Omni-sourced Webly-supervised Learning for Video Recognition

Haodong Duan[1], Yue Zhao[1], Yuanjun Xiong[2], Wentao Liu[3], and Dahua Lin[1]

[1] The Chinese University of Hong Kong
[2] Amazon AI
[3] Sensetime Research
dh019,zy017,dhlin@ie.cuhk.edu.hk
bitxiong@gmail.com liuwentao@sensetime.com

1 Datasets

In the main paper, we conduct experiments on three benchmarks, namely Kinetics-400, Youtube-car and UCF101. The detailed statistics of the target and auxiliary datasets are listed in Table 1. Our framework is very data efficient, comparing to approaches which use billion of images, dozens of millions of videos for pretraining. All the web data we collected are only several Tera-Bytes. After filtering, remained web data are only around 3TB, which can easily fit into one hard drive. In stark comparison, the space required by [4] is estimated to be at least 100TB. In this section, we visualize videos in these three datasets, and data in the auxiliary datasets we construct, to show why OmniSource benefits these tasks in different levels.

Table 1: Dataset Statistics. Here we show the statistics of dataset we use in our experiments. We report storage amount of lowest cost format for videos (videos when using high fps for training, and frames when using low fps for training). Our framework is data efficient, the amount of data we used is two orders less than web data pretraining approach. Tri-vid denotes trimmed videos and Unt-vid denotes untrimmed videos.

Target Dataset	Type	Training Size	Storage	Source Dataset	Type	Raw size	Raw storage	Clean Size
Kinetics-400	Tri-Vid	240K 40K mins	140 GB	GG-k400	Img	6M	350 GB	2M
				IG-img	Img	7.4M	450 GB	1.5M
				IG-vid	Tri-Vid	1.1M 480K mins	1.74 TB	500K 250K mins
				k400-untrim	Unt-Vid	670K mins	2.44 TB	500K mins
Youtube-car	Unt-Vid	10K 21K mins	92 GB	GG-car	Img	70K	12 GB	50K
				YT-car-17k	Unt-Vid	28K 63K mins	66 GB	17K 38K mins
UCF101	Tri-Vid	10K 1.2K mins	7 GB	GG-UCF	Img	200K	12 GB	100K

Kinetics-400 We visualize some images in `GG-k400` and some videos in `k400-tr`, `IG-vid` in Fig. 1. The observations are summarized below: (1) Web data have much more diversified appearance comparing to the target dataset. (2) Web data are very noisy. Teacher network tells us that almost 60% - 70% data in the web data is irrelevant to the task we are interested in. (3) We can eliminate noise in web data at the cost of dropping some false negative samples, resulting in a much cleaner auxiliary dataset.

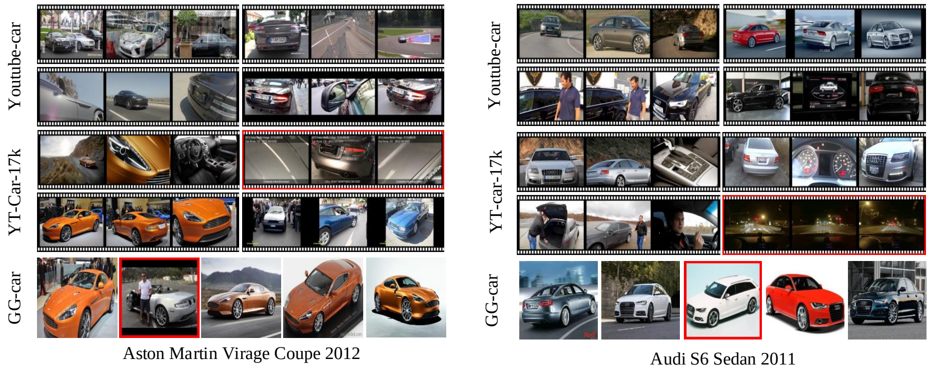

Fig. 2: Youtube-car. Visualization of data in Youtube-car and its auxiliary dataset. Since one can easily get images of centain types of cars by querying its name, the quality of the auxiliary dataset is much better. The high quality web data leads to considerable gain in model performance.

Youtube-Car Youtube-Car is the benchmark on which our framework benefits most. It mainly has two reasons: (1) The web data are much cleaner: when searching with the name of a car, it is easy to get a bunch of images with little noise, since nothing is ambiguous. (2) The source for both target and auxiliary dataset is YouTube, which mean the domain gap is much smaller. Some samples from Youtube-Car and its auxiliary datasets are visualized in Fig. 2.

Fig. 3: UCF101. Visualization of data in UCF101 and its auxiliary dataset. For some classes, web data are more diversified and contain more discriminative poses.

UCF101 Our framework also works on UCF101, which is a small-scale video recognition dataset. UCF101 has much less data diversity and lower visual quality, while auxiliary web data can be complementary in these two aspects. For example, from Fig. 3, one can hardly tell the difference between BreastStroke and FrontCrawl videos in UCF101. The difference is much more significant in web data. Using our framework, models can learn those discriminative features from web data, and can better recognize videos in the target dataset.

2 Implementation Details

Here, we report the implementation details for all our experiments for Kinetics-400 and transfer learning in UCF101 and HMDB51.

Fig. 1: Kinetics. Data from Kinetics and data from auxiliary datasets are visualized, both raw and clean. Red boxes denote that the image is identified as negative by teacher. There might be some false-negative during teacher filtering, but the data filtered out by teacher are almost clean.

2.1 Experiments on Kinetics-400

For all experiments on Kinetics-400, we use an SGD with momentum of 0.9, and weight decay of 10^{-4}. The initial learning rate (LR) we use linearly scales with the number of samples and is decreased to its 10^{-1}. For TSN-2D experiments, we use 4×10^{-5}/sample as the starting LR. The training process lasts 100 epoches and LR decays at 40 and 80 epochs. For 3D-ConvNet experiments, we use 1.6×10^{-4}/sample as the starting LR for experiments with ImageNet-pretrain, 1.6×10^{-3}/sample as the starting LR for train-from-scratch experiments. For ImageNet-pretrain experiments, training lasts 150 epochs and LR decays at 90 and 130 epochs. For train-from-scratch experiments, we use CosineLR schedule instead of StepLR schedule, and training lasts for 256 epoches and 196 epoches respectively for SlowOnly-4x16 and SlowOnly-8x8, same as training schedules used in [3]. For IG-65M pretrained irCSN-152, we use 5×10^{-6}/sample as the starting LR. The training process lasts 58 epochs and LR decays at 32 and 48 epochs, which is consistent with [4]. Warmup is also used in our experiments, which lasts 34 epochs for the train-from-scratch SlowOnly approach, 16 epochs for irCSN-152. During warmup, learning rate grows linearly from 0 to the starting LR. The warmup schedules follows [3,4].

2.2 Experiments for Transfer Learning on UCF-101 and HMDB-51

We use one simple schedule for all transfer learning experiments. We we use an SGD with momentum of 0.9, and weight decay of 10^{-4}. The starting LR is set to 5×10^{-6}/sample. We train 90 epoches on UCF101 and HMDB51 and the first 20 epoches are used for warmup, during which learning rate grows linearly from 0 to the starting LR. No LR decay is performed during training.

3 Experiments

Due to space limitation, some experiment results are not described in detail in the main paper. In this part, we discuss these experiments at length.

3.1 Verifying the efficacy of OmniSource.

Why do we need teacher filtering and are search results good enough? In the main text, we argue that directly using collected web data for joint training leads to a significant performance drop (Top-1 Accuracy: 70.6% to 67.4%) on TSN, which proves the necessity of having a teacher network. However, since we crawl Top 1000 images for each class name from search engines, one may argue that too many queries lead to bad data quality. In response to this question, we construct two subset of `GG-k400-Raw`, which include Top ¼ (`GG-k400-Raw-1/4`) and Top ½ (`GG-k400-Raw-1/2`) results in `GG-k400-Raw` respectively. To make sure web images are much more than trimmed videos in the target dataset, we construct a subset of `k400-tr`, named `k400-tr-half`, which includes half classes and half videos per class. We jointly train `k400-tr-half` with different auxiliary datasets. From Table 2, we see that raw web data are of low quality, even for top search results. Thus teacher filtering is an essential step in OmniSource.

Table 2: Joint training `k400-tr-half` with different raw web datasets. We see that even top search results are of bad quality, lead to inferior performance. Thus teacher filtering is essential

Target Dataset	Source Dataset	Top-1	Top-5
k400-tr-half	/	72.2	90.3
	GG-k400-Raw	70.3	89.2
	GG-k400-Raw-1/2	69.8	88.7
	GG-k400-Raw-1/4	69.9	88.7

Does every data source contribute? In the main text, we use two groups of experiments which use ImageNet pretrained TSN-3seg-R50 and SlowOnly-4x16-R50 as baselines, to prove that every source contributes. Besides that, the conclusion also holds for SlowOnly-4x16-R50 trained from scratch. From Table 3, we see that for the train-from-scratch setting, each data source not only contributes to the target task, but the improvement is much larger than the ImageNet-pretrain setting.

Table 3: For the train-from-scratch setting, every data source also contributes to the target task. The improvement is much larger compared to the ImageNet-pretrain setting. (FT: ImageNet-pretrain; SC: train-from-scratch)

Arch/Dataset	K400-tr	+GG-k400	+GG&IG-img	+IG-vid	+K400-untr	+ All
SlowOnly 4x16, R50 [FT]	73.8/90.9	74.5/91.4	75.2/91.6	75.2/91.7	74.5/91.1	76.6/92.5
SlowOnly 4x16, R50 [SC]	72.9/90.9	74.1/91.0	74.8/91.4	75.8/92.0	74.8/91.2	76.8/92.5

Do features learned by OmniSource transfer to other tasks? In this section, we provide extensive experiment results on transfer learning, much more than results presented in the main text. Table 4 lists transfer learning results on UCF101-split1 and HMDB-split1. Those results further support 2 points proposed in the main text: (1) OmniSource framework can learn better representation, which leads to significant performance improvement on downstream tasks. (2) ImageNet-pretraining is not indispensable for OmniSource to learn good representation. When combined with flow stream, state-of-the-art results on UCF101 and HMDB51 can be achieved by finetuning models jointly trained on Kinetics and auxiliary datasets. Table 5 compares the transfer learning performance of OmniSource trained models with other state-of-the-art approaches. We see that OmniSource outperforms other methods by a large margin.

3.2 Validating the good practices in OmniSource

Impact of teacher choice. In the main paper, we mention that for web video data, 3D teachers always outperform 2D ones. Besides that, the conclusion that the accuracy of the student network increases when a better teacher network is used also holds for web video data. Here, we provide some quantitative results to prove those conclusions in Table 6. SlowOnly-4x16-R50 with ImageNet-pretrain is used as the student network.

Table 4: Detailed results of transfer learning. We report Top-1 accuracies on the official split-1 of UCF101 and HMDB51. We see that OmniSource framework can learn better representation which transfers to other recognition tasks well, even without ImageNet pretraining.

Architecture	w/. ImageNet-pretrain	w/. OmniSource	UCF101-Top1	HMDB51-Top1
TSN-3seg ResNet50	✓		91.51	63.53
	✓	✓	93.29	65.88
TSN-3seg Efficient-b4	✓		92.52	66.27
	✓	✓	93.05	66.54
SlowOnly-4x16 ResNet50	✓		94.69	69.35
	✓	✓	95.98	70.71
			94.05	65.82
		✓	96.01	70.98
SlowOnly-8x8 ResNet101	✓		96.40	76.41
	✓	✓	97.38	78.95
			96.61	75.82
		✓	97.52	79.02

Table 5: We compare transfer learning results with state-of-the-art approaches. We report mean Top-1 accuracies on three splits of UCF101 and HMDB51. We see that OmniSource framework not only outperforms RGB-Only methods. When fused with the flow stream, it surpasses all methods by a large margin, even for those which ensemble results of RGB, Flow and other modalities (*We reimplement Flow-I3D as our flow stream)

Model	Pretrain	UCF101	HMDB51
Two-Stream [5]	ImageNet	88.0	59.4
TSN [6]	ImageNet	94.2	69.4
RGB-I3D[1]	ImageNet + Kinetics	95.6	74.8
Flow-I3D[1]	ImageNet + Kinetics	96.7	77.1
Two-Stream-I3D[1]	ImageNet + Kinetics	98.0	80.7
I3D + PoTion[2]	ImageNet + Kinetics	98.2	80.9
I3D + PA3D[7]	ImageNet + Kinetics	/	82.1
SlowOnly-8x8-R101	Kinetics + OmniSource	**97.3**	**79.0**
SlowOnly-8x8-R101 + Flow[1]	Kinetics + OmniSource	**98.6**	**83.8**

Table 6: More results on the impact of teacher choice. 3D teachers always outperform 2D ones. The accuracy of the student network increases when a better teacher network is used.

Aux. Dataset	Teacher	Teacher Top-1	2D / 3D ?	Top-1	Top-5
IG-vid	TSN-3seg-R50	70.6	2D	73.2	90.8
	SlowOnly-4x16-R50	73.8	3D	75.2	91.7
	IRCSN-152	82.6	3D	75.4	91.9
K400-untr	TSN-3seg-R50	70.6	2D	74.1	91.0
	SlowOnly-4x16-R50	73.8	3D	74.5	91.1
	IRCSN-152	82.6	3D	75.0	91.4

Untrimmed videos to snippets. In the main paper, we mention that combining negative frames and positive frames is a good practice to construct harder snippets, which leads to better recognition performance. We provide detailed results in Table 7, in which we explore each possible combinations during joint training `k400-tr` and `k400-untr` with TSN-3seg-R50 baseline. We find that combining one positive frame and two negative frames to form a 3-frame snippet leads to best performance.

Table 7: We explore different combinations to build a 3-frame snippet, and find that 1 Pos. + 2 Neg. is the best choice.

Configuration	Top-1	Top-5
3 Rand.	71.42	89.34
3 Pos.	71.22	89.54
2 Pos. + 1 Neg.	71.44	89.57
1 Pos. + 2 Neg.	71.66	89.63

4 Improvement Analysis

We further study the improvement of our framework, when using the full auxiliary set for training. Recall that our framework can improve 3.0% and 3.9% respectively on 2D and 3D baseline with all auxiliary data we collected, We analyze the improvement on confusing pairs over these two cases. We use delta of confusion score (Δ_{ij}) to denote the improvement:

$$\Delta_{ij} = Oscore_{ij} - Bscore_{ij}, \tag{1}$$

where $Oscore_{ij}$ denotes the confusion score of pair $< i, j >$ when trained with OmniSource, and $Bscore_{ij}$ denotes the confusion score of pair $< i, j >$ of baseline model.

Case	Action 1	Action 2	$\Delta_{ij} \downarrow$
Success	rock scissors paper	shaking hands	-0.160
	headbutting	sniffing	-0.159
	sweeping floor	mopping floor	-0.113
	eating chips	eating doughnuts	-0.103
	eating ice creams	eating cake	-0.100
Failure	rock scissors paper	slapping	+0.176
	drinking	drinking shots	+0.158

Table 8: Confusion Score Delta for 2D models. Lower delta means larger gain in discriminative power of these two classes. Top-5 and Lowest-2 entries are displayed.

Case	Action 1	Action 2	$\Delta_{ij} \downarrow$
Success	slapping	headbutting	-0.235
	eating doughnuts	eating hotdog	-0.153
	eating chips	eating hotdog	-0.121
	faceplanting	drop kicking	-0.120
	cooking chicken	cooking sausages	-0.110
Failure	baking cookies	making a cake	+0.119
	yawning	sneezing	+0.104

Table 9: Confusion Score Delta for 3D models. Lower delta means larger gain in discriminative power of these two classes. Top-5 and Lowest-2 entries are displayed.

We show success and failure cases of 2D model in Table 8. The contribution of our framework mainly attributes to the better object recognition ability. Besides that, it also improves when discriminative element can be found in web data, like two hands touched in handshaking, two head touched in headbutting, etc.. There are also failure cases when motion is needed for action recognition or when the taxonomy is not reasonable.

We show success and failure cases of 3D model in Table 9. Thanks to the capability of using motion cues for action recognition, the pair 'rock scissors paper' and 'slapping' is no longer a failure case (Δ from +0.176 to -0.059). However, when appearance and motion are all similar, our framework might fail due to the introduced noises.

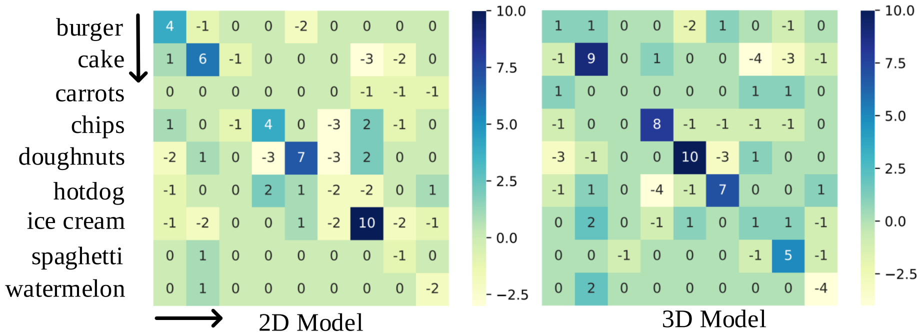

Fig. 4: Improvement on eating something. Rows denote groundtruth and columns denote predictions. Block$_{ij}$ represents the difference in numbers of samples which belongs to class i but recognized as class j between the baseline and our model.

Due to the improved ability of object recognition, the accuracy improvement on actions of eating something is much more significant. On average, the accuracy for eating something improved 5.8%, 8.3% for 2D and 3D models respectively, while the average improvement for all classes are 3.0% and 3.9%. We visualize the improvement on this subset in Fig. 4.

References

1. Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017. 6
2. Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7024–7033, 2018. 6
3. Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019. 4
4. Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12046–12055, 2019. 1, 4
5. Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 6
6. Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 6
7. An Yan, Yali Wang, Zhifeng Li, and Yu Qiao. Pa3d: Pose-action 3d machine for video recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7922–7931, 2019. 6