Geometry Guided Convolutional Neural Networks for Self-Supervised Video Representation Learning

Chuang Gan[1], Boqing Gong[2], Kun Liu[3], Hao Su [4], Leonidas J. Guibas [5]
[1] MIT-IBM Watson AI Lab , [2] Tencent AI Lab, [3] BUPT, [4] UCSD, [5] Stanford University

Abstract

It is often laborious and costly to manually annotate videos for training high-quality video recognition models, so there has been some work and interest in exploring alternative, cheap, and yet often noisy and indirect training signals for learning the video representations. However, these signals are still coarse, supplying supervision at the whole video frame level, and subtle, sometimes enforcing the learning agent to solve problems that are even hard for humans. In this paper, we instead explore geometry, a grand new type of auxiliary supervision for the self-supervised learning of video representations. In particular, we extract pixel-wise geometry information as flow fields and disparity maps from synthetic imagery and real 3D movies, respectively. Although the geometry and high-level semantics are seemingly distant topics, surprisingly, we find that the convolutional neural networks pre-trained by the geometry cues can be effectively adapted to semantic video understanding tasks. In addition, we also find that a progressive training strategy can foster a better neural network for the video recognition task than blindly pooling the distinct sources of geometry cues together. Extensive results on video dynamic scene recognition and action recognition tasks show that our geometry guided networks significantly outperform the competing methods that are trained with other types of labeling-free supervision signals.

1. Introduction

Video understanding is among one of the most fundamental research problems in computer vision and machine learning. The ubiquity of video acquisition devices (e.g., smart phones, surveillance cameras, etc.) has created videos far surpassing what we can watch. It has therefore been a pressing need to develop automatic video analysis and understanding algorithms for various applications.

To recognize actions and events happening in videos, recent approaches that employ deep convolutional neural networks (CNNs) [12, 17, 31, 34, 35], recurrent networks [15, 33, 4], and attention networks [23, 22] have achieved state-of-the-art results. They fall into the paradigm of supervised learning and rely on the existence of large-scale well-labeled training data.

However, it is extremely laborious and costly to manually annotate videos. The actions of interest, for instance "cutting in kitchen", may last for only several seconds in an hour-long video. In order to obtain a training example of this action, the annotator needs to watch through the lengthy video, manually localize those positive frames, and then trim the video. Even with sophisticated GUIs, the labor cost for obtaining one training video sequence is still much higher than that of labeling many images. This problem becomes more severe as the number of action classes grows.

To alleviate the demand for costly human annotations, there has been some work and interest in exploring alternative, cheap, and yet often noisy and indirect training signals. By pre-training a neural network with large-scale data with such supervision signals, a strongly discriminative network can then be obtained afterwards through fine-tuning on a small-scale human annotated dataset. Various signals have been explored in the past [37, 26, 7, 1, 16, 24, 27]. However, these signals are still coarse or vague. The auxiliary signal is often at the whole video frame level rather than pixel level. In addition, some of such supervisions are subtle; for example, the temporal ordering used in [24] is even hard for humans to determine.

In this paper, we explore a new type of auxiliary supervision signal — geometry cues. In particular, we extract pixel-wise geometry information such as flow fields and disparity maps from synthetic images and real 3D movies, respectively. Although the geometry and semantics seem to be two distant topics historically, surprisingly, we find that the network pre-trained by the geometry cues can be well adapted to semantic understanding tasks. Empirical results show that our geometry guided networks significantly outperform the baseline methods pre-trained by other auxiliary signals in the previous work. Intuitively, the signal that can be used to assist semantic understanding has to be strongly correlated with semantics. Our experimental results therefore also indicate the intrinsic correlation between the geometry and semantics.

Two basic observations have motivated our interest in exploring the geometry signal to assist video understanding. First, more and more temporal visual data with geometry information is emerging recently. The increasingly available 3D movies are a primary source. In addition, synthetic imagery from 3D data is now quite close to realistic photos [9]. Not only such synthetic data is large in scale, but also the geometry information is accurate thanks to the known 3D models. An important upcoming source of 3D data is the 3D video streams captured by commodity 3D sensors thanks to their popularization. Second, the geometry information, such as the flow field and disparity value used in this paper, embodies very rich information because depth is a continuous variable and much semantic information can be inferred from it. In addition, it is densely defined at every pixel since every object in the video must have its geometry. Such richness and omnipresence of geometry signal make it possibly superior to previously explored auxiliary signals, e.g., tracking based signals [37] are only available for moving objects but not static objects in a scene.

More fundamentally, our interest is inspired by the perceptual ability of biological agents for sensing the environment. The 3D geometry is almost always available like free food for their binocular visual systems. Is it possible that these agents can use such free data as a source of supervision for learning useful perceptual representations? In addition, prior to gaining the ability to infer semantics which can be as sophisticated as recognizing thousands of object categories or the intention behind actions, their visual systems have to enable inferring the geometry which is vital for them to navigate in space and act upon objects, even with an impaired eye.

Exploiting the geometry information from synthetic images and 3D videos for network pre-training, however, is technically non-trivial. Due to the idiosyncrasies of different data sources, there exist domain gaps of various degrees between the training data and the testing videos of interest. In the context of this paper, on the one hand, the synthetic imagery is very discrepant from the real videos and is yet with pixel-wise accurate geometry cues. The 3D movies, on the other hand, are visually more similar to our testing videos but we can only infer noisy disparity maps from them. The gain of pre-training via blindly pooling them together is only marginal. To tackle this problem, we instead use a progressive training strategy leveraging a learning without forgetting cost function [21]. The idea embodies curriculum learning [3], namely, we carefully organize the training data, teach the network with accurate geometry cues in the synthetic imagery first, and then feed the network real-world appearance information conveyed by the 3D movies. In summary, our work makes the following contributions:

- To the best of our knowledge, we are the first to utilize the geometry cues for the self-supervised learning of video feature representations.
- We propose an end-to-end trainable geometry guided CNN framework, which can leverage different types of labeling-free geometry data: synthetic images and real 3D movies.
- Our geometry guided CNN significantly outperforms other self-supervised approaches and is also complementary to the ImageNet pre-trained models on two publicly available action recognition datasets.

The rest of this paper is organized as follows. In Section 2, we review related work in video recognition and self-supervised learning. Section 3 presents the framework of geometry guide CNN framework. We show how it utilizes the synthesized image data and real 3D movies to learn the generic video feature representations. Experimental settings and evaluation results are presented in Section 4. Section 5 concludes the paper.

2. Related work

Our work touches two threads: video recognition and self-supervised feature learning, which will be discussed respectively.

2.1. Video Recognition

Impressive progress has been achieved in video recognition with the use of deep learning. Karpathy *et al.* [17] first compare several architectures for action recognition. Tran *et al.* [34] propose to learn generic spatial-temporal features with 3D convolutional filters. Simonyan *et al.* [31] propose a two-stream architecture to capture both spatial and motion information with a pixel stream and an optical flow stream respectively. Wang *et al.* [36] further improve the results by using temporal segments and deeper neural networks. Gan *et al.* [12] proposed to learn temporal dynamics using a cross-frame max-pooling layer. However, all these approaches require high-quality labeled training data to initial the network. More recently, Recurrent Neural Networks (RNNs) are shown effective to model temporal information in videos. Donahue *et al.* [8] train a two-layer LSTM network for action classification. Srivastava *et al.* [33] propose an LSTM encoder-decoder framework to learn video representations in an unsupervised manner [33]. However, this approach requires a pre-train on ImageNet to extract frame-level features and thus is not a unsupervised feature learning approach.

2.2. Self-supervised Feature Learning

A recently emerging research line is training a network on an auxiliary task where ground-truth is obtained automatically, called as self-supervised learning[37, 26, 7, 26].

Figure 1. The first two columns show the examples of the FlyingChair dataset [9]. From top to bottom are the first frame, the second frame, and the ground truth optical flow between them, respectively. The last two columns show the examples of 3D movie frames and the estimated disparity map. From top to bottom are the left view image, the right view, and the estimated disparity map.

The merit of this line work does not require manually annotations but still utilized supervised learning by inferring supervisory signals from data structure. For example, Wang *et.al.* [37] proposed to generate pairs-of-patch by tracking objects in videos and then use a Siamese triplet network to learn feature representations that the similarity between two matching patches should be larger than the similarity between two random pairs. Doech *et.al.* [7] explored the spatial consistency of image as context prediction task to learn feature representation. Owens *et.al.* [26] used audio signals from videos to learn visual representations. Other approaches such as [1, 16] have tried to use videos and ego-motion to learn the underlying network. More recently, [24, 11, 20] presented CNN-based unsupervised video representation learning method. In order to capture the temporal information, they design a learning task to verify the sequence of frames is presented in the correct order or not. Despite these approach having to determining the correct temporal order, but do not learn dense pixel-level movement. Different from existing approaches, we use the geometry task to learn the dense pixel-level geometry cues, which can serves as more strong supervised signals to learn robust video feature representation for the semantic recognition task.

Previous work that are most similar in spirit to ours are [1, 27, 2, 38]. They also leveraged the geometry information to learn features for image. Different from them, our work uses pure pixel-level geometry signals to learn features for video semantic understanding.

3. Our Approach

In this section, we describe our geometry guided CNNs for the self-supervised learning of video feature representations. We jointly explore two types of geometry cues in synthetic images and 3D movies, respectively. These cues effectively drive the CNNs to extract generic knowledge from the conventional videos that is useful for the high-level video recognition task.

Next, we describe the details of the two types of geometry cues, followed by the approach to learning deep CNNs from them. Figure 2 illustrates the whole framework of the geometry guided CNN for video feature learning.

3.1. Geometry from synthetic 3D image pairs

The first geometry cue we use is from the renderings of 3D objects. Synthetic images can be generated by rendering virtual 3D objects. Since the 3D models are already given, exact geometry information at pixel-level in rendering can be extracted.

In this paper, we conduct a first study leveraging the geometry information extracted from a very simple synthetic image dataset – the FlyingChairs dataset [9]. Each 3D model is projected twice with a random rigid transformation between the two and superimposed to the same background. Since the relationship in the image pair is known, the pixel-wise correspondence across the two images can be exactly

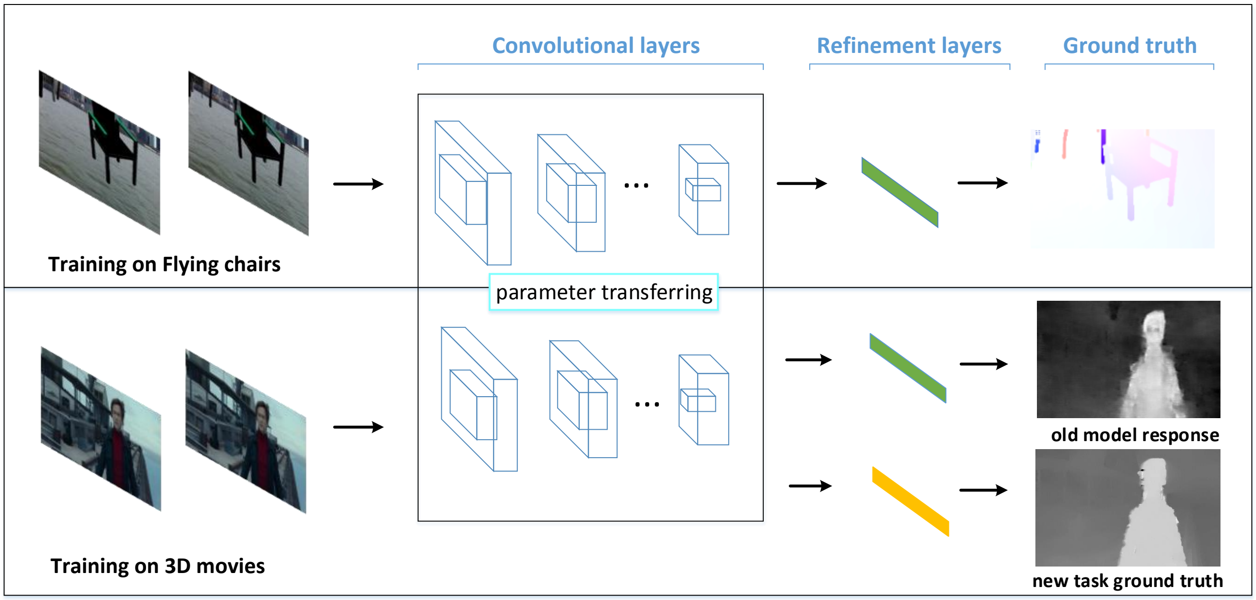

Figure 2. The framework of our proposed geometry guided CNN. We firstly use the synthetic images to train a CNN, and then use the 3D movies to further update the network.

extracted, represented as a flow map. This dataset contains 22,872 image pairs in total. We show some example images in Figure 1.

Although the flows are represented in the same form as classical optical flows [27], we would like to emphasize their geometric essence. Indeed, they only reflect the location and pose change of the foreground objects. In sharp contrast, the conventional optical flows blend information from a mixture of sources including object motion, camera motion, background, and even lighting conditions. We conjecture that the high precision and purity of this signal makes the learning relatively easier than from the 3D movies to be exploited afterwards.

It is also worth noting that using the simple FlyingChair dataset is just an easy starting point; more sophisticated synthetic imagery may be explored in the future. Recent years have witnessed emerging large-scale 3D synthetic datasets, e.g., ShapeNet [5] which contains millions of 3D models from thousands of categories. The intriguing observation is that, even on this surprisingly simple 3D dataset of FlyingChair, very positive results have been obtained. We therefore envision there will be greater gains from larger scale 3D datasets.

3.2. Geometry from real-world 3D movies

Even though the FlyingChairs has accurate flow fields, the variability of this dataset is limited. Besides, the artifacts due to the syntheses may cause a dramatic domain discrepancy between this dataset and real images or videos. It has been reported that the models trained with synthesis data perform relatively poor on the real image and video data [29]. In order to close the domain gap, we propose to leverage another type of geometry cue embedded in 3D movies. In 3D movies, there are generally two views at each time stamp that enhance the illusion of depth perceptions. Such video frames are usually stored in a stereoscopic format. For each frame, the format includes two projections of the same scene, one of which is exposed to the viewer's left eye and the other to the viewer's right eye.

We observe that the 3D movies contain rich geometry information that can be well utilized for learning generic video features. Particularly, we design a task of predicting the disparity map between the left and right views of the same frame. The disparity map mainly captures the depth of the scene. The challenge of this task is the lack of the groundtruth disparity maps. We propose to use the computational EpicFlow approach [28] to obtain the pseudo groundtruth disparity maps. We keep only horizontal disparities since in 3D movie the changes between the left and right views are horizontal. In our experiments, we crawl about 80 3D movies from the Web and sample about 40K video frames. Figure 1 shows some example video frames and the correspondingly estimated disparity maps.

However, it remains challenging to choose a correct geometry task and CNN architecture for using the geometry information contained in the 3D movies. We have explored several possible tasks and network architectures. For example, we first experimented to directly regress the depth value from a single view video frame using CNNs. However, the CNNs fail to learn effective video features that are transferrable to semantic understanding tasks. Finally, we explore a task of estimating the disparity maps by taking both views as input and find it work well.

3.3. Learning CNNs guided by the geometry cues

We are now ready to describe our approach to training deep CNNs from scratch using the geometry cues. After that, we fine-tune the network on massively benchmarked video recognition datasets to solve the high-level human activity recognition problem.

Network architecture. To train a deep CNN model, we design it in the following way. The network takes as input an image pair and is desired to predict the flow fields or disparity map between the two images. For the FlyingChairs dataset, the network predicts both horizontal and vertical flows, and for 3D movies only horizontal flows are considered. In this paper, we use the FlowNet Simple architecture [9] as our base CNN network. We stack the input each pair of images and then feed them together through the network to regress the flows or disparities.

Progressive training. One of the distinctive properties of our self-supervised learning is that we employ a progressive training strategy, as opposed to blindly pooling the data into one training set. We first input the synthetic images to the network to train a network that has the basic knowledge about the flow fields. After that, we use the 3D movies to further train the network in order to distill video representations that are closer to the real videos'. The challenge here is how to incorporate the video domain knowledge but not forget the original geometry knowledge about the FlyingChairs. To solve this problem, we borrow the cost function of the knowledge distillation network [14] and the learning without forgetting network [21]:

$$\arg\min_{\theta_s,\theta_o,\theta_n} \quad L_{new}(Y_n,\hat{Y_n}) + L_{old}(Y_o,\hat{Y_o}), \qquad (1)$$

where the parameters $\theta = \{\theta_s, \theta_o, \theta_n\}$ are the weights of the CNN (coded by different colors in Figure 2). The weights θ_s are shared (i.e., convolutional layers) by different sequential tasks. θ_o are the old task specific parameters (i.e., learned for predicting the flow fields of FlyingChairs), and θ_n are the new task specific parameters (i.e., learning for predicting the disparity maps for 3D movie). Before fine-tuning, we first feed the frames of 3D movies into the network and record the response $\hat{Y_o}$ by the old task specific weights. The loss $L_{old}(Y_o,\hat{Y_o})$ is a cross-entropy, which enforces the output Y_o for each pair of 3D movie frames to be close to the recorded output $\hat{Y_o}$ from the old network. The other loss $L_{new}(Y_n,\hat{Y_n})$ is to control the quality of the disparity map estimation. The key merit of this cost function is that it defines a regularization using the old task; it constrains the output for the old task by the updated network to be close to the original network's output. Therefore, the network can effectively carry along the knowledge learned from the FlyingChairs dataset to the related and yet different 3D movies.

4. Experiments

In this section, we evaluate the quality of our geometry guided self-supervised learning approach on two fundamental video understanding tasks: dynamic scene recognition and action recognition. We first apply the learned CNN as an off-the-shelf feature extractor and report the results on the video dynamic scene recognition task. Better results indicate better qualities of the learned video representations. Secondly, we devise a benchmark task that reflects real-world constraints to yield useful conclusions. Prior work on self-supervised learning uses the learned neural networks as the initialization for a fine-tuning stage for a particular task, such as object detection [7, 37], scene classification [26], and video recognition [24, 37]. The intuition is that good representations should be able to serve as a warm starting point for the task-specific fine-tuning. In this work, we mainly investigate the fine-tuning of the learned feature representations for the video action recognition task and leave other specific tasks for future work. Arguably, the action recognition is a hallmark problem in video understanding, so it can serve as a general task as object recognition in image understanding.

4.1. Geometry guided pre-training of CNNs

To use the geometry cues, we proposed to pre-train the CNN of the FlowNet [9] architecture, a variant of CaffeNet [18], by changing the number of channels of the first convolutional layer from 3 to 6 and replacing the classification layer by refining layers. The refining layers consist of four up-convolutional layers and an endpoint error loss layer. The endpoint error loss layer computes the squared Euclidean distances between the predicted flow values and the groundtruth averaged over all pixels.

When using the FlyingChairs dataset for optical flow estimation, we augment the training data using multi-scale cropping, horizontal flipping, translation, and rotation following [9]. We implement these using the Caffe toolbox. We set the learning rate as 10^{-4}, reduced to 10^{-5} after 120 epochs, and reduced to 10^{-6} after 160 epochs. The training converges after 200 epochs. After that, we feed the 3D movies data into the network and further fine-tune the network. For the training with the 3D movie data, we set the learning rate as 5×10^{-4}, and reduce it to 5×10^{-5} after 60K iteration. The training converges after 80K iterations.

Table 1. Comparisons with other shallow video feature representations and state-of-the-art self supervised approaches for dynamic scene recognition on Maryland and YUPENN.

Method	Maryland Acc (%)	YUPENN Acc (%)
Spacetime [10]	67.7	86.0
Orientation [6]	43.1	80.7
Object Patch [37]	48.46	70.47
Seq Ver. [24]	51.53	76.67
Geometry (Ours)	**69.23**	**86.90**

4.2. Dynamic Scene Recognition

For the dynamic scene recognition, we evaluate our learned video representations on two benchmarks: YUPENN [6] and Maryland [30], which contain 420 videos of 14 scene categories and 130 videos of 13 scene categories, respectively. We follow the standard leave-one-out evaluation protocol for comparisons.

We compare against both the shallow feature representations [10, 6] and CNN based self-supervised feature leaning approaches [37, 24]. For the results of shallow representations, we directly quote the numbers reported by the original paper for fair comparisons. As for the self-supervised feature representation learning approaches, we first extract all the video frame and then perform a feed-forward pass into the CNNs to extract the last convolutional layer as feature representation. And then we use an average pooling followed by L2 normalization to arrive at the video-level feature representations. Finally we use linear SVM to conduct the same leave-one-out evaluation protocol as described by the authors of these datasets.

The comparison results are shown in Table 1. We can observe that our geometry guided CNNs consistently outperform both shallow feature representations and state-of-the-art self-supervised approaches. Particulary, our approach achieves 17.7% better on the Maryland dataset and 10.2% better on the YUPENN dataset than the second best self-supervised representation learning approaches [24]. These results verify our claim that the pure pixel-level geometry signals can serve as a strong supervision for learning generic video feature representation for semantic tasks.

4.3. Action Recognition

In this section, we examine the generalization abilities of the learned network for action recognition task by domain-specific fine-tuning.

4.3.1 Datasets

We conduct the video recognition experiments on two publicly available action recognition datasets, namely UCF101 [32] and HMDB51 [19]. UCF101 is a large video dataset for action recognition collected from YouTube. It consists of 101 action classes, 13K clips, and 27 hours of video data in total. The task is generally considered challenging since many videos are captured under poor lighting, with cluttered background, or severe camera motion. The HMDB51 dataset is a large collection of realistic videos captured from various sources, such as movies and Web videos. This dataset contains 6,766 video clips from 51 action classes. We use the averaged classification accuracy as the evaluation metric. To be noted, all the results reported in this paper are on UCF101 and HMDB51 training/test splits.

4.3.2 Fine-tuning for action recognition

Once we finish the pre-training of the geometry guided CNN, we keep its convolutional layers and weights fixed and replace the refining layers with classification layers (i.e., a fully-connected layer coupled with a cross-entropy loss). After that, we fine-tune this model and test it for action recognition.

We uniformly sample 25 frames per video in the UCF101 and HMDB51 dataset as suggested in [24]. We then feed them into the pre-trained CNN model. All frames are randomly shuffled and organized as mini-batches with the size of 200. For the experiments on UCF101 dataet, the learning rate starts from 10^{-2} and decreases to 10^{-3} after 6K iterations, and then further to 10^{-4} after 12K iterations. We terminate the training after 15K iterations. For the experiments on HMDB51, the learning rate starts from 10^{-2} and decreases to 10^{-3} after 3.5K iterations, and then to 10^{-4} after 4K iterations. The training is stopped after 5K iterations. To prevent over-fitting to the training sets, we also use dropout with the rate of 0.1 after the fully connected layer. During the final inference stage, we take two consecutive video frames as input and output one softmax confidence score. The video-level score is obtained by average fusion over the 25 frames sampled from the video to be classified.

4.4. Discussions

Does CNN learn useful knowledge for video recognition from geometry? In this section, we first examine whether the geometry guided CNN learns useful knowledge for the seemingly distant task — semantic video recognition. Specifically, we evaluate our approach by transferring the feature representations learned in the self-supervised

Table 2. Comparisons with other state-of-the-art self supervised approaches for action recognition on UCF101 and HMDB51.

Method	UCF101 Acc (%)	HMDB51 Acc (%)
DrLim [13]	38.4	13.4
TempCoh [25]	45.4	15.9
Object Patch [37]	42.7	15.6
Seq Ver. [24]	50.9	19.8
OPN [20]	**56.3**	22.1
Geometry (Ours)	54.1	**22.6**
Bidireational Geometry (Ours)	55.1	**23.3**

Table 3. Comparisons with different initialization approach for action recognition on UCF101 and HMDB51.

Initializations	UCF101 Acc (%)	HMDB Acc (%)
Random	38.4	13.4
FlyingChairs	50.2	19.1
3D Movies	50.1	18.9
Geometry (Ours)	**54.1**	**22.6**

Table 4. Comparisons with other alternative combination approaches on UCF101.

Method	UCF101 Acc (%))
Early ensemble	52.4
Late ensemble	52.6
Fine-tuning	50.0
Reverse	52.9
Geometry (Ours)	**54.1**

Table 5. Fusion results with ImageNet pre-trained CNN model on UCF101 and HMDB51.

Method	UCF101	HMDB
geometry (Ours)	**54.1**	**22.6**
ImageNet	63.3	28.5
ImageNet + geometry (Ours)	**66.1**	**30.7**

manner to the video recognition tasks with labeled data. We compare, in Table 3, the recognition results on UCF101 and HMDB51 obtained by fine-tuning the same CNN architecture but with different initialization of the weights.

From this table, we have two interesting observations: 1) the feature representations learned from solving the geometry tasks are significantly better than those of training from scratch. Even using only 20K synthesis FlyingChairs images, we can also achieve 11.8% performance gain over the randomly initialized network on UCF101 dataset. 2) By adding real 3D movie data, we further obtain an additional 3.9% performance gain. These results imply our models can take advantage of the pixel-level geometry information (e.g. optical flow and disparity map) for action recognition, which is hard to solve if we only use the given training data in UCF101 or HMDB51. In short, the geometry cues help solve the semantic video recognition problem. We also visualize the class-specific discriminative regions [39] for action recognition in Figure 3.

Is the progressive training necessary? We investigate the efficacy of the progressive training strategy by comparing it to a few alternatives:

- Early fusion: we mix the FlyingChairs and 3D movie data together and then train a single network.
- Late fusion: we train two models respectively using FlyingChairs and 3D movies. At the test stage, we average the classification scores of the two networks to generate the action labels.
- Fine-tuning: we directly use the 3D movies to fine-tune the network without the knowledge distillation term in learning without forgetting.
- Reverse: We reverse the training order, namely, we first use 3D movies to pre-train the network, followed by using Flyingchairs to progressively update the model.

The results are reported in Table 4. It is clear that, from the table, when our geometry guided CNN is trained with the progressive strategy, it outperforms all the other alternative training schemes. Besides, we also draw two key observations. First, the progressive training, which is equipped with the learning without forgetting regularization, is very effective in capturing the two distinct geometry cues, compared with the naive early fusion and late fusion. Second, it is vital to order the synthesized FlyingChairs dataset, which is with accurate geometry groundtruth, before the 3D movies whose geometry information is relatively coarser. We expect these findings will benefit future research on the related subjects.

Comparison with state-of-the-art self-supervised approaches. In this section, we compare our results with other state-of-the-art self-supervised methods. Particularly, we compare with DrLim [13], TempoCoh [25], Object patch [37], Seq. Ver [24] and using the RGB images. We quote the numbers directly from the published papers. As shown in Table 2, our geometry guided CNN can achieve significantly better results than the other self-supervised approaches. We improve the results by almost 3.2% absolutely on UCF 101 and 2.8% on HMDB51, compared to the results [24]. Our results also comparable to the recently published approach [20], which improves the results

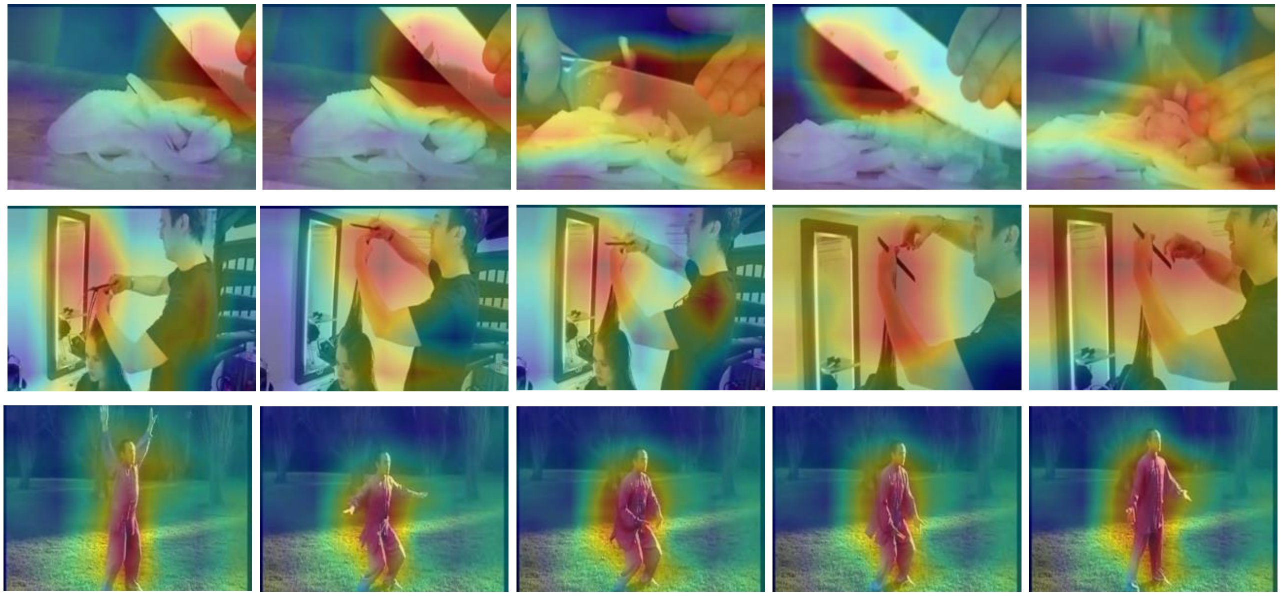

Figure 3. Visualization of class knowledge inside Geometry guided CNN model by using discriminative localization [39]. We select *cutting in kitchen*, *blow dry hair* and *taichi* (from top to bottom) for the visualization.

reported in [24] by sorting a tuple of frames from videos. These results indicate the dense pixel-level geometry information is very useful for semantic video recognition. We also observe that an ensemble of networks, which are trained by changing the temporal distance between the pair of frames, can bring additional 1% performance gain. To be noted, we achieve this result by using 10 times less training data during the pre-training task. We believe better results can be achieved when we render more synthesis pairs of images, e.g., using the large-scale 3D ShapeNet models or collecting more 3D videos.

Does geometry cues complement the visual knowledge base ImageNet? We have seen that the unsupervised pre-training using geometry data gives significant boost over training from scratch or using other labeling-free signals, but it still has gap compared with the CNN model pre-trained on ImageNet. It remains unclear whether it can improve existing ImageNet supervised feature representations. To answer this question, we conduct late fusion of the classification scores of the two types of networks. From Table 5, we can see that the late fusion leads to 2.8% performance gain on UCF101 dataset and 2.2% gain on HMDB51 dataset over the single ImageNet pre-trained model. This implies that the features learned with geometry are complementary with the features learned on the massively supervised ImageNet. By analyzing the action classes, we find that 48 classes out of the 101 classes of UCF101 and 26 out of the 51 classes of HMDB51 benefit from the fusion. Particularly, we find that the action classes like *hug, push up, jump Rope, cricket shot, long jump* have been improved significantly more than the other classes.

5. Conclusion

In this paper, we present a simple but effective geometry guided Convolutional Neural Network for the self-supervised video representation learning. To achieve this goal, we leverage two type of free geometry data: optical flow from synthesis image and disparity map from real 3D movies. These cues effectively drive the CNNs to extract generic knowledge from the conventional videos that is useful for the high-level semantic video understanding task. Learned features can directly used for video dynamic scene recognition, as well as action recognition after further domain-specific fine-tuned. Experimental results on four public available video semantic understanding datasets confirm that the effectiveness of our framework. We hope this paper will open up avenues for exploitation of geometry data in various computer vision tasks.

Acknowledgement The authors wish to thank the support of a Toyota Research Institute award, NSF grants DMS-1521608 and CCF-1514305, and a Samsung GRO award. Dr. Boqing Gong was supported by NSF awards IIS-1741431 and IIS-1566511 and gifts from Adobe and NVIDIA.

References

[1] P. Agrawal, J. Carreira, and J. Malik. Learning to see by moving. In *ICCV*, pages 37–45, 2015. 1, 3

[2] A. Bansal, X. Chen, B. Russell, A. G. Ramanan, et al. Pixelnet: Representation of the pixels, by the pixels, and for the pixels. *ICCV*, 2017. 3

[3] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. *Journal of the American Podiatry Association*, 60(60):6, 2009. 2

[4] Y. Bian, C. Gan, X. Liu, F. Li, X. Long, Y. Li, H. Qi, J. Zhou, S. Wen, and Y. Lin. Revisiting the effectiveness of off-the-shelf temporal modeling approaches for large-scale video classification. *arXiv preprint arXiv:1708.03805*, 2017. 1

[5] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 4

[6] K. G. Derpanis, M. Lecce, K. Daniilidis, and R. P. Wildes. Dynamic scene understanding: The role of orientation features in space and time in scene classification. In *CVPR*, pages 1306–1313, 2012. 6

[7] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015. 1, 2, 3, 5

[8] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CVPR*, 2015. 2

[9] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *CVPR*, pages 2758–2766, 2015. 2, 3, 5

[10] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Spacetime forests with complementary features for dynamic scene recognition. In *BMVC*, 2013. 6

[11] B. Fernando, H. Bilen, E. Gavves, and S. Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, pages 5729–5738, 2017. 3

[12] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *CVPR*, pages 2568–2577, 2015. 1, 2

[13] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, pages 1735–1742, 2006. 7

[14] G. E. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *CoRR*, 1503.02531, 2015. 5

[15] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1

[16] D. Jayaraman and K. Grauman. Learning image representations tied to ego-motion. pages 1413–1421, 2015. 1, 3

[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 1, 2

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 5

[19] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, pages 2556–2563, 2011. 6

[20] H.-Y. Lee, J.-B. Huang, M. Singh, and M.-H. Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, pages 667–676, 2017. 3, 7

[21] Z. Li and D. Hoiem. Learning without forgetting. In *ECCV*, pages 614–629, 2016. 2, 5

[22] X. Long, C. Gan, G. de Melo, X. Liu, Y. Li, F. Li, and S. Wen. Multimodal keyless attention fusion for video classification. AAAI, 2018. 1

[23] X. Long, C. Gan, G. de Melo, J. Wu, X. Liu, and S. Wen. Attention clusters: Purely attention based local feature integration for video classification. *CVPR*, 2018. 1

[24] I. Misra, C. L. Zitnick, and M. Hebert. *Shuffle and Learn: Unsupervised Learning Using Temporal Order Verification*. ECCV, 2016. 1, 3, 5, 6, 7, 8

[25] H. Mobahi, R. Collobert, and J. Weston. Deep learning from temporal coherence in video. In *ICML*, pages 737–744, 2009. 7

[26] A. Owens, J. Wu, J. H. Mcdermott, W. T. Freeman, and A. Torralba. *Ambient Sound Provides Supervision for Visual Learning*. 2016. 1, 2, 3, 5

[27] D. Pathak, R. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *CVPR*, 2017. 1, 3, 4

[28] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Epicflow: Edge-preserving interpolation of correspondences for optical flow. pages 1164–1172, 2015. 4

[29] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*, pages 213–226, 2010. 4

[30] N. Shroff, P. Turaga, and R. Chellappa. Moving vistas: Exploiting motion for describing scenes. In *CVPR*, pages 1911–1918, 2010. 6

[31] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1, 2

[32] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 6

[33] N. Srivastava, E. Mansimov, and R. Salakhutdinov. Unsupervised learning of video representations using lstms. *ICML*, 2015. 1, 2

[34] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. C3D: Generic features for video analysis. *ICCV*, 2015. 1, 2

[35] L. Wang, W. Li, W. Li, and L. Van Gool. Appearance-and-relation networks for video classification. *CVPR*, 2018. 1

[36] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 2

[37] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. In *ICCV*, pages 2794–2802, 2015. 1, 2, 3, 5, 6, 7

[38] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *CVPR*, volume 1, page 6, 2017. 3

[39] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 7, 8