S. A. Komkov, M. D. Dzabraev, A. A. Petiushko, Mutual modality learning for video action classification,
Компьютерная оптика, 2023, том 47, выпуск 4, 637–649

Mutual modality learning for video action classification

S.A. Komkov [1,2], M.D. Dzabraev [1,2], A.A. Petiushko [1]
[1] Lomonosov Moscow State University, 119991, Russia, Moscow, Leninskie Gory GSP-1;
[2] Huawei Moscow Research Center, 121099, Russia, Moscow, Smolenskaya ploshchad 7–9

Abstract

The construction of models for video action classification progresses rapidly. However, the performance of those models can still be easily improved by ensembling with the same models trained on different modalities (e.g. Optical flow). Unfortunately, it is computationally expensive to use several modalities during inference. Recent works examine the ways to integrate advantages of multi-modality into a single RGB-model. Yet, there is still room for improvement. In this paper, we explore various methods to embed the ensemble power into a single model. We show that proper initialization, as well as mutual modality learning, enhances single-modality models. As a result, we achieve state-of-the-art results in the Something-Something-v2 benchmark.

Keywords: video recognition, video action classification, video labeling, mutual learning, optical flow.

Citation: Komkov SA, Dzabraev MD, Petiushko AA. Mutual modality learning for video action classification. Computer Optics 2023; 47(4): 637-649. DOI: 10.18287/2412-6179-CO-1277.

Introduction

Video Recognition has progressed a lot during the last several years. Datasets have enlarged from thousands of clips [1, 2] to hundreds of thousands [3, 4] and even to hundreds of millions [5]. Neural network-based approaches for video processing evolved from simple 3D-convolutions [6] to Parvo- and Magnocellular counterparts emulation [7] and absorbed developments of classical Image Recognition [8, 9].

Nevertheless, classical results in the domain of Video Processing are still useful: Optical Flow estimation for a video sequence can significantly improve the quality of video recognition [10]. However, the common ways to estimate Optical Flow require an amount of calculation that is comparable to the whole further neural network inference. That is why a number of works are devoted to the implicit Optical Flow estimation during the RGB-based neural network inference [11 – 14].

In our work, we provide extensive experimental research on the ways to embed multi-modality ensemble power into a single model without modifications to its architecture. There are recent distillation-based works dedicated to this task [12, 14]. However, two classical approaches: Mutual Learning (ML) [15] and initialization with the weights pre-trained on other modality were not touched in the literature on the mentioned problem.

We find out that omitted methods mentioned above outperform the ones examined previously. To this end, we introduce Mutual Modality Learning (MML) which is an extension of ML that can deal with inputs from different modalities. We show that MML with the proper initialization boosts a single model with RGB input better than existing methods. Moreover, MML allows us to simultaneously improve different single-modality models in contrast to the prior art. Our single model with RGB input achieves the state-of-the-art (SOTA) results among single-modality models reported previously in the Something-Something-v2 (SmSm-v2) benchmark [4].

Additionally, we examine how to use ML to achieve the best results of the multi-modality ensemble. Based on our experiments, we construct an ensemble that achieves SOTA results among all approaches reported previously in SmSm-v2.

We show that MML works with various modalities, architectures and loss functions. That is why we believe that our findings may be useful for the improvement of a wide range of video recognition models. To ensure reproducibility, we make our code publicly available (*https://github.com/papermsucode/mutual-modality-learning*).

1. Related work

1.1. 3D-approaches

A video sequence is a 4d-tensor with the following parameters: height of frames, width of frames, number of frames, and number of channels per frame (3 in case of RGB input). Therefore, we can process it using Convolutional Neural Networks (CNNs) where 3d-convolutions are applied instead of 2d-convolutions (with the new temporal dimension). Tran *et al.* are the first to propose 3d-convolutional networks based on this idea [6].

Carreira and Zisserman propose to inflate the trained weights of the Image Recognition network and to use them as initialization for 3d-CNN [8]. Nowadays, this is a common approach for video model initialization.

Wang *et al.* implement an attention mechanism that helps to find dependencies between far positions on different frames [16].

To reduce the number of parameters of 3d-CNNs, the first convolutions can be replaced by the per-frame 2d-convolutions [17, 18]. Also, 3d-convolutions can be decomposed as 2d-spatial-convolutions plus 1d-temporal-convolutions [17, 19].

Feichtenhofer *et al.* present a SlowFast Network architecture that emulates Parvo- and Magnocellular counterparts by sampling video frames with two different framerates and by feeding them to two branches with different computational power [7].

The Temporal Pyramid Networks (TPN) of Yang *et al.* can be viewed as an extension of SlowFast networks [20]. A thinned out frames sequence flows to the different branches from intermediate layers instead of entering from the input.

1.2. 2D-approaches

The early CNN-based models for video with 2d-convolutions consist of two streams. The first stream called Spatial takes RGB frames as an input. The second stream called Temporal takes a stack of consecutive Optical Flow estimations [10, 21]. The final prediction is an average of the predictions of both streams.

Nowadays, the idea of features sharing between frames is used to simulate a 3d-inference using 2d-convolutions. The pioneering work in this scope is Temporal Shift Modules network (TSM) by Lin *et al.* that applies ordinary 2d-ResBlocks to each input frame [9, 22]. The single difference is that TSM replaces a one-eighth of channels with the same channels from the previous frame and another one-eighth of channels with the same channels from the future frame before each first convolution of the ResBlock.

Based on the idea of feature sharing, Shao *et al.* present Temporal Interlacing Network [23].

1.3. Transformer-based approaches

Recent studies show that transformer-based architectures, borrowed from Natural Language Processing [24], may also be beneficial for Computer Vision tasks [25]. In particular, some developments in the area of transformer-based video action classification overtake previous convolution-based approaches in terms of accuracy [26, 27].

We did not analyze the performance of our method for these types of architectures since the raise of transformer-based architectures happened after the original research presented in our paper. However, we believe the MML approach applies to transformer-based architectures since it is architecture-agnostic.

1.4. Optical flow distillation

Despite all the aforementioned progress, most of works can be improved by averaging of their predictions with the predictions of the same network trained on the Optical Flow modality [8, 9, 17 – 19, 21].

Since the Optical Flow calculation is a time-consuming operation, a number of works is devoted to incorporation of the motion-estimation blocks inside the CNN architecture [11, 13, 28]. However, knowledge distillation from the Optical Flow modality to any RGB single-modality network seems to be of more interest.

Three basic works that should be mentioned are Knowledge Distillation (KD) [29], ML [15], and Born-Again Networks (BAN) [30].

The first proposes to use soft-predictions of the model called Teacher network to train the smaller model called Student network. It turns out that this technique is helpful for video action classification task not as a neural network compression method but as a transfering of modality knowledge. Zhang *et al.* use KD to train a two-stream network with Motion Vector as the second modality [31]. Stroud *et al.* confirm by constructing Distilled 3D Networks (D3D) that KD from the Optical Flow stream improves the quality of the RGB stream [14]. Motion-Augmented RGB Stream (MARS) of Crasto *et al.* distills the knowledge not from the prediction of the Optical Flow stream but from its feature maps before the global averaging operation [12].

In contrast to the mentioned works, we utilize the idea of ML to train jointly several single-modality networks and improve the quality of each of them (subsection 3.5). Motivated by BAN, we show that the relaunch of training procedure can further boost the performance of models (subsection 3.4). Additionally, we show that proper initialization improves our results (subsection 3.2) as well as results for MARS and D3D works (subsection 3.3).

Note that we target on the single-modality model quality. The improvement of the average predictions of several streams is a different branch of research. An example of an approach that addresses this problem is Gradient-Blending [32]. Nevertheless, we examine the ability of ML to improve the average prediction the multi-modality ensemble. It turns out that proposed initialization with relaunches of single-modality ML provides the best result for the ensemble.

2. Proposed solution

The proposition of the best single-modality model training pipeline is depicted in Figure 1. The pipeline consists of three parts: initialization preparation, ML implantation and MML. The importance of each part is confirmed in section 3.

2.1. Initialization preparation

The standard starting point for the video action classification models training is an ImageNet [33] pretrained model. Inflating of 2d-convolutions proposed in [8] makes that possible for both 3d-models and 2d-models.

If we use the input modality different from RGB then we have to change the shape of the first convolution from $(C,3,K,K)$ to (C,N,K,K). Here, C is a number of output channels of the first convolution, K is a kernel size and N is a number of channels of the new input. The pseudocode for the weights of the new convolution is as follows:

```
for i in 1:N do
W_new[:,i] = (W[:,1]+W[:,2]+W[:,3])/3
end
```

In the proposed pipeline, we use ImageNet initialization only for the first step. The next two steps use weights from the previous step (with a change in the first convolution shape if it is needed).

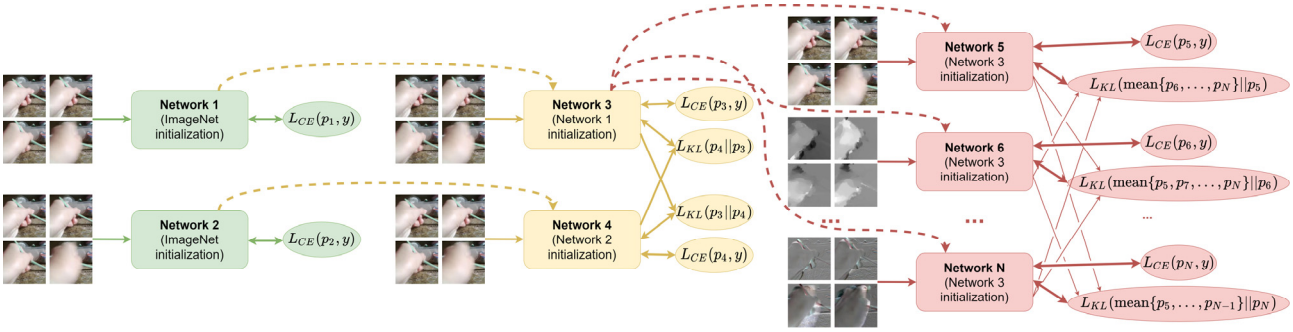

Fig. 1. Best viewed in color. Solid arrows denote flows of data. Dashed arrows denote weights transferring for initialization. Green part: first, we train two networks with RGB input initialized by ImageNet weights using cross-entropy loss. Yellow part: next, we use weights from the first step as initialization for two networks with RGB input that are trained jointly using Mutual Learning. Red part: finally, we apply Mutual Modality Learning to obtain the best single-modality model for each modality. We use weights of the network from the second step as initialization for each model in the third part

2.2. Mutual learning implantation

ML is a technique of training two models together in a way that they help each other to reach better convergence. To achieve that, we modify the loss functions of the networks as follows:

$$L_1 = L_{CE}\left(p_1, y\right) + L_{KL}\left(p_2 \middle\| p_1\right), \tag{1}$$

$$L_2 = L_{CE}\left(p_2, y\right) + L_{KL}\left(p_1 \middle\| p_2\right). \tag{2}$$

Here, L_i is a loss of the i-th network, p_i is a vector of the predicted class probabilities by the i-th network, y is a ground-true class label, L_{CE} is a cross-entropy loss and L_{KL} is the Kullback Leibler (KL) Divergence loss given by the formula

$$L_{KL}\left(p_i \middle\| p_j\right) = \sum_{n=1}^{N} p_i^n \cdot \log \frac{p_i^n}{p_j^n}. \tag{3}$$

In this formula p_i^n stands for a probability for the n-th class predicted by the i-th model. Thus, models teach each other using dependencies that they found during training and thereby improve their performance.

If there are more than two models involved in ML then the loss function is

$$L_i = L_{CE}\left(p_i, y\right) + L_{KL}\left(\frac{\sum_{j \neq i} p_j}{M-1} \middle\| p_i\right), \tag{4}$$

where M is a number of models.

2.3. Mutual modality learning

In the original ML, both models use the same modality as an input. We propose to use different modalities of the video obtained from the same frames as inputs for different models. Thus, we share the knowledge obtained from one modality to other modalities.

Note that we need two consecutive frames to calculate the Optical Flow. Thus, if there are N RGB frames in total then there are only $N-1$ Optical Flow frames in total.

So, suppose that the model requires T input-frames for the prediction and we have two representations of the video by different modalities: one representation with n frames and another with N frames ($N > n$).

For this and similar cases in our work, we first sample frames with numbers $(i_1, \ldots, i_T)$ for the modality with the least number of frames, and then we use frames with numbers $(i_1+\xi, \ldots, i_T+\xi)$ for the modality with the biggest number of frames. Here $\xi \sim \text{unif}\{0, \ldots, N-n\}$.

3. Ablation studies

There are several conclusions that we make:

- Initialization with the RGB model trained on the same video dataset enhances the performance for various modalities and training scenarios (not only ML but MARS and D3D also).
- MML performs better than MARS or D3D approaches.
- Two iterations of ML are better than one and there is no need for more.
- MML performs better than ML as a final step.
- The behavior described above preserves when we use modalities different from the Optical Flow.

3.1. Experiment setup

For the ablation studies, we use "TSM on SmSm-v2" with the code provided by the authors (the main setup, we use it unless otherwise specified) and "I3D [8] on Charades [34]" with the code provided in [7].

We obtain the Optical Flow using the TV-L1 algorithm [35] since it is a generally accepted algorithm of Optical Flow estimation for video recognition [8, 9, 21, 28]. Then we combine 5 consecutive evaluations of the Optical Flow by the x- and y- axes as one input-frame.

For the RGBDiff modality, we take 6 consecutive RGB frames to obtain 5 consecutive differences between them. Obtained differences are concatenated and considered as one input-frame.

3.1.1. TSM on Something-Something-v2

We use the standard setup for the TSM+ResNet-50 training proposed by the authors with batch size 64, ImageNet pretrain, 0.025 initial learning rate. The only difference is the frames sampling strategy. Instead of using one sampling strategy, we use both uniform sampling and dense sampling. The first one works as follows: we split video into T equal parts and take a random frame from each of them. Dense sampling requires taking each τ-th frame starting from a random position. We apply each of the two sampling strategies with 50 % probability. See Appendix 1 as an explanation for this strategy.

We use single uniform sampling with one spatial 224×224 center crop during testing for the ablation studies. That is why the baseline result is worse than the same in [9] where 256×256 central crop is used during testing.

3.1.2. I3D on Charades

This setup is used to show the advantages of MML regarding other approaches. Both D3D and MARS deal with 3d-models, that is why we use the I3D ResNet-50 model to make a fair comparison with the mentioned methods.

Besides, Charades is the dataset with multiple corresponding classes per one clip, so we show how to extend the proposed MML to the multi-label task.

Optimizer, the number of epochs and other hyperparameters are taken from the standard config-file for the Charades training without any changes. We use model trained on Kinetics-400 [8] as a standard initialization instead of ImageNet initialization.

3.2. Initialization

First, we show that the proposed initialization is an important step. Specifically, initialization with the weights of a model with RGB input trained on the current dataset using cross-entropy loss improves the performance of other single-modality models, MARS and D3D models, MML and ML models.

An abbreviation "Flow from ImageNet" means that we initialize a model that takes Optical Flow as an input with the weights of the model trained on ImageNet. An abbreviation "Diff from RGB" means that we initialize a model that takes differences between RGB frames as an input with the weights of the model with RGB input trained on the current dataset using the cross-entropy loss and initialized by a model trained on ImageNet. We make other abbreviations in a similar way.

We do not include training from scratch into the ablation studies since this is a well-known fact that ImageNet initialization outperforms random initialization for the training of one-stream video models [8, 9].

We can see from Table 1 that RGB initialization outperforms ImageNet initialization in the case of ordinary cross-entropy training of the Flow and Diff models: *55.2 / 84.1* vs. *52.3 / 81.8* and *59.0 / 86.3* vs. *58.7 / 84.4*. At the same time, Flow initialization is useless for RGB models: *58.1 / 84.6* vs. *57.5 / 84.4*.

Tab. 1. Single-modality models trained using cross-entropy with different initializations

Model	Top-1 / Top-5	Model	Top-1 / Top-5
RGB from ImageNet	**58.10 / 84.61**	RGB from Flow	57.53 / 84.42
Flow from ImageNet	52.32 / 81.84	Flow from RGB	**55.19 / 84.14**
Diff from ImageNet	58.74 / 84.39	Diff from RGB	**58.98 / 86.33**

We use weights of "RGB from ImageNet" and "Flow from ImageNet" models from Table 1 as RGB and Flow initializations in all experiments unless otherwise specified. Also, we use "RGB from ImageNet" and "Flow from ImageNet" models as teacher networks for MARS and D3D experiments.

We apply MARS and D3D approaches in both directions for RGB and Flow models. Table 2 shows that RGB initialization improves results in each scenario. It should be noted that both MARS and D3D approaches mainly target 3d-models. That is why we do not directly compare MML with MARS and D3D at this point.

Tab. 2. The first column with results: the performance of models trained by the MARS approach using different modalities and initializations. The second column: the performance of models trained by the D3D approach

Model	Teacher modality	MARS training Top-1 / Top-5	D3D training Top-1 / Top-5
RGB from ImageNet	Flow	57.56 / 84.39	58.99 / 85.18
RGB from RGB	Flow	**59.11 / 85.24**	**59.95 / 85.86**
Flow from ImageNet	RGB	57.46 / 85.01	55.04 / 83.36
Flow from RGB	RGB	**58.23 / 85.37**	**56.41 / 83.98**

Tab. 3 and 4 are related to the MML training. We train Flow and RGB models together using MML with all possible pairs of initializations. The results of the RGB models trained using MML are presented in Tab. 3. The results of the Flow models trained using MML are presented in Tab. 4.

Tab. 3. Results of the models with RGB input trained using MML. The columns correspond to different initializations of the model with Flow input. The rows correspond to different initializations of the model with RGB input

RGB results	Flow from ImageNet	Flow from Flow	Flow from RGB
RGB from ImageNet	56.25 / 84.07	60.02 / 86.08	58.70 / 85.18
RGB from RGB	**60.80 / 86.47**	**60.94 / 86.67**	**60.82 / 86.75**
RGB from Flow	58.37 / 84.82	58.56 / 85.28	58.62 / 85.36

Tab. 4. Results of the models with Flow input trained using MML. The columns correspond to different initializations of the model with Flow input. The rows correspond to different initializations of the model with RGB input

Flow results	Flow from ImageNet	Flow from Flow	Flow from RGB
RGB from ImageNet	54.94 / 83.82	57.06 / 84.87	**57.84 / 85.15**
RGB from RGB	54.76 / 83.61	56.74 / 84.78	**57.95 / 85.44**
RGB from Flow	55.85 / 84.33	56.86 / 84.80	**57.79 / 85.34**

As we can see, the middle values of each column in Table 3 are the best as well as the right values of each row in Tab. 4. Hence, RGB initialization for the RGB model during MML is the best regardless of the initialization of the second model. The same is for the Flow model initialization. Thus, the consistency of better initialization is preserved in the case of MML.

Finally, even if we train models on one modality using ML then RGB initialization is still the best. The first three rows and the last two rows of Tab. 5 independently confirm that.

3.3. MML versus MARS and D3D

Although MML outperforms MARS and D3D in "TSM on SmSm-v2" setup (60.8 / 86.7 vs. 59.1 / 85.2 and 59.9 / 85.9), we expand our experiments to make sure of preserving the dependency. MARS and D3D works target mainly 3d-models. That is why we use the "I3D on Charades" setup in this subsection to make a fair comparison of the methods.

We also use a reduced pipeline of the MML that is depicted in Fig. 2. Thus, we train a single-modality model first and then use both modalities for the final training. Weights from the first step are used as initialization for both models on the second step. There fore the reduced MML requires two steps as well as MARS and D3D.

Best viewed in color. Green part: first, we train a network with RGB input initialized by ImageNet weights using cross-entropy loss. Red part: we apply MML to enhance the model from the first step. We use weights of the network from the first step as initialization for both models in the second step.

Tab. 5. Results of the same-modality models trained using ordinary ML. We use abbreviations ImageNet2 and RGB2 to point out that we use different initialization obtained in the same way (KL loss is equal to zero otherwise)

First model	Top-1 / Top-5	Second model	Top-1 / Top-5
RGB from ImageNet	57.76 / 84.42	RGB from ImageNet2	58.15 / 84.64
RGB from RGB	57.84 / 84.55	RGB from ImageNet	60.20 / 86.33
RGB from RGB	**60.54 / 86.23**	RGB from RGB2	60.47 / 86.08
Flow from ImageNet	52.94 / 82.21	Flow from ImageNet2	53.44 / 82.50
Flow from RGB	57.58 / 85.17	Flow from RGB2	**57.71 / 85.26**

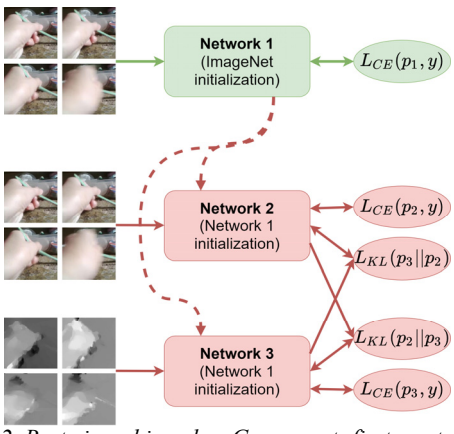

Fig. 2. Best viewed in color. Green part: first, we train a network with RGB input initialized by ImageNet weights using cross-entropy loss. Red part: we apply MML to enhance the model from the first step. We use weights of the network from the first step as initialization for both models in the second step

It should be noted that ordinary KL loss implementation uses the "batchmean" regime of averaging, i.e. we divide the sum of losses by the number of instances in one batch. However, we have to use the "mean" regime of averaging when we train the multi-label model using Binary Cross-Entropy losses (BCE), i.e. we divide the sum of losses by the multiplication of two factors: the number of instances in one batch and the number of classes. See Appendix 2 as an explanation of this point.

By similar reasoning, we divide additional loss functions of MARS and D3D by the number of classes.

The mean Average Precision (mAP) results of all approaches are presented in Tab. 6. The results of the ordinary training of models with RGB and Flow inputs using BCE loss are in the first row: 33.7 for the model with RGB input and 15.8 for the model with Flow input.

We use the Flow model from the first row as a teacher for MARS and D3D experiments and the RGB model from the first row as initialization for the initialization experiments.

Tab. 6. Result of models trained on Charades using BCE, MARS, D3D or MML with and without initialization from a pre-trained model. The right column is empty for methods that do not train a model with Flow input

Training pipeline	RGB model mAP	Flow model mAP
Ordinary training from Kinetics	33.72	15.81
MARS training from Kinetics	28.74	
MARS training from RGB	34.40	
D3D training from Kinetics	33.03	
D3D training from RGB	35.48	
MML training from Kinetics	33.84	17.34
MML training from RGB	**35.96**	**29.12**

The right column in Tab. 6 is empty for MARS and D3D experiments since these approaches do not modify the Optical Flow model during training.

We can see from Tab. 6 again that RGB initialization improves the performance of each method: 28.7 vs. 34.4, 33.0 vs. 35.5 and 33.8 vs. 36.0.

Native MARS and D3D do not use initialization by the pre-trained models. However, MML outperforms even boosted MARS and D3D: 36.0 vs. 34.4 and 35.5.

We assume that performance correlates negatively with the strength of the supervision signal. Since we apply KL loss to probits, then any l_2 distance between logits is possible during MML. Thus, we weakly bound the feature extraction strategy of a network. In the case of D3D training, we minimize l_2 distance between logits only. Thus, D3D does not force a network to estimate the same features in contrast to MARS.

We want to stress that we can significantly improve a single-modality model different from RGB, e.g. MML improves mAP of the Flow model by about 2 times: 29.1 vs. 15.8. We believe that with some further research these findings may be helpful for video recognition by event cameras [36].

3.4. MML versus ML

In this subsection, we confirm the contribution of the second step of our pipeline (yellow part in Fig. 1) and show an advantage of MML in regard to ML.

An abbreviation "RGB from **A**(RGB)" in Tab. 7 means that we initialize an RGB model with the weights of the RGB model tagged as **A** that was trained by MML. We make other abbreviations in a similar way. The right column of Tab. 7 contains tags for the models from the same row.

Tab. 7. Results of MML and ML. Non-standard initialization is used for some experiments comparing to the previous ones: we use weights of models from MML and ML as an initialization. We point out this using tags in the right column: "RGB from A(RGB)" means that we initialize weights by the weights of the RGB model from row tagged as A. We use abbreviations RGB2 and C2(RGB) to point out that we use different initialization obtained in the same way

	Method	*First model*	*Top-1 / Top-5*	*Second model*	*Top-1 / Top-5*	*Tag*
1	MML	*RGB from RGB*	60.82 / 86.75	*Flow from RGB*	57.95 / 85.44	**A**
2	MML	*RGB from RGB*	60.88 / 86.86	*Flow from RGB2*	57.87 / **85.53**	
3	MML	*RGB from **A(RGB)***	61.18 / 86.81	*Flow from **A(RGB)***	58.02 / 85.49	**B**
4	MML	*RGB from **B(RGB)***	61.15 / 86.81	*Flow from **B(RGB)***	57.96 / 85.30	
5	ML	*RGB from RGB*	60.54 / 86.23	*RGB from RGB2*	60.47 / 86.08	**C**
6	ML	*RGB from **C(RGB)***	60.68 / 86.35	*RGB from **C2(RGB)***	60.88 / 86.44	
7	MML	*RGB from **C(RGB)***	**61.30 / 86.99**	*Flow from **C(RGB)***	**58.36** / 85.49	

Motivated by BAN, we explore the possibility of performance improvement by iterative training. Rows number 1 and number 3 from Tab. 7 demonstrate that relaunch of MML can improve the performance: 61.2 / 86.8 vs. 60.8 / 86.7. At the same time, row number 4 demonstrates that the second relaunch is useless: 61.2 / 86.8 vs. 61.2 / 86.8.

Rows number 5 and number 6 demonstrate that iterative training may also be helpful for ML: 60.9 / 86.4 vs. 60.5 / 86.2. Nevertheless, each iteration of ML provides weaker results than the same iteration of MML: 60.5 / 86.2 vs. 60.8 / 86.7 and 60.9 / 86.4 vs. 61.2 / 86.8. Moreover, each additional iteration of ML requires one more launch of the last training. Overwise, KL loss is equal to zero.

We believe that multi-modality training and ML provide their own separate gains. To strengthen both of them, we train models using ML first (yellow part in Fig. 1) and then utilize the advantages of multi-modality training using MML (red part in Fig. 1). Row number 7 confirms that this pipeline achieves the best results.

Finally, rows number 1 and number 2 demonstrate that initialization with different RGB weights does not significantly affect the performance of MML: 60.8 / 86.8 vs. 60.9 / 86.9.

3.5. Other modalities

We expand our experiments to the Diff modality to examine the preservation of the found consistencies.

The last row from Table 1 confirms that RGB initialization is also useful for Diff model.

Row number 5 compared with rows number 4 and number 1 in Tab. 8 confirms that MML is not worse than or even better than single-modality ML in case of RGB and Diff modalities.

Finally, the comparison of row number 6 with rows 1 – 5 in Tab. 8 demonstrates that MML with all three modalities outperforms or is not worse than any other ML in terms of individual results for each modality.

4. Ensemble performance

The predictions of RGB and Flow models can be highly correlated since we train them using KL loss. Thus, an averaging of the predictions may perform worse than the averaging of ordinary RGB and Flow models trained using cross-entropy. The same logic is applicable to MARS or D3D training.

We show results of ensembles of two models in Appendix 3.1 and some results of ensembles of three different models with RGB, Flow and Diff input modalities in Appendix 3.2.

The main conclusions are as follows:

- RGB models that do not use Optical Flow during training perform the best in ensemble with Flow models. Models trained using ML with RGB only are the first, RGB models trained using MML with RGBDiff are the second.
- RGB models that use Optical Flow during training are the worst in the ensemble with Flow models.
- Performance in the ensemble with Flow models from better to worse: MML, D3D, MARS. We believe that this order is caused by the same reasons that are mentioned in the subsection 3.3.
- The same behavior preserves when we combine Flow models with/without RGB signals in loss function during training with RGB models. The only point we want to stress is that "Flow from RGB" models still perform better than "Flow from ImageNet" models in ensembles with RGB models.
- It is also better to combine models trained using single-modality ML when we average the predictions of the RGB and Diff models.
- An ensemble of RGB and Diff models can achieve results that are similar to the results of the RGB and Flow ensemble.
- Models trained using single-modality ML achieve the best results in the ensemble of three different modalities in our experiments. See Appendix 3.2 for more details.

Tab. 8. Experiments with models that use differences between rgb-frames as inputs

Row number	First model	Top-1 / Top-5	Second model	Top-1 / Top-5	
(if used)	Top-1 / Top-5				
1	RGB from RGB	60.54 / 86.23	RGB from RGB2	60.47 / 86.08	
2	RGB from RGB	60.82 / **86.75**	Flow from RGB	57.95 / 85.44	
3	Flow from RGB	57.58 / 85.17	Flow from RGB	57.71 / 85.26	
4	Diff from RGB	60.66 / 87.65	Diff from RGB2	61.07 / 87.73	
5	RGB from RGB	60.52 / 86.52	Diff from RGB	62.13 / 87.57	
6	RGB from RGB	**61.03** / 86.71	Flow from RGB	**58.03 / 85.61**	Diff from RGB **62.51 / 87.95**

Thus, although MML provides the best single-modality models, ordinary ML performs better for ensembles. Considering the aforementioned observations, we propose a pipeline for the best ensemble training that is depicted in Appendix 3.3.

First, we train two "RGB from ImageNet" models using cross-entropy. Second, we launch two single-modality ML procedures for the obtained RGB models. Finally, we train models using single-modality ML for each modality that we want to use in the ensemble.

5. Comparison to state-of-the-art

We make a comparison for three different scenarios. The first one is the single inference of ResNet-50 with 8 input frames. This is a standard scale for ablations for which we can make a direct comparison of the models. The second scenario is the prediction by a single RGB-based model without restrictions on the number of input frames, the number of layers and the number of launches. The final scenario is the performance of model ensembles. We use these three scales since almost all previous models can be assigned to one of these classes. Note that we compare with the approaches published before our original work only.

The overall results are available in Tab. 9. Note that we show not all possible models but the strongest ones.

The first simplest scenario is ResNet-50 as a base architecture with 8 input frames and one launch per prediction. Note that most of the works do not provide testing results for this scenario. That is why we use validation scores for comparison at this stage.

We achieve a strong +2.77 % improvement of the top-1 performance in comparison to the initial TSM solution. We want to stress that our model does not bring additional complexity to the inference.

The only model that slightly outperforms our solution on this scale is TPN. However, this model is much less efficient than ours. Although TPN uses ResNet-50 as a backbone, its additional pyramid structure brings a huge number of additional weights and computations.

We concentrate on testing results for the remaining two scenarios.

Tab. 9. Our results on something-something-v2 in comparison to the prior art

Solution	*Ensemble*	*Backbone architecture*	*Number of input frames*	*Spatial crops × Temporal clips for prediction*	*Top-1 on validation*	*Top-5 on validation*	*Top-1 on test*	*Top-5 on test*
TSM [9]	No	ResNet-50	8	1×1	59.1	85.6	–	–
bLVNet-TAM RGB [37]	Yes	ResNet-50	8	1×1	59.1	86.0	–	–
TIN [23]	No	ResNet-50	8	1×1	60.0	85.5	–	–
TPN [20]	No	ResNet-50	8	1×1	**62.0**	–	–	–
MML (our)	No	ResNet-50	8	1×1	61.87	**87.32**	–	–
STM [28]	No	ResNet-50	8	3×–	62.3	88.8	61.3	88.4
W3 [38]	No	ResNet-50	16	–×2	**66.5**	**90.4**	–	–
bLVNet-TAM RGB [37]	Yes	ResNet-101	32	1×1	65.2	90.3	–	–
STM [28]	No	ResNet-50	16	3×–	64.2	89.8	63.5	89.6
TPN [20]	No	ResNet-101	16	3×2	–	–	**67.72**	91.28
MML (our)	No	ResNet-101	16	1×3	65.9	90.15	66.83	**91.30**
bLVNet-TAM RGB+Flow [37]	Yes	ResNet-101	32+32	3×10	68.5	91.4	67.1	91.4
TSM RGB+Flow [9]	Yes	ResNet-50	16+16	–×–	66.0	90.5	66.55	91.25
RGB-only ensembl by Anonym	Yes	–	–	–×–	–	–	68.18	91.26
rgb+flow by BOE IOT AIBD	Yes	–	–	–×–	–	–	**69.26**	91.81
ML RGB+Flow (our)	Yes	ResNet-101	16+16	1×3	68.16	91.69	–	–
ML RGB+Flow++Diff (our)	Yes	ResNet-101	16+16+16	1×3	**69.07**	**92.07**	69.02	**92.70**

We achieve SOTA results among single models on the Top-5 metric. The only model that outperforms our solution on the Top-1 metrics is also TPN. We have already emphasized that this solution is more computationally expensive due to its pyramid structure. Additionally, it uses twice more launches per one prediction to achieve these results. Thus, our method allows achieving SOTA results without additional computations during inference.

Finally, we make a comparison with other ensembles. As we stressed earlier, the best ensemble performance is not the aim of our research. We target the best possible single RGB-based model. Nevertheless, our findings from section 4 help us to achieve the SOTA-level results among ensembles.

Note that not only academic results are presented for this scenario. Since SmSm-v2 is a public competition, there are many anonymous submissions. Best of anonymous submissions are also presented in Tab. 9.

First, we want to point out that we achieve significant improvement over the best previous published solution. +1.98 %/+1.3 % on the Top-1 and Top-5 metrics respectively. Second, the only anonymous solution that outperforms ours on one of the metrics was submitted 3 months later after our final submission. Thus, our pipeline for the best ensemble achieves the SOTA results among the ones reported previously in the SmSm-v2 benchmark.

Conclusion

We extensively research the ways to embed multi-modality ensemble power into a single-modality model for more efficient Video Recognition. It turns out that a slight modification of the well-known Mutual Learning technique outperforms existing approaches to this task.

We show that Mutual Modality Learning is robust to different modalities, datasets, architectures and loss functions. That is why we believe that it will be helpful for the final fine-tuning of various models in industry, competitions and academia.

Although the method is easily implementable, we make our code publicly available. Using this code, we achieve SOTA result in the Something-Something-v2 benchmark for both scenarios: among single-modality models and among ensembles.

References

[1] Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T. Hmdb: a large video database for human motion recognition. 2011 Int Conf on Computer Vision 2011: 2556-2563. DOI: 10.1109/ICCV.2011.6126543.

[2] UCF101 – Action recognition data set. Source: <https://www.crcv.ucf.edu/research/data-sets/ucf101/>.

[3] Kinetics. Source: <https://www.deepmind.com/open-source/kinetics>.

[4] Goyal R, Kahou SE, Michalski V, et al. The "something something" video database for learning and evaluating visual common sense. 2017 IEEE Int Conf on Computer Vision (ICCV) 2017: 5842-5850. DOI: 10.1109/ICCV.2017.622.

[5] Miech A, Zhukov D, Alayrac J-B, Tapaswi M, Laptev I, Sivic J. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) 2019: 2630-2640. DOI: 10.1109/ICCV.2019.00272.

[6] Tran D, Bourdev L, Fergus R, Torresani L, Paluri M. Learning spatiotemporal features with 3d convolutional

networks. 2015 IEEE Int Conf on Computer Vision (ICCV) 2015: 4489-4497. DOI: 10.1109/ICCV.2015.510.
[7] Feichtenhofer C, Fan H, Malik J, He K. Slowfast networks for video recognition. 2019 IEEE/CVF Int Conf on Computer Vision (ICCV) 2019: 6202-6211. DOI: 10.1109/ICCV.2019.00630.
[8] Carreira J, Zisserman A. Quo vadis, action recognition? a new model and the kinetics dataset. 2017 IEEE Conf on Computer Vision and Pattern Recognition (CVPR) 2017: 6299-6308. DOI: 10.1109/CVPR.2017.502.
[9] Lin J, Gan C, Han S. Tsm: Temporal shift module for efficient video understanding. 2019 IEEE/CVF Int Conf on Computer Vision (ICCV) 2019: 7083-7093. DOI: 10.1109/ICCV.2019.00718.
[10] Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos. NIPS'14: Proc 27th Int Conf on Neural Information Processing Systems 2014; 1: 568-576.
[11] Fan L, Huang W, Gan C, Ermon S, Gong B, Huang J. End-to-end learning of motion representation for video understanding. 2018 IEEE/CVF Conf on Computer Vision and Pattern Recognition 2018: 6016-6025. DOI: 10.1109/CVPR.2018.00630.
[12] Crasto N, Weinzaepfel P, Alahari K, Schmid C. Mars: Motion-augmented rgb stream for action recognition. 2019 IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR) 2019: 7882-7891. DOI: 10.1109/CVPR.2019.00807.
[13] Piergiovanni AJ, Ryoo MS. Representation flow for action recognition. 2019 IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR) 2019: 9945-9953. DOI: 10.1109/CVPR.2019.01018.
[14] Stroud JC, Ross DA, Sun C, Deng J, Sukthankar R. D3d: Distilled 3d networks for video action recognition. 2020 IEEE Winter Conference on Applications of Computer Vision (WACV) 2020: 625-634. DOI: 10.1109/WACV45572.2020.9093274.
[15] Zhang Y, Xiang T, Hospedales TM, Lu H. Deep mutual learning. 2018 IEEE/CVF Conf on Computer Vision and Pattern Recognition 2018: 4320-4328. DOI: 10.1109/CVPR.2018.00454.
[16] Wang X, Girshick R, Gupta A, He K. Non-local neural networks. 2018 IEEE/CVF Conf on Computer Vision and Pattern Recognition 2018: 7794-7803. DOI: 10.1109/CVPR.2018.00813.
[17] Xie S, Sun C, Huang J, Tu Z, Murphy K. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In Book: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, eds. Computer Vision – ECCV 2018. 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV. Cham, Switzerland: Springer Nature Switzerland AG; 2018: 305-321. DOI: 10.1007/978-3-030-01267-0_19.
[18] Zolfaghari M, Singh K, Brox T. Eco: Efficient convolutional network for online video understanding. In Book: Ferrari V, Hebert M, Sminchisescu C, Weiss Y, eds. Computer Vision – ECCV 2018. 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part II. Cham, Switzerland: Springer Nature Switzerland AG; 2018: 695-712. DOI: 10.1007/978-3-030-01216-8_43.
[19] Tran D, Wang H, Torresani L, Ray J, LeCun Y, Paluri M. A closer look at spatiotemporal convolutions for action recognition. 2018 IEEE/CVF Conf on Computer Vision and Pattern Recognition 2018: 6450-6459. DOI: 10.1109/CVPR.2018.00675.
[20] Yang C, Xu Y, Shi J, Dai B, Zhou B. Temporal pyramid network for action recognition. 2020 IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR) 2020: 591-600. IEEE, DOI: 10.1109/CVPR42600.2020.00067.
[21] Wang L, Xiong Y, Wang Z, Qiao Y, Lin D, Tang X, Van Gool L. Temporal segment networks: Towards good practices for deep action recognition. In Book: Leibe B, Matas J, Sebe N, Welling M, eds. Computer vision – ECCV 2016. 14th European conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII. 20-36. Cham, Switzerland: Springer Nature Switzerland AG; 2016. DOI: 10.1007/978-3-319-46484-8_2.
[22] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. 2016 IEEE Conf on Computer Vision and Pattern Recognition (CVPR) 2016: 770-778. DOI: 10.1109/CVPR.2016.90.
[23] Shao H, Qian S, Liu Y. Temporal interlacing network. Proc AAAI Conf on Artificial Intelligence 2020; 34(7): 11966-11973. DOI: 10.1609/aaai.v34i07.6872.
[24] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. 31st Conference on Neural Information Processing Systems (NIPS 2017) 2017: 1-11.
[25] Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houlsby N. An image is worth 16x16 words: Transformers for image recognition at scale. International Conference on Learning Representations (ICLR 2021) 2021: 1-21.
[26] Fan H, Xiong B, Mangalam K, Li Y, Yan Z, Malik J, Feichtenhofer C. Multiscale vision transformers. 2021 IEEE/CVF Int Conf on Computer Vision (ICCV) 2021: 6804-6815. DOI: 10.1109/ICCV48922.2021.00675.
[27] Liu Z, Ning J, Cao Y, Wei Y, Zhang Z, Lin S, Hu H. Video swin transformer. 2016 IEEE Conf on Computer Vision and Pattern Recognition (CVPR) 2022: 3202-3211. DOI: 10.1109/CVPR.2016.297.
[28] Jiang B, Wang M, Gan W, Wu W, Yan J. Stm: Spatiotemporal and motion encoding for action recognition. 2019 IEEE/CVF Int Conf on Computer Vision (ICCV) 2019: 2000-2009. DOI: 10.1109/ICCV.2019.00209.
[29] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv Preprint. 2015. Source: <https://arxiv.org/abs/1503.02531>.
[30] Furlanello T, Lipton Z, Tschannen M, Itti L, Anandkumar A. Born again neural networks. Proc 35th Int Conf on Machine Learning 2018: 1607-1616.
[31] Zhang B, Wang L, Wang Z, Qiao Y, Wang H. Real-time action recognition with enhanced motion vector cnns. 2016 IEEE Conf on Computer Vision and Pattern Recognition (CVPR) 2016: 2718-2726. DOI: 10.1109/CVPR.2016.297.
[32] Wang W, Tran D, Feiszli M. What makes training multi-modal classification networks hard? 2020 IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR) 2020: 12695-12705. DOI: 10.1109/CVPR42600.2020.01271.
[33] Deng J, Dong W, Socher R, Li L-J, Li K, Li F-F. Imagenet: A large-scale hierarchical image database. 2009 IEEE Conf on Computer Vision and Pattern Recognition 2009: 248-255. DOI: 10.1109/CVPR.2009.5206848.
[34] Sigurdsson GA, Varol G, Wang X, Farhadi A, Laptev I, Gupta A. Hollywood in homes: Crowdsourcing data collection for activity understanding. In Book: Leibe B, Matas J, Sebe N, Welling M, eds. Computer Vision – ECCV 2016. 14th European conference, Amsterdam, The

Netherlands, October 11–14, 2016, Proceedings, Part I. Cham, Switzerland: Springer Nature Switzerland AG; 2016: 510-526. DOI: 10.1007/978-3-319-46448-0_31.
[35] Zach C, Pock T, Bischof H. A duality based approach for realtime tv-L1 optical flow. In Book: Hamprecht FA, Schnörr C, Jähne B, eds. Pattern recognition. 29th DAGM Symposium, Heidelberg, Germany, September 12-14, 2007, Proceedings. Berlin, Heidelberg: Springer-Verlag; 2007: 214-223. DOI: 10.1007/978-3-540-74936-3_22.
[36] Gehrig D, Gehrig M, Hidalgo-Carrió J, Scaramuzza D. Video to events: Recycling video datasets for event cameras. 2020 IEEE/CVF Conf on Computer Vision and Pattern Recognition (CVPR) 2020: 3586-3595. DOI: 10.1109/CVPR42600.2020.00364.
[37] Fan Q, Chen C-FR, Kuehne H, Pistoia M, Cox D. More is less: Learning efficient video representations by big-little network and depthwise temporal aggregation. Advances in Neural Information Processing Systems 32 (NeurIPS 2019) 2019: 2264-2273.
[38] Perez-Rua J-M, Martinez B, Zhu X, Toisoul A, Escorcia V, Xiang T. Knowing what, where and when to look: Efficient video action modeling with attention. arXiv Preprint. 2020. Source: <https://arxiv.org/abs/2004.01278>.

Appendix 1. Sampling strategy

The common procedure for the Something-Something-v2 final testing is an averaging of two predictions for each video. For each prediction, we use central full-resolution crop and uniform sampling: we use frames with numbers

$$\left\{\frac{0\cdot T}{N},\ldots,\frac{(N-1)\cdot T}{N}\right\}$$

for the first prediction and frames with numbers

$$\left\{\frac{0.5\cdot T}{N},\ldots,\frac{(N-0.5)\cdot T}{N}\right\}$$

for the second prediction, where T is a total number of frames for current modality and N is the shape of the temporal dimension of the input. Note that uniform sampling, unlike dense sampling, allows any period between input frames and depends on the total length of the video.

We found out that the use of more than two temporal crops with the same sampling strategy or more number of spatial crops insignificantly improves the validation results. At the same time, the use of different sampling strategies during testing significantly improves results regardless of the sampling strategy during training. That is why we incorporate both samplings into training. The median testing results for full-resolution central crops testing are shown in Tab. 10.

Tab. 10. Testing with different sampling strategies

Sampling during training	*Dence 0 + Uniform 1*	*Dence 0 + Uniform 2*	*Dence 1 + Uniform 0*	*Dence 2 + Uniform 0*	*Dence 1 + Uniform 1*	*Dence 2 + Uniform 1*	*Dence 1 + Uniform 2*	*Dence 2 + Uniform 2*
Dense sampling	57.33 / 84.51	58.79 / 85.68	57.61 / 84.98	**58.70** / **85.66**	59.71 / 86.57	60.07 / 86.47	60.11 / 86.56	60.27 / 86.61
Uniform sampling	59.86 / **86.14**	61.16 / **87.03**	56.25 / 83.72	56.20 / 84.18	60.64 / 85.58	59.69 / 86.38	61.50 / 87.32	61.03 / **87.10**
Both samplings	**60.11** / 85.79	**61.38** / 86.82	**57.80** / **85.05**	58.66 / 85.32	**61.10** / **86.66**	**61.01** / **86.57**	**61.71** / **87.40**	**61.59** / 86.97

Label "Dense k + Uniform m" means that we use $k+m$ predictions per video using frames with numbers

$$\left\{\frac{i\cdot T'}{k},\frac{i\cdot T'}{k}+\tau,\ldots,\frac{i\cdot T'}{k}+\tau\cdot(N-1)\right\},\quad i\in\{0,\ldots,k-1\},$$

when $k>1$ or frames with numbers

$$\left\{\left\lfloor\frac{T'}{2}\right\rfloor,\left\lfloor\frac{T'}{2}+\tau\right\rfloor,\ldots,\left\lfloor\frac{T'}{2}+\tau\cdot(N-1)\right\rfloor\right\},$$

when $k=1$ and frames with numbers

$$\left\{\left\lfloor\frac{i/m\cdot T}{N}\right\rfloor,\ldots,\left\lfloor\frac{(N-1+i/m)\cdot T}{N}\right\rfloor\right\},\quad i\in\{0,\ldots,m-1\}.$$

Here T is a total number of frames for current modality, N is the shape of the temporal dimension of the input, $\tau<(T/(N-1))$ is a dense for the dense sampling and $T'=T-\tau\cdot(N-1)$. Note that there is no random nature in frame numbers during testing.

We make the next conclusions based on Tab. 10:

- Dense sampling training is not suitable for the Something-Something-v2.

- Uniform sampling training and Both samplings training are nearly equal if we use prediction for Uniform sampling.
- Both samplings training outperforms Uniform sampling strategy by up to one percent when tested with both strategies.
- It is better to average predictions for two Uniform samplings and one Dense sampling during testing.

Appendix 2. Loss modification for the BCE training

The ordinary implementation of the KL loss divides the sum of $B \cdot N$ terms by the B, where B is a batch size and N is a number of classes. The reason for that is that ordinary Cross-Entropy loss also divides the sum of $B \cdot N$ terms by the B, which can be unobvious:

$$L_{CE} = \frac{1}{B} \cdot \sum_{b=1}^{B} -\log \frac{\exp(l_b^{\mathrm{gt}_b})}{\sum_{j=1}^{N} \exp(l_b^j)} = \frac{1}{B} \cdot \sum_{b=1}^{B} \sum_{i=1}^{N} -y_b^i \cdot \log \frac{\exp(l_b^i)}{\sum_{j=1}^{N} \exp(l_b^j)} = \frac{1}{B} \cdot \sum_{b=1}^{B} \sum_{i=1}^{N} -y_b^i \cdot \log\, p_b^i = \frac{1}{B} \cdot H\left(y, p\right).$$

Here l_b^i is a predicted logit for the class number i for the instance number b, gt_b – ground truth class for the instance number b,

$$y_b^i = I_{\mathrm{gt}_b}\left(i\right) \text{ and } p_b^i = \frac{\exp(l_b^i)}{\sum_{j=1}^{N} \exp(l_b^j)}.$$

So the magnitudes of the CE loss and KL loss are the same. Since the multi-label BCE loss is divided by the $B \cdot N$:

$$L_{mlBCE} = \frac{1}{B \cdot N} \cdot \sum_{b=1}^{B} \sum_{i=1}^{N} -\left(y_b^i \cdot \log\sigma\left(l_b^i\right) + \left(1 - y_b^i\right) \cdot \log\left(1 - \sigma\left(l_b^i\right)\right)\right),$$

then we divide the KL loss by the $B \cdot N$ to make the magnitudes the same again.

The authors of the MARS and D3D approaches found the best weights for their loss functions in the case of the Cross-Entropy training (50 for MARS and 1 for D3D). Our experiments confirm that additional division of the loss by the number of classes improves the performance of these two methods in the case of multi-label training according to the reasoning made above.

Appendix 3. Ensembles

A 3.1. Ensembles of two models

Results of the ensembles of RGB and Flow models are depicted in Table 11. Results of the ensembles of RGB and Diff models are depicted in Table 12.

Tab. 11. Ensemble of rgb and flow models

RGB \ Flow	*CE from Ima- geNet*	*MARS from RGB*	*D3D from RGB*	*MML with Flow from RGB (**A**)*	*MML from RGB with Flow from Flow*	*ML from RGB 1 (**B**)*	*ML from RGB 2*	*ML from **B** 1*	*ML from **B** 2*	*MML fwith Flow from **A***	*MML with Flow from **B***	*MML with Diff from RGB*	*MML with Flow and Diff from RGB*
CE from Ima- geNet	63.67/ 88.41	60.40/ 86.68	61.72/ 87.15	62.65/ 88.03	62.56/ 88.02	64.62/ 89.14	64.33/ 88.76	64.45/ 88.82	64.42/ 88.97	62.70/ 88.09	62.96/ 88.19	63.77/ 88.64	63.31/ 88.29
CE from RGB	63.74/ 88.73	61.18/ 87.42	62.12/ 87.91	62.89/ 88.59	62.79/ 88.41	64.34/ 89.44	64.39/ 89.21	64.52/ 89.45	64.45/ 89.35	62.94/ 88.68	63.14/ 88.68	64.00/ 88.94	63.18/ 88.73
MARS from RGB	62.26/ 87.81	61.61/ 87.24	61.92/ 87.66	62.37/ 88.25	62.50/ 88.19	63.57/ 88.84	63.42/ 88.57	63.62/ 88.75	63.58/ 88.71	62.78/ 88.42	62.88/ 88.49	62.98/ 88.44	62.62/ 88.37
MARS from Im- ageNet	62.57/ 87.67	61.28/ 87.21	61.92/ 87.71	62.55/ 88.08	62.61/ 88.02	63.73/ 88.80	63.61/ 88.62	63.76/ 88.76	63.70/ 88.81	62.92/ 88.26	62.96/ 88.38	63.28/ 88.47	62.72/ 88.31
ML with Flow from Im- ageNet	63.41/ 88.44	60.73/ 86.72	62.24/ 87.35	63.13/ 88.26	63.03/ 88.25	64.88/ 89.23	64.67/ 88.93	64.64/ 89.13	64.57/ 89.33	63.32/ 88.43	63.64/ 88.46	64.31/ 88.79	63.56/ 88.38
ML Flow from RGB 1	63.97/ 88.93	61.91/ 87.78	62.78/ 88.31	63.74/ 89.01	63.60/ 88.91	65.19/ 89.77	64.98/ 89.45	65.20/ 89.74	65.06/ 89.72	63.74/ 89.09	64.08/ 89.20	64.66 89.22	64.16/ 89.16
ML Flow from RGB 2	64.23/ 88.99	62.04/ 87.82	62.93/ 88.31	63.89/ 88.93	63.66/ 88.94	65.24/ 89.72	65.11/ 89.48	65.08/ 89.78	65.09/ 89.77	64.02/ 89.19	64.21/ 89.16	64.83/ 89.41	64.28/ 89.09
MML with RGB from RGB	63.47/ 88.41	61.86/ 87.62	62.62/ 88.03	63.22/ 88.53	63.38/ 88.55	64.79/ 89.35	64.45/ 88.91	64.75/ 89.32	64.55/ 89.21	63.53/ 88.75	63.91/ 88.72	64.32/ 88.81	63.82/ 88.78
*MML with RGB from **A***	63.78/ 88.65	62.08/ 87.69	62.89/ 88.21	63.57/ 88.83	63.60/ 88.71	64.98/ 89.47	64.71/ 89.24	64.87/ 89.48	64.96/ 89.40	63.57/ 88.88	64.04/ 88.91	64.35/ 89.16	63.85/ 88.99
*MML with RGB from **B***	63.81/ 88.73	62.20/ 87.71	62.95/ 88.04	63.70/ 88.76	63.74/ 88.70	64.90/ 89.37	64.79/ 89.20	64.88/ 89.27	64.81/ 89.40	63.98/ 88.78	64.00/ 88.84	64.28/ 88.95	64.08/ 88.97
MML with RGB and Diff from RGB	63.62/ 88.61	61.94/ 87.56	62.57/ 88.14	63.41/ 88.64	63.50/ 88.64	64.90/ 89.55	64.71/ 89.20	64.76/ 89.42	64.50/ 89.48	63.59 88.81	63.82/ 88.89	64.43/ 89.06	63.64/ 88.83

"MML with Flow from RGB (**A**)" in the first row means that we use the model with RGB input that was jointly trained using Mutual Modality Learning with the model with Optical Flow input using RGB initialization for both models. Tag **A** means that we use the weights of this model as initialization for other models in the table.

"ML from **B** 2" in the first row means that we use the second model with RGB input that was jointly trained using Mutual Learning with the other (the first) model with RGB input. Both models were initialized by the weights obtained by the procedure with tag **B**

We color the cell on the intersection of the column and the row that are marked "CE from ImageNet" in Table 11 as white since it is the baseline ensemble. The more intense red color is, the higher the top-1 value for the ensemble is. The more intense light blue color is, the lower the top-1 value for the ensemble is.

The analysis of the tables is in section 4.

Tab. 12. ensemble of rgb and diff models

RGB / Diff	*CE from Ima- geNet*	*MARS from RGB*	*D3D from RGB*	*MML with Flow from RGB (**A**)*	*MML from RGB with Flow from Flow*	*ML from RGB 1 (**B**)*	*ML from RGB 2*	*ML from **B** 1*	*ML from **B** 2*	*MML fwith Flow from **A***	*MML with Flow from **B***	*MML with Diff from RGB*	*MML with Flow and Diff from RGB*
CE from Ima- geNet	63.26/ 88.46	63.24/ 88.29	63.55/ 88.59	63.97/ 88.89	63.90/ 88.77	64.54/ 89.33	64.26/ 89.01	64.52/ 89.27	64.48/ 89.14	64.15/ 89.03	64.03/ 89.07	63.94/ 88.86	63.95/ 88.98
CE from RGB	63.10/ 88.37	63.24/ 88.44	63.57/ 88.66	63.64/ 88.61	63.86/ 88.59	64.37/ 88.91	64.03/ 88.70	64.55/ 88.95	64.16/ 88.90	63.87/ 88.91	64.00/ 88.80	63.85/ 88.55	63.83/ 88.59
ML from RGB 1	63.68/ 88.45	64.13/ 88.83	64.11/ 88.88	64.49/ 89.13	64.52/ 89.06	64.87/ 89.38	64.51/ 88.96	65.05/ 89.22	64.86/ 89.22	64.63/ 89.19	64.74/ 89.32	64.30/ 88.92	64.41/ 88.95
ML from RGB 2	63.87/ 88.59	64.29/ 89.03	64.33/ 89.05	64.56/ 89.33	64.75/ 89.25	64.93/ 89.33	64.86/ 89.04	65.08/ 89.26	64.87/ 89.36	65.02/ 89.43	64.92/ 89.36	64.64/ 89.13	64.77/ 89.29
MML with RGB from RGB	62.75/ 88.12	63.70/ 88.62	63.73/ 88.66	63.59/ 88.71	63.92/ 88.73	64.35/ 89.07	63.69/ 88.61	64.30/ 88.89	63.98/ 88.90	64.13/ 88.83	64.04/ 89.00	63.54/ 88.47	63.72/ 88.64
MML with RGB and Flow from RGB	63.48/ 88.40	63.38/ 88.40	63.61/ 88.61	64.13/ 88.97	64.15/ 88.91	64.89/ 89.42	64.46/ 88.99	64.91/ 89.35	64.52/ 89.32	64.30/ 89.12	64.48/ 89.20	64.18/ 88.91	64.18/ 89.01

A. 3.2. Ensembles of three models

We evaluate the validation results for each combination of three models with different input modalities. We sort all the results of the ensembles of three models by the descending order. We show the sum of all indexes of positions for each model in Table 13. So, the smaller value stands in the table the better model is in ensemble with two other modalities. Note that the magnitude of sums vary across the input modalities since there are different numbers of models for each modality are tested.

Tab. 13. The relevance for the ensemble with other modalities

RGB model	Sum of positions	Flow model	Sum of position	Diff model	Sum of positions
ML from **B** 1	9486	ML Flow from RGB 2	22434	ML from RGB 2	42079
ML from RGB 1 (**B**)	10711	ML Flow from RGB 1	25793	ML from RGB 1	49533
ML from **B** 2	13112	MML with RGB from A	28193	CE from RGB	67475
ML from RGB 2	15670	MML with RGB from B	28954	MML with RGB from RGB	67528
MML with Diff from RGB	26918	ML with Flow from ImageNet	28972	MML with RGB and Flow from RGB	69895
MML with Flow from **B**	28069	CE from ImageNet	29238	CE from ImageNet	71179
MML with Flow and Diff from RGB	31281	CE from RGB	32372		
CE from ImageNet	31793	MML with RGB and Diff from RGB	32659		
MML with Flow from **A**	31968	MML with RGB from RGB	34290		
MML from RGB with Flow from Flow	35475	MARS from ImageNet	50577		
MML with Flow from RGB (**A**)	36167	MARS from RGB	54171		
D3D from RGB	46044				
MARS from RGB	50959				

3.3. Ensemble preparation

The pipeline for the best ensemble training is depicted in Fig. 3.

First, we train two "RGB from ImageNet" models using cross-entropy. Second, we launch two single-modality ML procedures for the RGB models from the previous step. Finally, we train models using single-modality ML for each modality (e.g. RGB, Flow, Diff) that we want to use in the ensemble. We use weights of the RGB models from the second step as an initialization for the third step. This is the reason why we have to launch two training procedures on the second step. KL loss is already optimized otherwise.

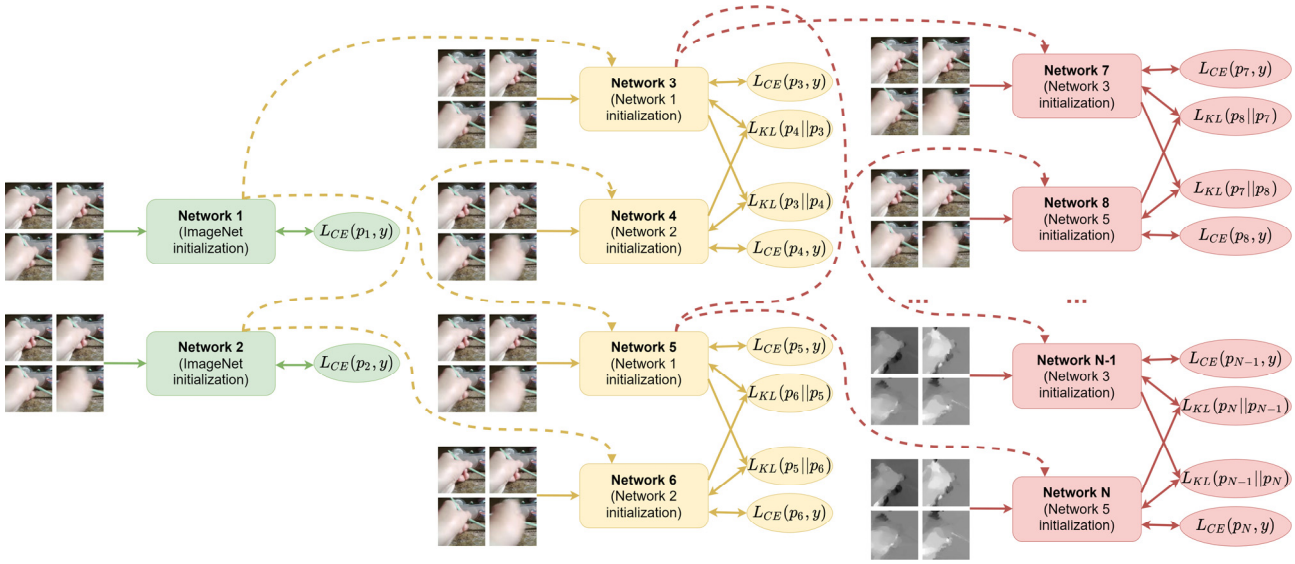

Fig. 3. Best viewed in color. Solid arrows denote flows of data. Dashed arrows denote weights transferring for initialization. Green part: first, we train two networks with RGB input initialized by ImageNet weights using cross-entropy loss. Yellow part: next, we launch RGB-only Mutual Learning for two times. We use the weights from the first step as initialization for each launch of Mutual Learning. We have to use two launches for the second step because we need to obtain two models for which KL loss has not been optimized yet. Red part: finally, we apply single-modality Mutual Learning to each modality that we want to use in the ensemble. We use the weight from one model from each pair from the previous step as the initialization

Authors' information

Stepan Alekseevich Komkov (b. 1994) graduated from Mechanics and Mathematics faculty of Lomonosov Moscow State University in 2018 as a student in Mathematics and in 2022 as a post-graduate student in Mathematics. In 2022 S.A. Komkov defended a PhD thesis on Discrete Mathematics and Mathematical Cybernetics. Currently, he works as a senior research engineer at the Moscow Research Center of Huawei. Research interests are deep learning and discrete mathematics. E-mail: *stepan.komkov@intsys.msu.ru* .

Maksim Dmitrievich Dzabraev (b. 1993) graduated from Mechanics and Mathematics faculty of Lomonosov Moscow State University in 2017 as a student in mathematics. Currently, he works as a senior research engineer at the Moscow Research Center of Huawei. Research interest is deep learning.
E-mail: *dzabraev.maksim@intsys.msu.ru* .

Aleksandr Aleksandrovich Petiushko (b. 1983) graduated from Mechanics and Mathematics faculty of Lomonosov Moscow State University in 2006 as a student in Mathematics and in 2012 as a post-graduate student in Mathematics. In 2016 A.A. Petiushko defended a PhD thesis on Discrete Mathematics and Mathematical Cybernetics. Research interests are deep learning and discrete mathematics. E-mail: *petiushko.aleksandr@intsys.msu.ru* . ORCID profile: 0000-0001-9692-8134. Personal page: *petiushko.info* .

Code of State Categories Scientific and Technical Information (in Russian – GRNTI)): 28.23.37
Received January 13, 2023. The final version – January March 29, 2023.