}{750}\right), \quad n = 1, \ldots, 62. \tag{3.2}$$

After computing FFT features, we extract the required frequency windows (theta 5-7 Hz, alpha 8-13 Hz, beta 14-30 Hz, and gamma 31-60 Hz) using the Nyquist's sampling theorem. We concatenate across different channels and timestamps.

For the **Wavelet transform**, we took advantage of fieldtrip toolbox to compute the mixed and induced power spectrum of Morelet Wavelet function in which we applied an absolute baseline removal strategy in the time window [-900, -300] milliseconds before the video starts. To control the number of cycles of the Wavelet function, we perform and adaptive strategy in which we linearly scale the number of cycles (from 3.5 cycles at 2 Hz to 18 cycles at 60 Hz). We downsample the temporal resolution of the input data by a factor of 3 before computing the wavelet function and we perform zero-padding of the input signal by adding 0.2 secs before and 0.2 secs after it. Again, in correspondence to the selected frequencies of interest, the cut of the computed features is done by using the well-known theory of Nyquist's sampling frequency, and we concatenate the obtained feature representation into a vectorial embedding to represent each instance to be classified.

The entropy of a scalar probability density f supported over the Lebesgue measurable

space $\mathcal{X}$ is given by

$$h(f) = -\int_{\mathcal{X}} f(x)\log(f(x))dx \tag{3.3}$$

and, in the assumption of f being distributed as a Gaussian of mean μ and covariance σ^2 is easily computable through the formula

$$h(f \sim \mathcal{N}(\mu, \sigma^2)) = \frac{1}{2}\log(2\pi e\sigma^2). \tag{3.4}$$

As showing in [Shi et al 2012], there is a linkage between the logarithm energy spectrum and the differential entropy of a random variable, which allows us to estimate the **differential entropy** for each of the component of the multivariate time-series $\mathbf{x}_t$ encoding our EEG data. In particular, for each channel - indexed by n - we compute the scalar value h_n given by

$$h_n = \frac{1}{2}\log(P_n) + \frac{1}{2}\log\left(\frac{2\pi e}{750}\right) \tag{3.5}$$

where

$$P_n = \sum_{t=1}^{750} \left|\tilde{x}_t^{(n)}\right|^2, \tag{3.6}$$

begin $\tilde{x}^{(n)}$ the result of a bandpass filtering of the raw EEG data in correspondence to the frequency window of interest of our theta, alpha, beta or gamma.

For either FFT, DE or Wavelet encodings, we apply a zero-centering and a standardization on each feature component. We then train a linear support vector machine (SVM) using the libSVM library and the recommended default parameters choice.

2. *Methods for EEG classification, learned features (Tab. 3.3).*

We provide a more detailed description on the **Neural Networks** used in Tab. 3.3 to learn features from EEG data: the vanilla CNN, vanilla LSTM and two-branch LSTM (with attention). We provide a visualization of their connectivity graph, together with the size of the learnable parameters and othe specs: check Fig. 3.8. In all cases, the input data is shaped as 62×750 matrix: 62 is the number of channels (once HEOG and VEOG are removed) and 750 are the timestamps, corresponding to the acquisition time. The long-short term memory unit (LSTM) are paired with the number of hidden neurons inside the recurrent network. For the 2D convolutions, we report in brackets a triplet (h, w, n) providing height h and weight w of the kernels adopted, together with their number n. In the case of 1D temporal convolutions, the size of the filters is the very same of the 2D ones, but with the crucial difference that

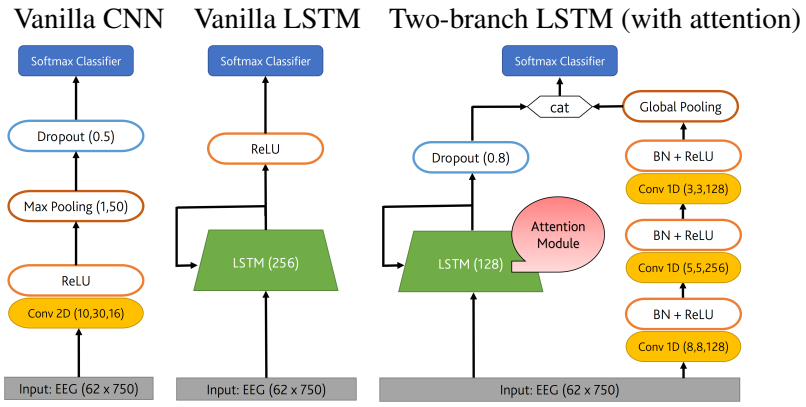


Figure 3.8: Visualization of the architectures used in to learn features from EEG data (check Tab. 3.3).

those filters are applied across timestamps (and not slided over a single frame as in 2D conv). Dropout layers are paired with the value of the retain probability p defined such that $1 - p$ is the dropout rate, that is, the probability of dropping a unit with the usual Bernoulli sampling as recommended in the original dropout implementation. Rectified Linear Units (ReLU) are occasionally adopted as non-linearity and eventually paired with batch normalization (BN). For the max-pooling operator in the Vanilla CNN, we only perform pooling in time (stride = 50), while doing nothing on the EEG channels. For additional details of the global pooling layer or the (optional) attention module of the two-branch LSTM, please refer to the paper by [Karim et al. 2018]. The "cat" module in the two-branch LSTM (with attention) stands for a concatenation operation. All three networks have a softmax classifier which are responsible for the final action recognition stage: we did not explicitly visualize the weights of the classifier which are nevertheless still applied to cast a vectorial embedding (produced by the previous layer) into a vector v whose size is equal to the number of classes (in our case, 10). The same vector v is fed to a softmax operator and casted into a probability vector over the classes to be recognized.

In order to create the **EEG images**, Fast Fourier Transform (FFT) is performed on the time series for each trial to estimate the power spectrum of the signal and the three frequency bands of theta (4-7Hz), alpha (8-13Hz), and beta (13-30Hz) are selected. Sum of squared absolute values within each of the three frequency bands was computed and used as separate measurement for each electrode. The resulting measurements are actedin into a 2D image to preserve the spatial structure, while using multiple color channels to represent the spectral dimension. To do so, first, the location of electrodes is projected from a 3D space onto a 2D surface using the Polar Projection. Width and height of the image represent the spatial distribution of activities over the cortex and the interpolation is applied to cope with

the scattered power measurements over the scalp and for estimating the values in-between the electrodes over a 32×32 planar square mesh (inducing the pixels). This procedure is repeated for each frequency band of interest, resulting in three topographical activity maps corresponding to each frequency band: red for theta, green for alpha and blue for beta. The ResNet-50 architecture fed with EGG images is the model pre-trained on ImageNet. For the MLP architecture, which is instead fed with differential entropy (DE) features, consists of a hidden layer of size 248 with rectified linear units as non-linearities and dropout with rate 0.5.

3. *Methods for video classification (Tab. 3.4).*

For the sake of performing video-classification we either employed dense trajectories (DT) [Wang & Schmid 2013], which are simply the most powerful hand-crafted descriptor proposed before the deep learning breakthrough. Among the feature learning methods, we applied Temporal Relation Network (TRN) [Zhou et al. 2018] and Temporal Shift Models (TSM) [Lin et al. 2019]. As all the relevant details about TRN has been presented in the main text, here, we will discuss only dense trajectories and TSM with extended details.

For the **dense trajectories** (DT), the idea is to take advantage of a spatio-temporal interest point detector which is able to retrieve "corners" in space+time which should represent voxel in which major dynamical variation happens: in the case of DT, those spatio-temporal interest points are found from optical flow and they are tracked for a number L of consecutive frames (we use the default parameter $L = 15$). In correspondence of each of the previous trajectories, a warped volume $\mathcal{V}$ is define so that the trajectory is always at the center of a vertical slice of $\mathcal{V}$ itself (we exploit the default parameter to define the dimension of the slice). From each volume $\mathcal{V}$, several histogram features are computed: histograms of oriented gradients (HOG), histograms of oriented optical flow (HOF) and motion boundary histograms (MBH) which are particularly useful to handle cases in which the camera moves with respect to the scene captured in the video (MBHx and MBHy for either horizontal or vertical displacement). For each of these class of histogram descriptors, many of them are extracted from a single video footage: the aggregation process of them into a fixed vectorial embedding with which the video can be represented is done by means of bag-of-words pooling (using a dictionary of 1000 codewords extracted by means of K-means clustering, $K = 1000$). At the end of this process, a χ^2 kernelized SVM is trained and responsible for the final video classification. To do so, we took advantage of libSVM library, using default parameters.

In **temporal shift models**, standard convolutional neural network baseline architectures (here, we used ResNet 50 as in [Lin et al. 2019]) are extended to handle temporal data by introducing, in addition to frame-wise 2D convolutions, 1D temporal convolution among

temporal shifted version of the input video across time frames. For instance, given an input video of frames I_t indexed over a timestamps t, in addition to 2D convolutions acting on I_t for each t in parallel, temporal shift models also compute a 1D temporal shifted convolutions according to the formula $w_1 I_{t-1} + w_2 I_t + w_3 I_{t+1}$ in the case of a temporal kernel of length 3. Note that the weights of the temporal kernel for shifted convolutions are shared across different shifted version of the input video.

4. *Fusion methods for joint EEG and video classification (Tab. 3.5).*

In the **kernel fusion** approach that we consider in this study, we took advantage of multiple Gram matrices, each of them computed from a single descriptor out of the many we considered: the MBHx and MBHy features (encoded with Bag of Features) extracted with dense trajectories, the hidden representation of the MLP fed with DE features and the feature vector produced by the last average pooling layer of ResNet 50 fed with EEG images. For each feature, we computed a linear kernel and the resulting Gram matrices are averaged and fed to a support vector machine (SVM) for classification. To train this kernelized SVM machine, we took advantage of libSVM library using default parameters.

We also explored a late **fusion of logits**: we selected the best video model (TSM) and the best model for EEG (MLP fed with DE features). In each model, the input vector to a softmax operator is extracted, averaged together and the final classification performance is computed by arg-maxing over it.

3.7.2 Additional details on data acquisition

As we explain in Sec. 3.3, within each batch of videos showed to the participant, 3 of them were "dummy": during dummy videos the fixation cross (which, on "regular" video is located at the center of the screen and white in color) becomes red. This should trigger the participant's response of pressing the spacebar: we monitored, not only that the spacebar was pressed from any of the user after each dummy video, but we also monitored the reaction time for this to happen: the results, available in Tab. 3.9, show that the participants were paying a high attention to the visual stimuli so that the space bar was pressed in correspondence to a dummy video for 94.53%$\pm$8.48% of the time (on average) with an average response time of 1.20$\pm$0.89 seconds that occur after the video ends (and before the spacebar is pressed). We therefore conclude that such a sharply correct execution of this attention task is capable of guaranteeing that each participants paid a high level of attention to the visualized stimuli.

In addition to a careful acquisition stage, we also took advantage of an established preprocessing technique to make sure that the EEG data convey data regarding the visual stimuli: baseline removal. In order to explain why baseline removal can actually help in this respect,

let us point out that one second of time passes after the class name is disclosed to the participant and before the video start. Given our efforts in preserving the participant's attention towards the video screen, we conjecture that, during this second the participant is abstractly thinking about the action's category that will be soon displayed on the video. In other words, in this first second, we are capturing the pure conceptual mental activity of each of the participant when he/she abstractly imagine the class whose semantic label has been disclosed. We use the average EEG activity related to this 1 second segment as the baseline that we adopt afterwards for the pre-processing. Mathematically, we subtract each element of the time series corresponding to the EEG data concurrent to the video by this baseline: this means that we remove the average neural activity related to an exclusive conceptual mentalization of a given action class, so that, the residual EEG activity that results from this operation encapsulates video-related visual activity, cleaned of category-related abstract thinking.

Table 3.9: For each of the 50 participants (referred as S followed by a progressive number in the range $\{1, .., 50\}$), we report two quantitative indicators to monitor their correct accomplishment of our adopted oddball-like task (fixation cross changing color). We report the accuracy with which the space bar is pressed each time a dummy video is shown ("acc"), expressing such value as a percentage. We also provide the maximal response time that was taken by each single subject to press the space bar, while considering all dummy videos he was shown (this value is referred as "$\max_t$" and it is espressed in seconds. For a comprehensive evaluation, we provide also the average and the standard deviation for the acc and $\max_t$ values (in bold).

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11
acc [%]	100	100	100	100	80.00	100	100	86.67	100	86.67	93.33
$\max_t$ [s]	1.13	0.63	1.11	1.90	1.40	0.68	0.63	1.42	0.75	1.25	6.11
	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22
acc [%]	93.33	86.67	100	100	73.33	100	66.67	86.67	100	100	100
$\max_t$ [s]	0.85	0.75	1.13	0.98	1.28	0.78	1.45	1.3	0.99	1.22	1.10
	S23	S24	S25	S26	S27	S28	S29	S30	S31	S32	S33
acc [%]	100	80.00	100	80.00	73.33	100	100	93.33	100	100	100
$\max_t$ [s]	0.63	0.67	3.42	0.80	1.08	0.67	0.68	1.40	0.95	0.65	1.05
	S34	S35	S36	S37	S38	S39	S40	S41	S42	S43	S44
acc [%]	100	93.33	93.33	100	100	93.33	86.67	100	100	93.33	100
$\max_t$ [s]	0.65	2.25	0.65	1.38	0.85	0.77	0.75	0.67	0.77	1.30	0.72
	S45	S46	S47	S48	S49	S50	**AVG**	**std**			
acc [%]	100	93.33	93.33	100	100	100	**94.53**	$\pm$**8.48**			
$\max_t$ [s]	1.00	2.18	1.25	1.57	0.88	1.29	**1.20**	$\pm$**0.89**			