Multi-Dataset, Multitask Learning of Egocentric Vision Tasks

Georgios Kapidis, *Student Member, IEEE*, Ronald Poppe, *Member, IEEE*, and Remco C. Veltkamp, *Member, IEEE*

Abstract—For egocentric vision tasks such as action recognition, there is a relative scarcity of labeled data. This increases the risk of overfitting during training. In this paper, we address this issue by introducing a multitask learning scheme that employs related tasks as well as related datasets in the training process. Related tasks are indicative of the performed action, such as the presence of objects and the position of the hands. By including related tasks as additional outputs to be optimized, action recognition performance typically increases because the network focuses on relevant aspects in the video. Still, the training data is limited to a single dataset because the set of action labels usually differs across datasets. To mitigate this issue, we extend the multitask paradigm to include datasets with different label sets. During training, we effectively mix batches with samples from multiple datasets. Our experiments on egocentric action recognition in the EPIC-Kitchens, EGTEA Gaze+, ADL and Charades-EGO datasets demonstrate the improvements of our approach over single-dataset baselines. On EGTEA we surpass the current state-of-the-art by 2.47 percent. We further illustrate the cross-dataset task correlations that emerge automatically with our novel training scheme.

Index Terms—Egocentric vision, action recognition, multi-dataset training, multitask learning

1 Introduction

Classification models for egocentric vision tasks such as action recognition are predominantly trained using supervised learning schemes. While action recognition from first- and third-person videos can be assumed to have a comparable complexity, labeled datasets for the third-person perspective, (e.g., [1], [2], [3], [4], [5], [6]) are typically orders of magnitude larger than egocentric datasets, (e.g., [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]).

While more general egocentric video datasets exist, (e.g., [7], [8], [15], [21]), they focus on longer-term activities such as walking [15], socializing [10], [11], [16] or doing sports [12]. The recognition of the actions that make up those activities, such as ‘cut a carrot’ and ‘open the fridge’, requires a more granular analysis over shorter video fragments. Egocentric video datasets that address such action recognition tasks are homogeneous in terms of the action domain, recording environment and the recorded actors. While there is a steady progression in the variation within the datasets that have been introduced over the years, each dataset has a focus on a specific task or application.

ADL [9] was one of the first egocentric video datasets that focused on human activities in indoor environments. Participants performed daily activities such as cooking and cleaning in their homes with annotations of the temporal range of activities, objects used and the locations in the house [22]. To increase granularity and specialization in cooking activity recognition, the EGTEA datasets were introduced [14], [23] where participants followed narrated recipes for meal preparation in their kitchens. To scale up the dataset size and remove the use of scripts, the EPIC-Kitchens dataset [19] introduced a culturally diverse set of videos with a large variety of actions and interactions with cooking ingredients and kitchen-related objects. Additional modalities such as object presence are predominantly used both during training and at test time. While the performance of some detection tasks such as object detection is impressive, the requirement of additional inputs for testing is a limiting factor.

Obtaining egocentric videos with relevant labels for various tasks is labor-intensive, and there is a need for learning schemes that can reduce overfitting without requiring more annotated data. In this paper, we introduce such a scheme that uses annotations from both related tasks and related datasets. Using an extended multitask learning (MTL) scheme, we exploit annotations of related tasks during training, while only video data are required for testing. We base our work on ideas developed in [24], where joint training with related video recognition tasks such as object, hand and gaze detection have been shown to improve action recognition performance. We investigate the concept of task relatedness [25]. Our premise is that common actions in different datasets such as ‘cut’ and ‘open’ are associated by the network and the same neural pathways are reused, producing efficient and robust multi-purpose models. This

- *The authors are with the Department of Information and Computing Sciences, Utrecht University, 3584 CC Utrecht, The Netherlands. E-mail: {g.kapidis, r.w.poppe, R.C.Veltkamp}@uu.nl.*

Manuscript received 21 February 2020; revised 12 October 2020; accepted 12 February 2021. Date of publication 23 February 2021; date of current version 5 May 2023.
(Corresponding author: Georgios Kapidis.)
Recommended for acceptance by A. Furnari, D. Crandall, D. Damen, K. Grauman, and G.M. Farinella.
Digital Object Identifier no. 10.1109/TPAMI.2021.3061479

provides an effective and efficient way to utilize additional training data from diverse sources. We allow for video data from other datasets to be used in the training process, and treat the issue of different label sets as additional tasks. Our novel learning scheme is termed Multi-Dataset, Multitask Learning (MD-MTL).

To demonstrate the benefits of our approach, we adapt a 3D-convolutional neural network [26] to include additional task-specific output layers [24] for the tasks of other datasets. In MD-MTL, each epoch consists of the data of the combined training sets, while each batch comprises data randomly chosen out of all datasets, to allow for batch loss calculation that represents the full spectre of available domains. We also experiment with other batch division strategies.

We experiment with combinations of data from EPIC-Kitchens [19], EGTEA Gaze+ [14] and ADL [9] to demonstrate the effectiveness of our multi-dataset, multitask training scheme for egocentric action recognition. Specifically, regarding ADL we investigate the potential improvements on longer term activity recognition performance by utilizing the short-term actions from EPIC and EGTEA. Lastly, we use Charades-EGO [8] to investigate the benefits from associated third-person videos in egocentric action recognition.

The contributions of this paper are the following:

- We extend Multitask Learning (MTL) to include training data from multiple datasets (MD-MTL) with a simple but effective network modification.
- We introduce a batch formation scheme for on-the-fly association of dataset-specific samples to dataset-specific tasks.
- We demonstrate the improvements of MD-MTL in classification performance for the main action recognition tasks. We also highlight the reuse of the same pathways for related classes across datasets.

In Section 2 we review recent advances in video action recognition, multitask learning, and multi-dataset training. In Section 3 we introduce MD-MTL. In Section 4 we describe our experiments and discuss the results in Section 5. In Section 6 we conclude the paper.

2 Related Work

We first provide an overview of video action recognition, with a focus on egocentric action and activity recognition. In Section 2.2 we discuss advances in multitask learning and in Section 2.3 we review related work on multi-dataset training.

2.1 Video Activity Recognition

Since the introduction of large-scale image [27] and video [1], [2] datasets, convolutional neural networks (CNNs) have consistently produced state-of-the-art results [2], [28], [29], [30], [31], [32], [33], [34], [35], [36] for image and video recognition tasks. Likewise, CNN-based approaches have been adopted and adapted to tackle first-person video understanding [37], [38], [39], [40], [41], [42], [43].

Egocentric action recognition has seen incremental improvements over the years with the prominent works of [9], [14], [37], [38], [40], [44], [45], [46], [47]. Originally, feature-based techniques [37], [48], [49], [50] were developed to explicitly model and capitalize on the inherent characteristics of first-person videos such as motions [37], [45], [48], [50], [51], [52], ego-motion [37], [49], [53], human gaze [37], [52], [53], [54], [55] and the presence and movement of hands [9], [37], [44], [49], [55] and objects [9], [37], [44], [49], [55].

The wide use of CNNs in third-person vision was followed by their extensive application in egocentric action and activity recognition [16], [21], [38], [40], [41], [56]. Earlier approaches handled CNN features as an additional modality to handcrafted features [49] or as a feature combination mechanism on previously extracted egocentric features [16]. Fully convolutional approaches viewed action recognition as a learning-based problem with CNNs being used as appearance [24], [57] and motion [58] feature extractors. More data hungry methods used multi-stream deep networks that utilized optical flow alongside RGB images as input modalities [21], [38], [59], [60], [61] to be able to focus on motion. In [61], [62] optical flow was employed to detect salient regions, which were cropped from the original RGB frames and were given to the network as a second, more focused RGB stream. Other input modalities have been employed including depth [7], [41], egocentric cues comprising hand [63], [64], [65] and object regions [64], [66], [67], head motions [63] and gaze-based saliency maps [63], [65], sensor-based modalities [15], [56], [59] and sound [43], [68], [69]. In [38], [40] object and hand localization and segmentation were intermediate learning steps that forced the network to focus on important egocentric cues prior to action prediction.

Explicit attention modeling mechanisms are increasingly common in egocentric video action recognition [14], [54], [55], [65], [66], [70], [71], [72], [73], [74], [75], [76], [77] to influence the network towards focusing on the significant spatio-temporal features of videos. Self-attention approaches do not require additional data but learn the spatial or temporal importance of input video frames with carefully designed attention layers [70], [73], [75], [76] or dynamically weigh the importance of input modalities [71]. In [14], [54], [65], [72], [77], gaze supervision was explicitly required to construct attention maps to weigh the last layer's features before classification. Shen *et al.* [65] used hand segmentation masks in addition to gaze to regulate attention onto informative frames. Finally, in [66] motion- and object-based features extracted from past frames were forwarded to an attention mechanism that effectively combined them with the present and selected the most relevant information to represent the ongoing action. In our work, we model video action and activity recognition with 3D-CNNs to jointly model spatio-temporal features without requiring additional input modalities.

2.2 Multitask Learning

Caruana [25] was one of the first to show the benefits of multitask learning by assigning multiple tasks to be solved jointly by a single model. Recently, this concept has found successful application in image and video understanding tasks [78], [79], [80], [81], [82], [83], [84], [85]. Misra *et al.* [78], investigated the number of task-specific layers that should stem from the backbone network to find the optimal setup to train task dualities, pairing segmentation with surface normal prediction and object detection with attribute classification. In [83], video captioning with action prediction and action performance quality were combined as separate task outputs. In [79], an object detection scheme was proposed where action labels were predicted for each detected object in addition to the object class. In [82], human pose was used prior to action recognition in an intermediate, secondary task. In our work, all task outputs are parallel and do not affect each other, apart from sharing the backbone network.

Another promising direction in multitask learning is adaptive training. In [80], in every training iteration the gradients were scaled per task to find an optimal solution to be backpropagated, whereas in [85], network parameters were randomly selected to be either task-specific or shared across tasks. In [81], an attention mechanism was applied to weigh each layer's activations according to the specified task and in [84] task-specific attention was modulated at the channel level. In this work, we treat all tasks equally to assess the effects of the varied dataset distributions and the complementarity of tasks in the learning process.

In the egocentric activity recognition domain, Yan *et al.* [52] considered the activities performed by each individual participant as separate tasks where the objective was to cluster common activities among participants without supervision. In [38] an object detection and an action classification task were combined after two separate streams were individually trained. In [73] the action classifier was used to bias the classification output of the verb and noun parts of the action label. In [14], [72] the networks were trained to produce gaze maps as intermediate tasks which were applied to the final activations to weigh the action output accordingly. We follow the structure from [24] where a single network was trained on multiple tasks including classification, hand detection, and gaze prediction and extend it to handle tasks originating from different datasets.

2.3 Multi-Dataset Training

Multi-dataset or multi-domain learning is related to transfer learning in the sense that we wish to utilize data from numerous sources in order to optimize the learning process. Usually, this is unfeasible due to the lack of universally compatible annotations that capture all tasks across datasets [86]. Thus, multi-dataset training refers to the combination of diverse data sources concurrently during training to jointly optimize the gradients of a multitask loss from the tasks of all datasets [87], [88]. Kokkinos [88] proposed UberNet to tackle the tasks of boundary, semantic boundary and saliency estimation, surface normal prediction, segmentation and object detection in a single network. The lack of a dataset with annotations for all tasks led to a gradient accumulation update rule that only updated gradients for a task when enough samples had been seen for it. However, it risked memory constraints from maintaining task-specific gradients until the threshold was met. Additionally, the gradient updates for the main block may not be representative of all the tasks in each training step, affecting the statistics in the batch normalization layers [89]. To alleviate this issue, [90] proposed training on interleaved mini-batches per dataset and the use of group normalization [91] to facilitate network convergence. The main difference in our approach is that we create mixed batches that enable the network to grasp information across datasets on every training iteration.

Chong *et al.* [92] jointly modeled human attention with separate output layers for gaze and saliency estimation. Each layer branched-off from a single backbone that was trained with mixed batches. There were as many backpropagation steps per batch as the number of available output layers, which could negatively affect training of the backbone as in [88]. Guo *et al.* [93] proposed several approaches to combine datasets for human pose estimation including the unification of datasets towards a single prediction task, transfer learning between datasets in a sequence, and a multitask scheme to jointly supervise each dataset's output poses. Of the latter, outputs were eventually combined with a voting mechanism. This approach used fully compatible datasets, from a task perspective, making task fusion feasible. We also investigate mapping related tasks across datasets.

A more related approach to ours [94] considered concatenating output layers for cross-dataset classification, but without leveraging the possible class similarities throughout tasks. Alternatively, [95] performed inter-dataset experiments on EPIC-Kitchens and EGTEA Gaze+, but only on the subset of common classes. Our approach is different in that we construct a single model that fully encapsulates both datasets. Lastly, [96] considered explicit task outputs for face attribute classification, with mixed batches across datasets and masked losses, while attempting to diversify the learned manifold by adding a domain adaptation output to discriminate the datasets during training. In contrast, we focus on unifying the learned representations.

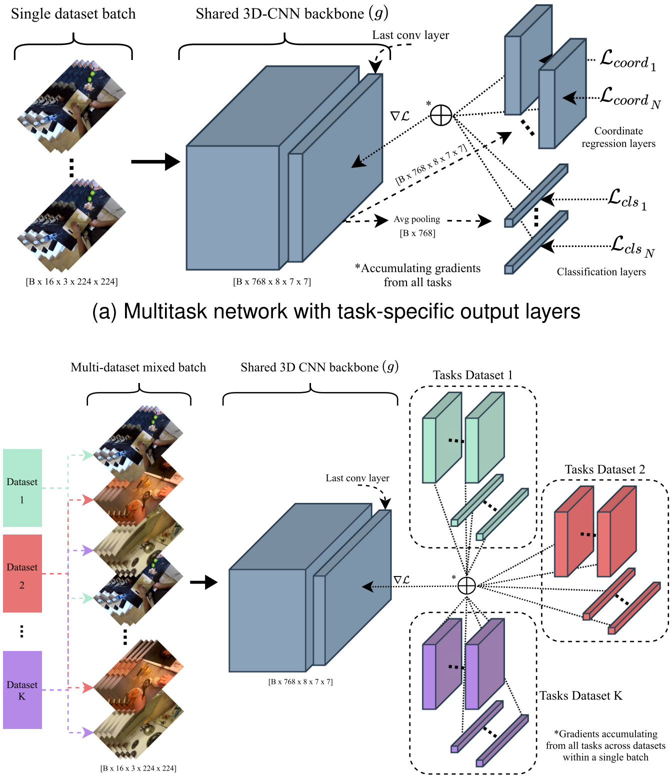
3 Methodology

In this section we describe the extension of a single task network to multitask (MTL) (Section 3.1), and subsequently describe our process to adapt it to multiple datasets (MD-MTL) (Section 3.2).

3.1 Multitask Network Structure

We adopt the multitask network with task-specific output layers (MTL) from [24]. It comprises a 3D-CNN backbone feature extractor [26] that receives a short video clip and outputs spatio-temporal features after the last convolutional layer. We prefer 3D-CNNs because they can handle motion information from the temporal structure of the video without requiring an additional optical flow input. Recent approaches to capture motion from RGB, e.g., [35], [69], are promising developments to further acquire temporal motion features but these are out of scope for this work. Fig. 1a shows the MTL network with task-specific layers.

In our MTL setting we define a set of tasks $\mathcal{T}$ with a distinct task-specific output layer for its respective results. Formally, for each task $t \in \mathcal{T}$ we define an output function $f_t(g(x);\theta_t)$, with $g(x;\theta_s)$ the shared block, θ_t the task-specific parameters, θ_s the shared parameters from g and x the network input. Each task-specific layer comprises a distinct loss function designed to accommodate the type of task it represents. We use classification and coordinate regression tasks. Classification tasks consist of a fully connected layer. Their inputs are the activations of $g(x)$, followed by an average pooling operation to reduce the temporal dimension, and their outputs are the per-class probabilities. To train classification tasks we use the categorical cross-entropy loss.



(a) Multitask network with task-specific output layers

(b) Multitask network adapted for multiple datasets

Fig. 1. We adapt the MTL network structure from [24] to accommodate the tasks from a range of datasets within a single network. In (a) the network combines task-specific output layers by aggregating the gradients from each output. In (b) we extend the structure by further attaching task-specific layers for the additional tasks in the new datasets.

We use coordinate regression tasks in our experiments (see Section 4) to find egocentric hand locations and estimate gaze. These are implemented with the numerical coordinate regression layer, introduced in [97], to predict a coordinate for every two input frames and extended in [24] to handle 3D feature volumes as input. The coordinate regression layer begins with a 3D convolution. The 3D output is split along the temporal dimension with each slice Z being passed to a Differential Spatial to Numerical Transform (DSNT) layer [97] to produce a coordinate for each. In the DSNT layer, each slice is passed through a softmax activation to produce a 2D probability distribution $\hat{Z}$ that represents the abstract location; the final (x, y) coordinate is taken as the probability distribution's expectation for each dimension. To train the coordinate regression layer we utilize the DSNT loss which is defined as the euclidean distance between the predicted (c_p) and the ground truth (c_{gt}) coordinate regularized with the Jensen-Shannon divergence to smooth the gradients around the prediction with a factor $\lambda = 0.5$. Analytically, the DSNT loss function is given in Equation (1):

$$\mathcal{L}_{coord} = \lambda \mathcal{L}_{euc}(c_p, c_{gt}) + (1 - \lambda) JS(\hat{Z} \parallel \mathcal{N}(c_{gt}, \sigma^2)). \quad (1)$$

3.2 Multi-Dataset Network Adaptation

Our extension from single- to multi-dataset training (MD-MTL) requires two modifications. The first is to append task-specific layers for the tasks of the additional datasets and the second is to adapt the training process to accommodate for the induced variation in the mixed training batches.

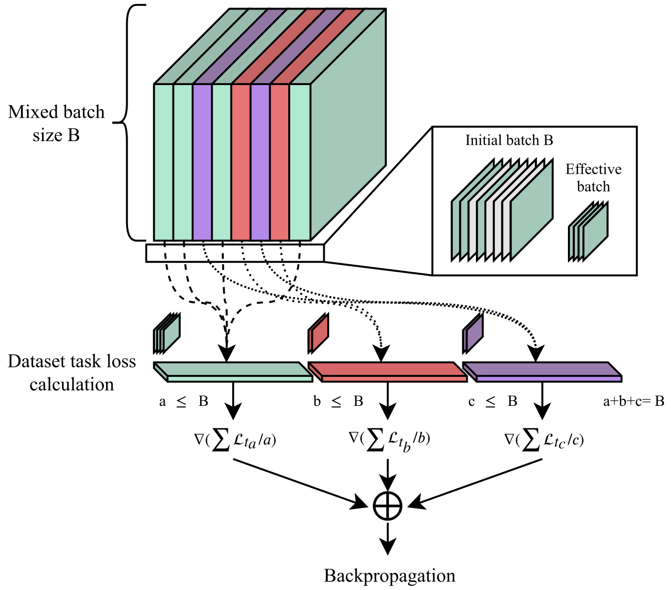

Mixed batch size B
Initial batch B
Effective batch
Dataset task loss calculation
a ≤ B
b ≤ B
c ≤ B
a+b+c= B
$\nabla(\sum \mathcal{L}_{t_a}/a)$
$\nabla(\sum \mathcal{L}_{t_b}/b)$
$\nabla(\sum \mathcal{L}_{t_c}/c)$
Backpropagation

Fig. 2. Mixed-batch loss approximation. A batch is subsampled for each task. The loss from each task layer is averaged over its dataset's samples. Task-specific losses are passed through their respective layers.

We handle the additional tasks in the way we would treat any added task from the initial dataset, i.e., we add task-specific layers to the shared network block that produce a distinct output, independent of the other tasks. Similar to the single-dataset MTL network, each output layer in MD-MTL utilizes its own loss function for training. A visualization of this extension is given in Fig. 1b.

We need to accommodate for the fact that no samples within a mixed batch have labels associated with all the available tasks, since each subset corresponds to a distinct dataset. Our strategy is to leverage the process of averaging the loss across a batch, which is commonly employed when training neural networks with mini-batches.

The premise is that for a batch of size B the loss is calculated B times and averaged to provide an approximation of a B-sized mini-batch. Loss averaging is not possible when batches assimilate different datasets and tasks. In this case, we subsample each batch based on its origin dataset i and produce an effective batch per dataset of size b_i. Then, we calculate each task's loss for the appropriate samples only, zeroing out those that were forwarded through a task-specific layer for which there is no available label. The losses are then averaged over the size of the effective batch b_i and gradients for each task-specific layer are calculated with respect to the dataset tasks' losses. Once the per-task gradient approximation is handled, they are accumulated before being backpropagated through g. Consequently, all tasks are contributing into training the shared network block regardless of the number of samples that were taken from each dataset. We visualize this process in Fig. 2.

Multi-dataset training with mixed batches (instead of interleaved batches or alternating datasets sequentially) allows the network to gather gradients from samples representing the full range of available datasets in a single training step. Hence, the direction of the gradient will not be representative of one dataset as in single dataset training. Instead, it will be biased by all datasets in a ratio defined by the sampling process during batch formulation. We permute all datasets and allow the imbalance to be induced in the network. Due to the similarities of datasets in our experiments, we expect mixed batches to contain complementary information and prevent divergence in training. Indeed, we see in Section 4.1 evidence of improved performance when the datasets are related and deterioration when they represent a different domain. In the supplemental material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2021.3061479, we visualize and discuss the progression of the training losses.

4 Experiments

First we discuss the datasets we experimented with and the training and evaluation settings. In Section 4.1 we analyze the experiments with egocentric datasets and in Section 4.3 we delve into a task mapping scheme to capitalize on the semantic class relationships. In Sections 4.2 and 4.4 we analyze the mechanics of MD-MTL models to demonstrate the correlations across tasks from different datasets and in Section 4.5 we focus on the extension for datasets between first and third-person vision. In Section 4.6 we experiment with additional batch formation strategies. Finally, in Section 4.7 we provide a comparison with the state-of-the-art on EPIC-Kitchens and EGTEA Gaze+.

Datasets. We design multi-dataset experiments on egocentric video datasets EPIC-Kitchens [19], EGTEA Gaze+ [14] and ADL [9], all of which capture activities performed in homes from the first-person perspective. EGTEA Gaze+ consists of scripted meal preparation activities, whereas EPIC promotes action variability by encouraging participants to behave consistently to their routines. Videos from both datasets take place in kitchens, ensuring homogeneous locations, and consist of specialized and related sets of short duration actions such as ‘open’, ‘close’ and ‘cut’. ADL is less specific in terms of environments and actions, capturing a predefined set of daily living activities occurring throughout the participants’ homes, performed in an unscripted manner. These annotations represent temporally longer activities such as ‘washing dishes’ or ‘watching tv’, which makes it harder to represent the whole activity in the short video segments that are used as input to the network. Hence, content-wise, EPIC and EGTEA are suitable candidates for our task- and dataset-relatedness experiments in order to estimate the possible benefits of joint training. On the other hand, the more varied context of ADL allows us to investigate whether our multi-dataset training approach can adapt to a more diverse domain within a single model.

Furthermore, we perform experiments on the Charades-EGO [8] dataset. It comprises a joint collection of first and third-person videos. For each third-person video there is an associated egocentric one, recorded by the same participant for the same activities and environment. This allows researchers to model the association between the two video perspectives. Our aim is not to capture the inter-video associations but to examine if a model trained on contrasting perspectives can be efficiently applied to both, simultaneously. Table 1 lists the datasets and their characteristics.

TABLE 1
List of Datasets and Their Characteristics

Name	ADL	EGTEA	EPIC	CH-EGO
Videos	fpv	fpv	fpv	fpv/3rd
Participants	20	32	32	112
Scripted	partially	yes	no	yes
Labels				
Actions	18	106	2513	157
Verbs	-	19	125	33
Nouns	-	53	352	38
Locations	8 [22]	kitchen	kitchen	16
Other	objects	gaze, recipes, hand segm.	objects, narrations	narrations

We emphasize on the sizes of the classification tasks.

Following [24] we also leverage hand location predictions. They have been found to improve classification performance when included as additional tasks in a multitask setting, due to the implicit focus on the salient regions. For the annotations, we synthesize the left and right hand location coordinates for each frame of ADL, EGTEA, and EPIC-Kitchens using the hand detection algorithm presented in [64]. As shown in [24], these synthetic hand annotations lead to accurate hand detection models.

Training and Evaluation Settings. For all experiments we use a Multi-Fiber Network (MFNet) [26] pretrained on Kinetics-400 [1] as the backbone feature extractor. It acts as the initial structure upon which task-specific layers are attached. Our choice is justified by the fact that it comprises a 3D CNN structure, able to capture spatio-temporal information without the need for an optical flow stream, with a significantly lower number of parameters (~8M), for a depth similar to a 3D ResNet-50 (~47M). We train all models with triangular cyclical learning rate [98] oscillating from 0.0005 to 0.005 and back within 20 epochs. Our training cycle is repeated three times, (i.e., 60 epochs) unless otherwise stated. We use stochastic gradient descent for optimization, with Nesterov momentum (0.9) and weight decay (0.0005). The input for training is a sequence of 16 frames uniformly sampled from a 32-frame window randomly chosen to represent the action segment for an epoch. The selected frames are scaled to 256×256 and randomly cropped to 224×224. Additionally, we perform color augmentations and flip the sequence horizontally with a 50 percent chance. Even though it is counter-intuitive to train a hand detector that identifies left and right hands with random video flipping, early experiments showed that it does not affect hand estimation. Lastly, we use batch size of 32 for both single and multi-dataset experiments, for comparison purposes.

To evaluate an action segment, we select 16 frames from a 32-frame window around the clip’s temporal center. We resize to 256×256 and use the 224×224 center crop. The indicated performances are derived from the best performing weights for the action task, acquired with early stopping.

4.1 Multi-Dataset Experiments on EPIC, EGTEA, and ADL

Single Dataset Baselines. In the single dataset (SD) MTL setup in [24] the trainable tasks for EPIC are action, verb, and noun classification and left/right hand location prediction

TABLE 2
SD-MTL Top1/Top5 Accuracy (%) for Actions (A), Verbs (V) and Nouns (N) for EPIC and EGTEA (Reported From [24]) and for Activities (A) and Locations (L) for ADL for the Best Performing Weights on (A)

Model	Top1 (A-V-N/A-L)			Top5 (A-V-N/A-L)		
E_{ALL} (EPIC)	19.29	48.9	27.27	35.39	78.18	47.85
G_{ALL} (EGTEA)	68.99	79.08	79.03	91.74	99.26	96.39
A_{ALL} (ADL)	64.65	72.22		88.38	96.97	

TABLE 3
EPIC, EGTEA, and ADL MD-MTL Task Combinations

Tasks	Top1	Top5	Top1 S1	Top1 S2
E_{ALL}	19.29	35.91	**29.73**	**17.86**
E_A+G_A	18.15	35.93	24.35	17.04
E_{ALL}+G_{ALL}	**19.69**	**36.68**	26.69	17.17
E_{ALL}+G_{ALL}+A_{ALL}	18.29	34.15	24.17	15.84

(a) EPIC-Kitchens: Top1/Top5 (%) action classification accuracy on the validation set and Top1 on the S1 and S2 test sets

Tasks	Top1	Top5	Mean cls acc.
G_{ALL}	68.99	91.74	61.40
E_A+G_A	69.78	**93.37**	62.31
E_{ALL}+G_{ALL}	**70.38**	93.08	**62.61**
E_{ALL}+G_{ALL}+A_{ALL}	69.34	92.63	60.87

(b) EGTEA Gaze+: Top1/Top5 (%) and mean class accuracy (%) for the action classification task on test split 1

Tasks	Top1	Top5	Mean cls acc.
A_{ALL}	**64.65**	**88.38**	**56.10**
E_{ALL}+G_{ALL}+A_{ALL}	58.08	86.87	43.61

(c) ADL: Top1/Top5 (%) and mean class accuracy (%) for the activity classification task on the validation set

An overview for all classification tasks appears in the supplemental material, available online.

(E_{ALL}). For EGTEA, gaze estimation is added to the set of trainable tasks (G_{ALL}). ADL annotations describe long-term activities with the addition of indoor locations from [22] (A_{ALL}). For EPIC, training and validation are performed on the custom train/val splits from [24], namely 26,375 action segments from participants 1-29 are used for training and the remaining 2,095 for validation, with the exception of videos withheld by the dataset authors for testing. The latter denote scenarios on seen (S1) and unseen (S2) kitchens. S1 consists of videos from participants that also have a number of videos in the training set, whereas in S2 all participant videos are excluded from the training set. S1 and S2 are evaluated on the EPIC-Kitchens server. On EGTEA we use the first split provided by the dataset authors which consists of 8,299 training and 2,022 validation segments and for ADL we train on the videos of participants 1-6 (111 clips) and validate for participants 7-20 (198 clips). We report the SD baselines in Table 2.

4.1.1 EPIC-Kitchens Analysis

We now turn to multi-dataset learning (MD-MTL). We incrementally add new datasets and their tasks to be trained alongside EPIC. The multi-dataset (MD) experiments are named after the included tasks, so E_{ALL}+G_{ALL} contains all tasks of the SD EPIC experiment (E_{ALL}) and all tasks from the SD EGTEA experiment (G_{ALL}). We also perform an MD experiment only on the action tasks for the two datasets (E_A+G_A) to show the effect of the missing classification and coordinate regression tasks in the MD-MTL setting. In Table 3a we compare models containing EPIC-Kitchens in the training set.

E_A+G_A For this experiment we trained only on the 2,513 and 125 action classes of EPIC and EGTEA, respectively. We achieve a similar level of overfit on the validation set but results on both test sets are below the SD baselines, especially for S1. This highlights the importance of the additional tasks to regularize training and enhance the information acquired by the network when they are present, verifying [24] about the merits of MTL, also in an MD setting.

E_{ALL}+G_{ALL} We proceed to integrate actions, verbs, nouns and hands from EPIC and actions, verbs, nouns, hands and gaze from EGTEA. The additional tasks offer a noticeable improvement on action classification for EPIC over the SD baseline on the validation set. This shows that the network is able to fit both training sets simultaneously and that there is potential benefit from our approach if applied on a larger scale. However, we also observe a decline in test set S1 performance. We highlight that performance on S2 is not as affected as in S1. The reason is that the additional tasks from EGTEA prohibit the network from overfitting on EPIC, resulting in a larger performance drop on the seen kitchens. The model's generalization capability to unseen data is less affected, manifesting relatively robust results on S2.

E_{ALL}+G_{ALL}+A_{ALL} The addition of the ADL action and location tasks reaches the limit of the learning capability of our model. The domain shift that occurs from the long unstructured activity videos prohibits convergence to the same minimum for EPIC. Thus, test performance also drops.

4.1.2 EGTEA Gaze+ Analysis

We now evaluate the EGTEA tasks of the previous models. Table 3b summarizes the results on the action task.

E_A+G_A In this experiment we train only on the EGTEA and EPIC action tasks. Performance improves from the SD baseline (+0.79 percent Top1, +0.91 percent mean class accuracy). This already shows the benefit from using MD-MTL. We are improving on EGTEA without adding data specifically for it, but only train jointly with a related task from a different dataset.

E_{ALL}+G_{ALL} Similar to EPIC, using all available classification tasks together with the coordinate regression layers further improves performance. It is +1.39 percent in Top1 and +1.21 percent in mean class accuracy up from the SD baseline and +0.60 and +0.29 percent, respectively, from E_A+G_A. This is another case of the benefits from using MTL to utilize not only the additional relevant data, but all the learnable tasks.

E_{ALL}+G_{ALL}+A_{ALL} Adding data and tasks from the ADL dataset worsens action classification performance on the EGTEA tasks. Since EGTEA has a larger window for

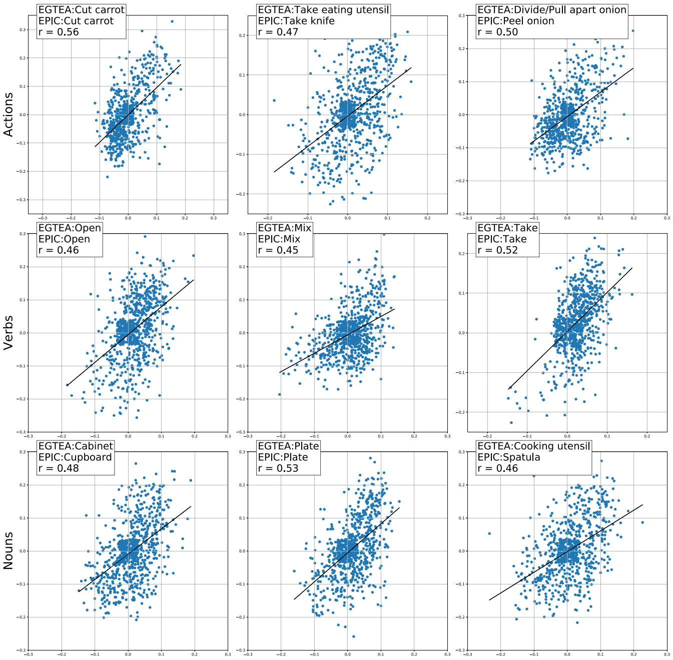


Fig. 3. Correlations for classification weights across tasks in multi-dataset model E_{ALL}+G_{ALL} (Zoom in for best view).

improvement, the decline due to ADL is not as strong as for the EPIC tasks and the SD baseline is still surpassed. However, the effects of the domain shift are evident. The training loss for the action task is higher (bottom graph in Fig. 1 of the supplemental material, available online) illustrating the difficulty to assimilate actions from the highly variable locations of ADL with the kitchen environments of EGTEA.

4.1.3 ADL Analysis

To train on ADL E_{ALL}+G_{ALL}+A_{ALL} we add one more learning cycle to the model and train for 80 epochs, to accommodate for the diverse distribution of the ADL dataset. Results on ADL are presented in Table 3c.

E_{ALL}+G_{ALL}+A_{ALL} Following the results on EPIC and EGTEA, the three-dataset model is unable to reach the single dataset baseline of ADL. This result verifies the previous conclusion that additional datasets without a related data distribution can hurt performance.

4.2 Weight Correlations

In this section we analyze the learned classification weights in MD networks. We measure the correlations between weights for the task pairs of actions, verbs and nouns. We find that positive correlations arise in the classification weights across tasks for classes with similar semantic interpretations. This is an important finding that demonstrates the ability of the network to capitalize on the relationships of the data without additional supervision. We highlight some examples in Fig. 3. We show correlations for classes with the same name, e.g., ′take′ in both EGTEA and EPIC ($r = 0.52$), but also on classes with similar semantic meaning, e.g., ′tomato′ in EGTEA correlates with ′heart′ ($r = 0.43$) in EPIC which refers to a tomato′s interior, with second best the correlation with the actual ′tomato′ class ($r = 0.38$). Correlation values are higher across action tasks, possibly due to their stricter nature in having to associate both the correct verb and the correct noun class. For example, the verb and noun constituents for ′divide/pull apart onion′ correlate with ′peel′ and ′onion′ in EPIC with $r = 0.26$ and 0.37 respectively, whereas the correlation with action ′peel onion′ is $r = 0.50$. This means that the model is more certain about the combination of features it requires when classifying a full action class instead of having to assess it as the union of a verb and a noun. In the following section we investigate a way to further exploit the associative ability of the network by mapping these classes into the same task.

4.3 EPIC and EGTEA With Task Mapping

In many cases, the datasets have partly overlapping label sets for some tasks. In this experiment we reduce the output layers of the network by mapping similar tasks across datasets. We combine the verb and noun classification tasks of EPIC and EGTEA and the hand coordinate layers. We leave the action layers and the gaze unchanged. Our aim is to connect the verb and noun tasks as much as possible while training the action tasks independently. This effort resembles the merged labels technique in [94]. Our approach differs in that we manually map the semantically similar verb and noun classes of EGTEA to EPIC since the majority of its labels are identical or synonyms. There are rare cases where an EPIC label needs to be assigned to multiple EGTEA labels. For example, verb classes ′wash′ and ‘clean/wipe’ are both assigned to EPIC′s ‘wash’ and noun classes that represent containers such as ′tomato container′ and ′bread container′ are assigned to ‘package’. This task combination scheme is less naive compared to our earlier MD approach. The downsides are that we are not able to properly evaluate the verb and noun tasks of EGTEA due to the many-to-one class assignments and that an almost direct mapping across tasks is not always feasible. The task mapping model is trained for 80 epochs (referred to as Verb-Noun Mapping).

Verb-Noun Mapping results for EPIC are presented in Table 4a. Action recognition performance is similar to the naive MD approach but with a significant increase in verb and noun classification as well as in Top1 on the EPIC test sets. In fact, with task mapping, the model is able to generalize as well as with the SD model on the S2 test set. This improvement shows that MD-MTL has an even greater potential when secondary tasks of the datasets can be combined explicitly.

Task mapping also proves beneficial for the action recognition task of EGTEA as shown in Table 4b. Verb-Noun Mapping is +0.99 percent from the previous best (E_{ALL}+G_{ALL}: 70.38 percent) and +2.38 percent from the SD baseline (G_{ALL}: 68.99 percent). Next, we present an additional experiment on SD EGTEA with its initial weights pretrained on the SD EPIC model E_{ALL}. It improves +1.09 percent from the SD model pretrained on Kinetics, but is still lower than both naive MD (-0.30 percent) and MD with task mapping (-1.29 percent). This shows that MD-MTL networks can capitalize on the additional data advantageously over transfer learning, while keeping the tasks of the initial dataset functional and potentially improved.

TABLE 4
Mapping EGTEA Verb-Noun Tasks on EPIC

Model	A	V	N	S1 A	S2 A
E_{ALL}+G_{ALL}	**19.69**	45.99	25.65	26.69	17.17
Verb-Noun Mapping	19.68	**48.33**	**28.32**	**28.1**	**17.86**

(a) EPIC-Kitchens: Top1 (%) action (A), verb (V), noun (N) accuracy on the validation set and Top1 for actions on S1-S2 test sets

Model	Top1	Top5	Mean cls acc.
G_{ALL}	68.99	91.74	61.40
G_{ALL} pretrained on EPIC	70.08	92.63	**62.66**
E_{ALL}+G_{ALL}	70.38	**93.08**	62.61
Verb-Noun Mapping	**71.37**	92.78	62.23

(b) EGTEA Gaze+: Top1/Top5 and mean class accuracy (%) for actions on test split 1

TABLE 5
Comparison Between SD and MD-MTL on the Mapped Verbs and Nouns

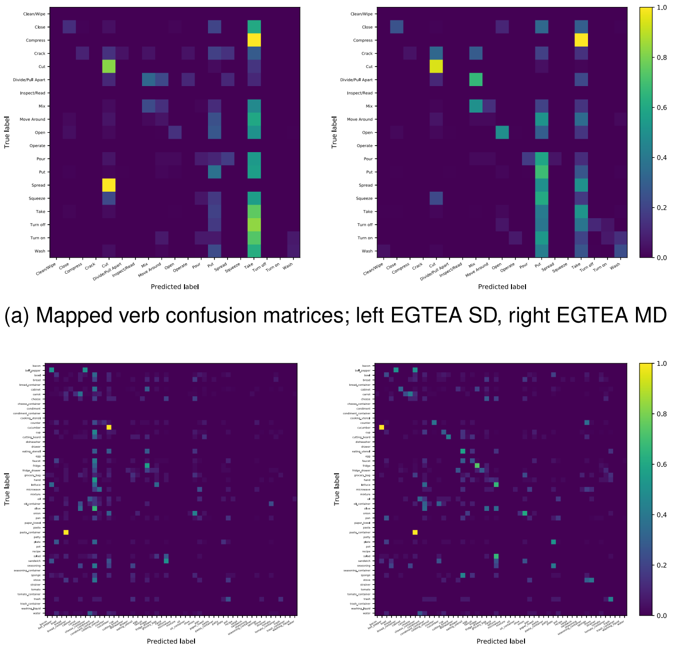
Model	Top1 (%)	Mean cls acc. (%)
Mapped Verbs		
G_{ALL}	32.68	13.98
E_{ALL}+G_{ALL}	**45.14**	**21.14**
Mapped Nouns		
G_{ALL}	9.40	6.86
E_{ALL}+G_{ALL}	**25.75**	**12.84**

Evaluating the EGTEA tasks on the EPIC validation split.

4.4 Task Affinities

The task mapping approach from Section 4.3 enhances the correlations across actions, while fixing some inaccurate cases of the previous model. For example, correlation for the 'cut carrot' action increases from $r = 0.56$ to $r = 0.66$ and for 'peel onion' from $r = 0.50$ to $r = 0.54$. Notably, for the latter, the second best correlated action to 'divide/pull apart onion' from the first model is 'peel potato' with $r = 0.44$ which drops to $r = 0.37$. This suggests that the model is now better able to tell apart the two objects.

To further demonstrate the correlated outputs we compare the performance of the EGTEA SD model (G_{ALL}) against the EGTEA verb and noun tasks of the E_{ALL}+G_{ALL} MD model on the EPIC validation split for the samples that comprise mapped classes. This corresponds to 1,677 samples for verbs and 1,107 for nouns. Table 5 shows Top1 and mean class accuracy for the mapped verbs and nouns. The improvement of the MD model is consistent over SD, achieving +12.46 and +7.16 percent on the two metrics for verbs and +16.35 and +5.98 percent for nouns. This increase demonstrates the generalization ability of MD-MTL for samples that do not belong in the data distribution for which the tasks are trained for. Finally, in Fig. 4 we visualize the normalized confusion matrices for these experiments. In Fig. 4a we observe fewer errors for verbs such as 'turn on' and 'turn off' and the performance of highly represented classes such as 'cut', 'open' and 'close' increases. Similarly for nouns, in Fig. 4b, we see that the SD model (left) tends to classify a number of samples as 'condiment container' which is largely fixed in the MD case (right). Generally, most noun classes have significant improvements.



(a) Mapped verb confusion matrices; left EGTEA SD, right EGTEA MD

(b) Mapped noun confusion matrices; left EGTEA SD, right EGTEA MD

Fig. 4. Confusion matrices for mapped verbs (a) and mapped nouns (b) from the EGTEA tasks on the EPIC validation split.

4.5 Multi-Dataset Experiments on Charades-EGO

We perform experiments on Charades-EGO to explore the associative ability of tasks when applied on data from different perspectives and the potential for performance improvements in the MD-MTL setting. We split the dataset into its first- and third-person constituents and treat them as two separate datasets. Consequently, we have two sub-datasets, charego1 and charego3, with the same classification tasks. We produce action segments from the video level annotations. This results in 33,099/9,148 action segments for charego1 and 34,269/9,386 for charego3 for training and validation, respectively. In Table 6 we report video level mean Average Precision (mAP) following [8] and Top1/Top5 accuracy for the action task. The performance of the remaining classification tasks can be found in the supplemental material, available online. We train three models in total. An SD model for charego1 for actions, verbs and nouns ($C1_{ALL}$), an SD model for charego3 for the same tasks ($C3_{ALL}$) and the MD combination with both sets of tasks ($C1_{ALL}$+$C3_{ALL}$).

Validation on charego1 shows that MD training provides a marginal improvement over the SD baseline on the video level mAP. This shows the benefit to the first-person tasks when using the third-person videos to train their distinct tasks in the MD setting. An interesting insight arises from evaluating on the first-person data using the respective $C3_{ALL}$ tasks of the MD model. Recognition performance is worse when compared to the egocentric tasks, however it is significantly higher from the charego3 SD network. This shows that the network learns to associate the internal

TABLE 6
Action Recognition Performance on Charego1 and Charego3, the First- and Third-Person Splits of Charades-EGO, Respectively

	Validation on charego1			Validation on charego3		
Model	Top1	Top5	mAP	Top1	Top5	mAP
$C1_{ALL}$ (SD)	**7.05**	24.21	21.90	3.55	14.70	12.30
$C3_{ALL}$ (SD)	3.61	15.40	14.70	**8.15**	**27.02**	**20.40**
$C1_{ALL}$ (MD)	7.01	**24.69**	**22.10**	6.79	22.85	18.20
$C3_{ALL}$ (MD)	5.81	21.75	20.10	8.12	26.04	20.00

SD models are trained on all tasks (actions, verbs, nouns) of their split. The MD model is trained on the combination of the tasks of both splits. Results in %.

TABLE 7
Comparison Across Batch Formation Strategies Mixed (MB), Interleaved Batches (IB), and Interleaved Datasets (ID) on the Task Combinations of EPIC and EGTEA

	Top1 (%)			Top5 (%)		
Strat.	Actions	Verbs	Nouns	Actions	Verbs	Nouns
MB	19.69	45.99	25.65	36.68	78.37	50.67
IB	**20.11**	47.76	**29.99**	**37.78**	**78.84**	**51.24**
ID	17.91	**48.42**	23.26	33.57	78.18	45.94

(a) Results on EPIC-Kitchens

	Top1 (%)			Mean cls acc. (%)		
Strat.	Actions	Verbs	Nouns	Actions	Verbs	Nouns
MB	**70.38**	80.57	**79.03**	**62.61**	80.02	**73.55**
IB	65.43	79.72	74.83	55.31	77.21	65.95
ID	69.68	**80.86**	78.64	61.31	**81.59**	72.13

(b) Results on EGTEA Gaze+

representations of classes that co-exist in different tasks and reuses them across perspectives (also confirming the findings of Sections 4.2 and 4.4 in this setting).

Similar insights can be inferred from the results of the third-person video split of Charades-EGO. In this experiment, the SD model exhibits marginally better mAP than the MD model, but the correlation property across tasks of different perspectives is still present. The first-person tasks of $C1_{ALL}$+$C3_{ALL}$ have +5.9 percent higher mAP from the SD $C1_{ALL}$ model when evaluated on charego3.

4.6 Batch Formation Strategies

The mixed batch (MB) formation strategy described in Section 3.2 is not the only way to load data in the MD-MTL network. To further demonstrate the ability of our batch formation strategy to allow optimal generalization across datasets, we compare against two alternative strategies: interleaved batches (IB) and interleaved datasets (ID). In interleaved batches, in every iteration a dataset is selected at random and the input to the network consists of data only from this dataset. In interleaved datasets, batch composition is the same, but each dataset's training set is fully processed before data from the remaining datasets are seen. In either case, the network sees the complete training set of every dataset per epoch. We experiment on the E_{ALL}+G_{ALL} tasks for every batch strategy, using the same training hyperparameters defined in Section 4. In Table 7 we summarize our results.

The three strategies have different effects on the performance. Interleaved batches outperform mixed batches on EPIC, albeit with a strong performance drop for EGTEA. A possible reason is that the size difference of the datasets (the training set of EPIC is almost three times larger than that of EGTEA) does not allow the network to equally capture fine-grained features from EGTEA. When using the interleaved datasets strategy, we see a significant performance drop for EPIC, with EGTEA being more robust. This is the result of the order with which datasets are seen on every epoch. In our ID experiment, the training set of EPIC is always seen first and EGTEA follows in every epoch. Information that is acquired in the beginning of an epoch is partly "unlearned" when the second dataset is seen. Mixed batches (MB) appear to perform somewhat more consistently. However, the modest differences between the strategies suggest that MD-MTL performs favorably over single-dataset MTL, independent of the choice of batch formation strategy.

TABLE 8
State-of-the-Art Comparison on EPIC-Kitchens

			Test S1 (Seen kitchens)						Test S2 (Unseen kitchens)					
			Top1 (%)			Top5 (%)			Top1 (%)			Top5 (%)		
Method	Modalities	Params	A	V	N	A	V	N	A	V	N	A	V	N
TSN [19]	RGB+F	20.2M	20.54	48.23	36.71	39.79	84.09	62.32	10.89	39.40	22.70	25.26	74.29	45.72
EF [43]	RGB	11M	19.86	45.68	36.80	41.89	85.56	64.19	10.11	34.89	21.82	25.33	74.56	45.34
R(2+1)D-34 [86]	RGB	64M	26.80	59.10	38.00	46.10	87.40	62.70	16.80	48.40	26.60	31.20	77.20	50.40
LSTA [73]	RGB+F	82M	30.33	59.55	38.35	49.97	85.77	61.49	16.63	47.32	22.16	30.39	77.02	43.15
VN Mapping	RGB	10M	28.10	55.62	38.04	49.38	86.39	62.69	17.86	46.57	25.74	36.26	77.60	51.86
MTL [24]	RGB	10M	29.73	56.00	40.15	50.95	87.06	64.07	17.86	45.99	26.25	35.68	77.98	50.19
VFS [68]	RGB+F+AU	218M	29.13	44.64	30.64	49.71	76.41	59.39	18.40	38.37	15.23	35.64	75.15	39.84
RU [71]	RGB+F+O	52.6M	33.06	56.93	43.05	55.32	85.68	67.12	19.49	43.67	26.77	37.15	73.30	48.28
GSM [36]	RGB	13M	33.45	59.41	41.83	-	-	-	20.18	48.28	26.15	-	-	-
EF [43]	RGB+F+A	32.6M	**36.66**	**66.10**	47.89	**58.62**	**91.28**	**72.80**	20.97	54.46	30.39	39.40	81.23	55.69
LFB [66]	RGB+TO	201.2M	32.70	60.00	45.00	55.30	88.40	71.80	21.20	50.90	31.50	39.40	77.60	57.80
SAP [99]	RGB+O	198.6M	34.80	63.20	**48.30**	55.90	86.10	71.50	23.90	53.20	33.00	40.50	78.20	58.00
AV-SF [69]	RGB+SF+AU	38.5M	35.90	65.70	46.40	57.80	89.50	71.70	24.00	55.80	32.70	**43.20**	**81.70**	**58.90**
R(2+1)D [86]	RGB+ED	118M	34.50	65.20	45.10	53.80	87.40	67.80	**25.60**	**57.30**	**35.70**	42.70	81.10	58.70

A = Actions, V = Verbs, N = Nouns, F = optical flow, AU = audio, O = objects/object features, TO = object features at various temporal locations, ED = Pretraining on very large scale external datasets, VN Mapping = Verb-Noun Mapping.

TABLE 9
Action Recognition Accuracy on EGTEA Gaze+

		Split 1		Avg. Splits 1-3	
Method	Modalities	Top1	Mean cls	Top1	Mean cls
Li *et al.* [14]	RGB+F	-	47.71	-	-
MCN [72]	RGB+F	55.63	-	-	-
RU [71]	RGB+F+O	-	-	60.20	-
ego-rnn [70]	RGB+F	62.17	-	60.76	-
LSTA [73]	RGB+F	-	-	61.86	-
SAP [99]	RGB+O	64.10	-	62.70	-
STAM [77]	RGB+F	68.60	60.54	65.97	57.02
MTL	RGB	68.99	61.40	65.70	57.60
MD-MTL	RGB	70.38	**62.61**	**68.44**	**59.90**
VN Mapping	RGB	**71.37**	62.23	-	-

Refer to Table 8 for the used abbreviations and number of parameters.

4.7 State-of-the-Art Comparison

EPIC-Kitchens. In Table 8 we compare against the state-of-the-art on the S1 and S2 test sets of EPIC-Kitchens. Our method shows competitive performance, however a number of methods have improved accuracy. One reason is the additional input data that most of these methods employ. For example, the top performing approach [86] utilizes a much larger network (118M parameters) and is pretrained on a video dataset about 3k times larger than Kinetics-400 (IG-Kinetics-65M). Interestingly, with Kinetics-400 pretraining on a network eight times larger than ours (R(2+1) D-34, 64M parameters) they perform -1.30 percent lower on S1 and -1.06 percent on S2 Top1 actions. Furthermore, a number of methods include optical flow, object and audio input streams which tend to leverage separate networks for each modality. We highlight [43] which outperforms us with their full model but when only the RGB stream is utilized we show a +7.75 percent improvement. The remaining approaches do not offer an ablation with only the RGB stream, therefore we cannot compare directly. We note [36] that only use RGB input and are +2.32 percent better on S2. Their model uses feature gating to encode temporal information forward and backward in time with a 2D network backbone. Applying this in our 3D network is an interesting direction for future work.

EGTEA Gaze+. Only a number of the aforementioned works provide action recognition results on EGTEA Gaze+. We compare against methods that utilize RGB and optical flow in Table 9. Despite the enhanced input, we are able to outperform all of them with significant margins. The previous state-of-the-art on split 1 of EGTEA Gaze+ [24] achieves 68.99 percent Top1 and 61.40 percent mean class accuracy, which we surpass by 1.39 and 1.21 percent with MD-MTL and by 2.38 and 0.83 percent with MD-MTL with task mapping, respectively. Furthermore, using MD-MTL the performance on the average of the three splits of EGTEA Gaze+ improves from [77] by 2.47 percent on Top1 and 2.88 percent on mean class accuracy despite the absence of optical flow in our method.

5 Discussion

In this work we introduced an effective batch scheme that comprises samples from multiple datasets and associates them with their respective tasks during training. This approach manifests a trade-off between acquiring the optimal estimation of the gradient direction from a batch from a single data distribution and the need to accommodate the presence of samples from multiple datasets in every training iteration. Essentially, we expect the network to find a minimum along a variety of manifolds which can be costly for optimization, and even not possible if the dataset distributions are incompatible. We found that EPIC and EGTEA show improvements in their validation sets which indicates that the scheme of multi-dataset training is potentially beneficial when semantically related datasets are combined. At the same time it is practical in terms of producing outputs that reflect tasks from multiple domains without sacrificing accuracy. However, the inclusion of ADL showcases the possible pitfalls of adding a dissimilar dataset. We also observed performance improvements when applying the multi-dataset training scheme on a combination of first- and third-person videos on Charades-EGO. This shows that a difference in video perspectives does not prohibit the network from learning a shared representation when other aspects of the datasets such as the environment and the performed actions are related.

In Section 4.3 we trained an SD model on EGTEA where we used weights pretrained on EPIC for initialization. Even though EPIC-Kitchens is not as large as the video datasets that are usually employed for pretraining video recognition models, (e.g., [1], [2]) we expected that the similarity between the source and the target domain would prove beneficial, and it did. We also showed that our multi-dataset approach outperforms pretraining, while retaining all tasks.

We showed in Sections 4.2 and 4.4 that MD training drives classification layers to reuse feature sets for similar classes across tasks. This is an important element of these models, as it occurs without additional supervision, i.e., we do not specify which labels across datasets are related. Our experiments show that this is a typical phenomenon in MD-MTL. It also reinforces the basic concept of multitask learning that related tasks, even from varied sources, support each other by affecting the shared parameters.

However, class correlations are not so strong to suggest full reuse of features for the same classes. This leads to two distinct observations. First, the capacity of a network when trained for a single dataset is not fully utilized. We showed that the underlying weights can be adapted to accommodate additional information. Hence, whatever minimum is reached with SD training does not necessarily correspond to an optimal exploitation of the millions of parameters of modern neural network architectures. Instead, our experiments show that their capacity is larger than SD fitting initially suggests. Second, adaptive training mechanisms that substitute hard parameter sharing, such as explicit task-attention mechanisms [81], [84] or implicit weight assignment to tasks [85] are simulating larger network capacity not by inducing better associations among the shared weights, which MD-MTL seems to be achieving, but by establishing mechanisms to mask noisy features that otherwise find their way to the task-specific prediction layers. We believe a soft parameter sharing mechanism is a promising way forward for MD-MTL as the two concepts are complementary.

6 Conclusion

In this work we introduced a deep learning training scheme that allows a single network to assimilate tasks from diverse datasets and tasks simultaneously. By combining samples across datasets within every batch, we effectively approximate having individual batches per dataset on every training iteration. We applied our scheme in the context of egocentric action classification, on EPIC-Kitchens, EGTEA Gaze+ and ADL datasets and the first- and third-person splits of Charades-EGO. Our results show that multi-dataset multitask (MD-MTL) training offers consistent improvements to classification tasks across datasets when the underlying data distributions are related. Furthermore, we demonstrated that networks acquire similar representations for semantically similar classification tasks without being instructed to do so. Results on EPIC-Kitchens show that our method is able to compete with the state-of-the-art. On EGTEA Gaze+ we outperform more complex networks, surpassing the state-of-the-art by 2.47 percent. We highlight that MD-MTL is an efficient technique to combine data from multiple sources without sacrificing the distinctive characteristics of one dataset in order to classify on another.

Acknowledgments

The authors would like to thank Noldus Information Technology for providing equipment for the experiments. This work was also partly carried out on the Dutch national e-infrastructure with the support of the SURF Cooperative. This work was supported by the Netherlands Organization for Scientific Research (NWO) with a TOP C2 Grant ARBITER.

References

[1] W. Kay *et al.*, “The kinetics human action video dataset,” 2017, *arXiv: 1705.06950*.

[2] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-scale video classification with convolutional neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1725–1732.

[3] K. Soomro, A. R. Zamir, and M. Shah, “UCF101: A dataset of 101 human actions classes from videos in the wild,” 2012, *arXiv:1212.0402*.

[4] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, “HMDB: A large video database for human motion recognition,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2556–2563.

[5] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, “Hollywood in homes: Crowdsourcing data collection for activity understanding,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 510–526.

[6] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, “ActivityNet: A large-scale video benchmark for human activity understanding,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 961–970.

[7] Y. Tang, Y. Tian, J. Lu, J. Feng, and J. Zhou, “Action recognition in RGB-D egocentric videos,” in *Proc. IEEE Int. Conf. Image Process.*, 2017, pp. 3410–3414.

[8] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari, “Charades-Ego: A large-scale dataset of paired third and first person videos,” 2018, *arXiv: 1804.09626*.

[9] H. Pirsiavash and D. Ramanan, “Detecting activities of daily living in first-person camera views,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2847–2854.

[10] Y.-C. Su and K. Grauman, “Detecting engagement in egocentric video,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 454–471.

[11] Y. J. Lee, J. Ghosh, and K. Grauman, “Discovering important people and objects for egocentric video summarization,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 1346–1353.

[12] G. Bertasius, A. Chan, and J. Shi, “Egocentric basketball motion planning from a single first-person image,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5889–5898.

[13] A. Iwashita, A. Takamine, R. Kurazume, and M. S. Ryoo, “First-person animal activity recognition from egocentric videos,” in *Proc. 22nd Int. Conf. Pattern Recognit.*, 2014, pp. 4310–4315.

[14] Y. Li, M. Liu, and J. M. Rehg, “In the eye of beholder: Joint learning of gaze and actions in first person video,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 639–655.

[15] K. Nakamura, S. Yeung, A. Alahi, and L. Fei-Fei, “Jointly learning energy expenditures and activities using egocentric multimodal signals,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6817–6826.

[16] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, “Lending a hand: Detecting hands and recognizing activities in complex egocentric interactions,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1949–1957.

[17] C. Li and K. M. Kitani, “Pixel-level hand detection in ego-centric videos,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3570–3577.

[18] Y. J. Lee and K. Grauman, “Predicting important objects for egocentric video summarization,” *Int. J. Comput. Vis.*, vol. 114, no. 1, pp. 38–55, Aug. 2015.

[19] D. Damen *et al.*, “Scaling egocentric vision: The EPIC-KITCHENS dataset,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 753–771.

[20] X. Ren and C. Gu, “Figure-ground segmentation improves handled object recognition in egocentric video,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3137–3144.

[21] Y. Poleg, A. Ephrat, S. Peleg, and C. Arora, “Compact CNN for indexing egocentric videos,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2016, pp. 1–9.

[22] G. Vaca-Castano, S. Das, J. P. Sousa, N. D. Lobo, and M. Shah, “Improved scene identification and object detection on egocentric vision of daily activities,” *Comput. Vis. Image Understanding*, vol. 156, pp. 92–103, 2017.

[23] A. Fathi, Y. Li, and J. M. Rehg, “Learning to recognize daily actions using gaze,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 314–327.

[24] G. Kapidis, R. Poppe, E. van Dam, L. Noldus, and R. Veltkamp, “Multitask learning to improve egocentric action recognition,” in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2019, pp. 4396–4405.

[25] R. Caruana, “Multitask learning,” *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[26] Y. Chen, Y. Kalantidis, J. Li, S. Yan, and J. Feng, “Multi-fiber networks for video recognition,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 364–380.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[28] J. Carreira and A. Zisserman, “Quo vadis, action recognition? A new model and the kinetics dataset,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4724–4733.

[29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[30] L. Wang *et al.*, “Temporal segment networks: Towards good practices for deep action recognition,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 20–36.

[31] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. M. Snoek, “VideoLSTM convolves, attends and flows for action recognition,” *Comput. Vis. Image Understanding*, vol. 166, pp. 41–50, 2018.

[32] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 4489–4497.

[33] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, “A closer look at spatiotemporal convolutions for action recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6450–6459.

[34] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.

[35] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6201–6210.

[36] S. Sudhakaran, S. Escalera, and O. Lanz, "Gate-shift networks for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1099–1108, doi: 10.1109/CVPR42600.2020.00118.

[37] Y. Li, Z. Ye, and J. M. Rehg, "Delving into egocentric actions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 287–295.

[38] M. Ma, H. Fan, and K. M. Kitani, "Going deeper into first-person activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1894–1903.

[39] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari, "Actor and observer: Joint modeling of first and third-person videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7396–7404.

[40] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori, "Object level visual reasoning in videos," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 106–122.

[41] G. Bertasius, H. S. Park, S. Yu, and J. Shi, "First person action-object detection with EgoNet," in *Proc. Robot., Sci. Syst.*, 2017, doi: 10.15607/RSS.2017.XIII.012.

[42] B. Tekin, F. Bogo, and M. Pollefeys, "H+O: Unified egocentric recognition of 3D hand-object poses and interactions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 4506–4515.

[43] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen, "EPIC-fusion: Audio-visual temporal binding for egocentric action recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 5491–5500.

[44] A. Fathi, A. Farhadi, and J. M. Rehg, "Understanding egocentric activities," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 407–414.

[45] M. S. Ryoo and L. Matthies, "First-person activity recognition: What are they doing to me?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2730–2737.

[46] D. Damen, T. Leelasawassuk, and W. Mayol-Cuevas, "You-do, I-Learn: Egocentric unsupervised discovery of objects and their modes of interaction towards video-based guidance," *Comput. Vis. Image Understanding*, vol. 149, pp. 98–112, Aug. 2016.

[47] A. Furnari, S. Battiato, K. Grauman, and G. M. Farinella, "Next-active-object prediction from egocentric videos," *J. Vis. Commun. Image Representation*, vol. 49, pp. 401–411, Nov. 2017.

[48] K. Zhan, S. Faux, and F. Ramos, "Multi-scale conditional random fields for first-person activity recognition," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2014, pp. 51–59.

[49] R. Yonetani, K. M. Kitani, and Y. Sato, "Recognizing micro-actions and reactions from paired egocentric videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2629–2638.

[50] S. Song, N.-M. Cheung, V. Chandrasekhar, B. Mandal, and J. Liri, "Egocentric activity recognition with multimodal fisher vector," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2016, pp. 2717–2721.

[51] S. Song, V. Chandrasekhar, N.-M. Cheung, S. Narayan, L. Li, and J.-H. Lim, "Activity recognition in egocentric life-logging videos," in *Proc. Asian Conf. Comput. Vis. Workshops*, 2015, pp. 445–458.

[52] Y. Yan, E. Ricci, G. Liu, and N. Sebe, "Egocentric daily activity recognition via multitask clustering," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 2984–2995, Oct. 2015.

[53] K. Ogaki, K. M. Kitani, Y. Sugano, and Y. Sato, "Coupling eye-motion and ego-motion features for first-person activity recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2012, pp. 1–7.

[54] Z. Zuo, L. Yang, Y. Peng, F. Chao, and Y. Qu, "Gaze-informed egocentric action recognition for memory aid systems," *IEEE Access*, vol. 6, pp. 12 894–12 904, 2018.

[55] K. Matsuo, K. Yamada, S. Ueno, and S. Naito, "An attention-based activity recognition for egocentric video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2014, pp. 565–570.

[56] R. Possas, S. P. Caceres, and F. Ramos, "Egocentric activity recognition on a budget," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5967–5976.

[57] H. F. M. Zaki, F. Shafait, and A. Mian, "Modeling sub-event dynamics in first-person action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1619–1628.

[58] G. Abebe and A. Cavallaro, "A long short-term memory convolutional neural network for first-person vision activity recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 1339–1346.

[59] S. Song *et al.*, "Multimodal multi-stream deep learning for egocentric activity recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2016, pp. 378–385.

[60] A. Furnari, S. Battiato, and G. M. Farinella, "Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2019, pp. 389–405.

[61] H. Jiang, Y. Song, J. He, and X. Shu, "Cross fusion for egocentric interactive action recognition," in *Proc. Int. Conf. Multimedia Model.*, 2020, pp. 714–726.

[62] L. Fa, Y. Song, and X. Shu, "Global and local C3D ensemble system for first person interactive action recognition," in *Proc. Int. Conf. Multimedia Model.*, 2018, pp. 153–164.

[63] S. Singh, C. Arora, and C. V. Jawahar, "First person action recognition using deep learned descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2620–2628.

[64] G. Kapidis, R. Poppe, E. van Dam, L. Noldus, and R. Veltkamp, "Egocentric hand track and object-based human action recognition," in *Proc. 19th IEEE Conf. Ubiquitous Intell. Comput.*, 2019, pp. 922–929.

[65] Y. Shen, B. Ni, Z. Li, and N. Zhuang, "Egocentric activity prediction via event modulated attention," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 202–217.

[66] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krähenbühl, and R. Girshick, "Long-term feature banks for detailed video understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 284–293.

[67] G. Kapidis, R. Poppe, E. van Dam, L. P. J. J. Noldus, and R. C. Veltkamp, "Object detection-based location and activity classification from egocentric videos: A systematic analysis," in *Smart Assisted Living*. Cham, Switzerland: Springer, 2020, pp. 119–145.

[68] A. Cartas, J. Luque, P. Radeva, C. Segura, and M. Dimiccoli, "Seeing and hearing egocentric actions: How much can we learn?" in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2019, pp. 4470–4480.

[69] F. Xiao, Y. J. Lee, K. Grauman, J. Malik, and C. Feichtenhofer, "Audiovisual slowfast networks for video recognition," 2020, *arXiv: 2001.08740*.

[70] S. Sudhakaran and O. Lanz, "Attention is all we need: Nailing down object-centric attention for egocentric activity recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2018, p. 229. [Online]. Available: http://bmvc2018.org/contents/papers/0756.pdf

[71] A. Furnari and G. Farinella, "What would you expect? Anticipating egocentric actions with rolling-unrolling LSTMs and modality attention," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 6251–6260.

[72] Y. Huang, M. Cai, Z. Li, F. Lu, and Y. Sato, "Mutual context network for jointly estimating egocentric gaze and actions," *IEEE Trans. Image Process.*, vol. 29, pp. 7795–7806, 2020, doi: 10.1109/TIP.2020.3007841.

[73] S. Sudhakaran, S. Escalera, and O. Lanz, "LSTA: Long short-term attention for egocentric action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9946–9955.

[74] H. R. Tavakoli, E. Rahtu, J. Kannala, and A. Borji, "Digging deeper into egocentric gaze prediction," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2019, pp. 273–282.

[75] H. Yu, M. Cai, Y. Liu, and F. Lu, "What I see is what you see: Joint attention learning for first and third person video co-analysis," in *Proc. ACM Int. Conf. Multimedia*, 2019, pp. 1358–1366.

[76] Z. Li, Y. Huang, M. Cai, and Y. Sato, "Manipulation-skill assessment from videos with spatial attention network," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2019, pp. 4385–4395.

[77] M. Lu, D. Liao, and Z.-N. Li, "Learning spatiotemporal attention for egocentric action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2019, pp. 4425–4434.

[78] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3994–4003.

[79] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid, "Joint learning of object and action detectors," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2001–2010.

[80] O. Sener and V. Koltun, "Multi-task learning as multi-objective optimization," in *Proc. 32nd Int. Conf. Neural Inf. Process. Syst.*, 2018, pp. 525–536.

[81] S. Liu, E. Johns, and A. J. Davison, "End-to-end multi-task learning with attention," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1871–1880.

[82] D. C. Luvizon, D. Picard, and H. Tabia, "2D/3D pose estimation and action recognition using multitask deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 5137–5146.

[83] P. Parmar and B. T. Morris, "What and how well you performed? A multitask learning approach to action quality assessment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 304–313.

[84] K.-K. Maninis, I. Radosavovic, and I. Kokkinos, "Attentive single-tasking of multiple tasks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1851–1860.

[85] G. Strezoski, N. Noord, and M. Worring, "Many task learning with task routing," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 1375–1384.

[86] D. Ghadiyaram, D. Tran, and D. Mahajan, "Large-scale weakly-supervised pre-training for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12 038–12 047.

[87] L. Kaiser *et al.*, "One model to learn them all," 2017, *arXiv: 1706.05137*.

[88] I. Kokkinos, "UberNet: Training a 'universal' convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5454–5463.

[89] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[90] K. Pfeiffer, A. Hermans, I. Sárándi, M. Weber, and B. Leibe, "Visual person understanding through multi-task and multi-dataset learning," in *Pattern Recognit.*, pp. 551–566, 2019.

[91] Y. Wu and K. He, "Group normalization," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[92] E. Chong, N. Ruiz, Y. Wang, Y. Zhang, A. Rozga, and J. M. Rehg, "Connecting gaze, scene, and attention: Generalized attention estimation via joint modeling of gaze and scene saliency," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 397–412.

[93] H. Guo, T. Tang, G. Luo, R. Chen, Y. Lu, and L. Wen, "Multi-domain pose network for multi-person pose estimation and tracking," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2019, pp. 209–216.

[94] T. Perrett and D. Damen, "Recurrent assistance: Cross-dataset training of LSTMs on kitchen tasks," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2017, pp. 1354–1362.

[95] H. Coskun *et al.*, "Domain-specific priors and meta learning for low-shot first-person action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Feb. 2021, doi: 10.1109/TPAMI.2021.3058606.

[96] S. Hosseini, M. A. Shabani, and N. I. Cho, "Distill-2MD-MTL: Data distillation based on multi-dataset multi-domain multi-task frame work to solve face related tasksks, multi task learning, semi-supervised learning," 2019, *arXiv: 1907.03402*.

[97] A. Nibali, Z. He, S. Morgan, and L. Prendergast, "Numerical coordinate regression with convolutional neural networks," 2018, *arxiv: 1801.07372*.

[98] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2017, pp. 464–472.

[99] X. Wang, Y. Wu, L. Zhu, and Y. Yang, "Symbiotic attention with privileged information for egocentric action recognition," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 12249–12256.

Georgios Kapidis (Student Member, IEEE) received the diploma degree in electrical and computer engineering and the MSc degree from the Democritus University of Thrace, Komotini, Greece, in 2014 and 2016, respectively. He is currently working toward the PhD degree from the Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands. He was a visiting researcher with Imperial College London and FAU University of Erlangen-Nürnberg. His research interests include egocentric action recognition and object detection. He is a Marie Skłodowska-Curie fellow under ITN project ACROSSING.

Ronald Poppe (Member, IEEE) received the PhD degree in computer science from the University of Twente, Enschede, The Netherlands, in 2009. He was a visiting researcher with the Delft University of Technology, Stanford University and the University of Lancaster. He is currently an assistant professor at the Information and Computing Sciences Department, Utrecht University. His research interests include modeling of visual attention and the analysis of human (interactive) behavior from videos and other sensors. In 2012 and 2013, he received the most cited paper award from Image and Vision Computing. In 2017, he received a TOP grant from the Dutch Science Foundation.

Remco C. Veltkamp (Member, IEEE) is currently a full professor of Game and Media Technology, Utrecht University, The Netherlands. His research interests include the analysis, recognition and retrieval of, and interaction with, games, 3D objects and scenes, images, video, and music. His interests include particular the algorithmic and experimentation aspects, with a special focus on game research. He is director of the Utrecht Center for Game Research, http://www.gameresearch.nl/, he has published more than 250 refereed papers in journals and conferences, and supervised 24 PhD theses.