Transfer Learning: Apply Image Recognition Models to Action Recognition with Dynamic Vision Sensor

First Author
Jiawei Liu
1753070@tongji.edu.cn

Second Author
Xin Wen
1751918@tongji.edu.cn

Abstract

Deep learning based methods have achieved great success in image recognition domain. Action recognition is a higher-level task that is hard to obtain satisfactory performance easily. The state-of-art models in image recognition domain are well designed with years of researchers' efforts, and have great ability in learning. On the one hand, action recognition models are either slow(2 stream based methods) or imprecise(3D convolotion based methods). On the other hand, image recognition methods have the limit that can only 'eat' 3-frame input.

To solve the problems, we stacked single frame data of dynamic vision sensor(DVS) which can capture motion data in an amazing speed, to 3-frame inputs which can be fed to image recognition models.

In our paper, we find that image recognition models trained with DVS data can do excellent work on action recognition domain and obtain 100% accuracy in UCF-50 DVS dataset[8](limited by computing resources, we only used 3 categories of them.). And we also find that freezing parameters of former layers series of transfer learning strategy is terrible when running into DVS data. Surprisingly, DVS data is very friendly with tiny models, tending to have better performance with them.

1. Introduction

Deep learning based image recognition algorithms is now achieving great success in computer vision domain. Recent years, hundreds of image recognition architectures(e.g. ResNet[7], GoogLeNet[20]) and datasets(e.g. ImageNet[3], CIFAR-10/100[12]) were designed to push the recognition ability of machines to the human level. On the other side, action recognition is a higher level mission that more and more researchers are now trying to design new models to boost the accuracy and speed in this domain. Unlike image recognition, action recognition is done via telling the category of a labeled video rather than an image. So the input of action recognition is a sequence of images, which means the related models need to deal with more information, which is not only the spatial component, but the temporal components as well. All these factors result in the unsatisfactory results in action recognition fields. A naive ResNet[7] model designed in 2015 is easy to achieve a 3.57% error rate in ImageNet test dataset. Although 2-stream ConvNet[17] extracts temporal component carefully, it still gets over 40% error rate with HMDB-51[13] dataset in 2014.

With all these evidences, we can see that state-of-art models in image recognition seems to have better performance in feature extraction and training as this field has a longer history than action recognition. Applying well-pretrained state-of-art models in image recognition to action recognition may be a great idea. However these 2 tasks have different input scale. For image recognition, 3-channel images is all the input. While in the action recognition field, the input scale can be various but never 3, as the input of it is a sequence of images whose amount of images is greater than 1. Moreover, sometimes we need to extract the temporal information using multi-frame dense optical flow, which means more frames. Hence, it seems to be impossible to set both the spatial and temporal components as the input of image recognition models.

As more and more algorithms are developed in action recognition field, there are also new hardware devices designed to produce high quality data which can be used as a new kind of input for the network architectures. Conventional cameras produce massive amounts of redundant data and are limited in temporal resolution by the frame rate. A conventional RGB camera produces a RGB image with large blocks of pixels in a low frame rate. Such kind of RGB frames contains a lot of redundant spatial data, but less temporal information. Apart from the conventional cameras itself, single RGB image show almost no temporal information. Extracting temporal information from RGB image sequences is of huge cost. Using the state-of-art dense optical flow algorithm[2], the computation time per frame can be 1.2 seconds to 23.4 seconds. Dynamic Vision

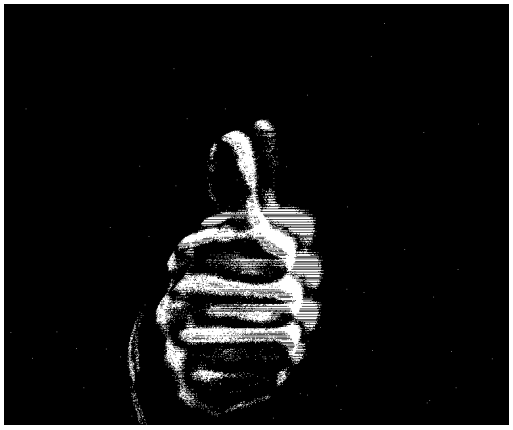

Figure 1. A visualized image extracted from DVS binaries. This is an image showing that Jiawei Liu is holding his thumb up to praise Xin Wen.

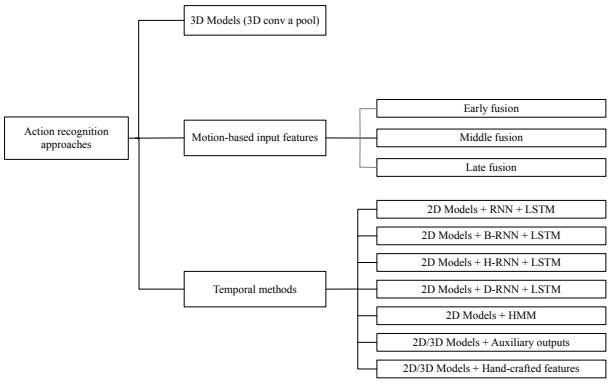

Figure 2. Main action recognition approaches.

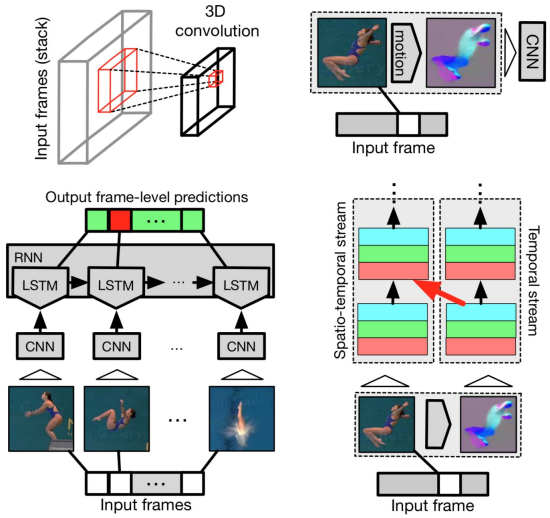

Figure 3. [1]The different architectures and fusion strategies. Top-left: 3D convolution. Top-right: motion pre-computation. Bottom-left: sequential modeling via LSTM. Bottom-right: fusion into a spatio-temporal stream.

Sensor(DVS)[14] is a event-driven vision sensor and can be used to solve such problems. Firstly, DVS is only interested in motion, which means it is able to tell which part of current frame is moving and which part of it is static. This kind of function directly produces motion information in a single frame. Secondly, it has many advantages in performance with 3.6μ latency which is much more faster than optical flow algorithms.

Unlike conventional RGB cameras which produce various binaries meaning different colors. DVS produce binaries to describe motion as figure 1 shows.

2. Related Work

Most deep learning based action recognition network architectures are designed to deal with the temporal or motion dimension component. There are 3 kinds of architectures mainly: 1. 3D-CNN models[10], 2. Multi-stream 2D-CNN models[17], 3. CNN with sequence models. As is shown in figure 2. Commonly, 3D-CNN models tend to be faster, while multi-stream 2D-CNN models tend to obtain higher accuracy.

3D-CNN models use fixed image sequences as 3D input and do 3D convolutional with 3D filters to extract motion and spatio-temporal features. Multi-stream 2D-CNN models use not only spatio-temporal sequence(input for one main stream), but also use other motion-based features(e.g. dense optical flow) as the input of other streams. Multi-stream models has another issue with the fusion of multi streams either in inputting or outputting. There are mainly 3 variants of fusion strategies: early fusion(making fusion before the data is fed into the models), middle fusion(making fusion at some intermediate layers) and late fusion(making fusions for the outputs of each stream). The fusion methods are mainly some naive ones like SVM, multi-layer perceptrons, average and so on. For the CNN with sequence models, they often set a sequence of images as input and then use 2D or 3D CNN to extract features as the input of sequence models. The sequence models are mainly RNN[5] based models. To deal with the RNN's drawback of short-term memory, Long Short-Term Memory(LSTM)[6] are more commonly used to be one unit in the RNN model to release short-term memory problems. There are also many new RNNs(e.g. Bidirectional RNN (BRNN)[16], Hierarchical RNN (H-RNN)[4], and Differential RNN (D-RNN)[22]) designed to boost the performance of naive RNN.

Figure 3 shows how the main architectures work when dealing with a sequence of input images.

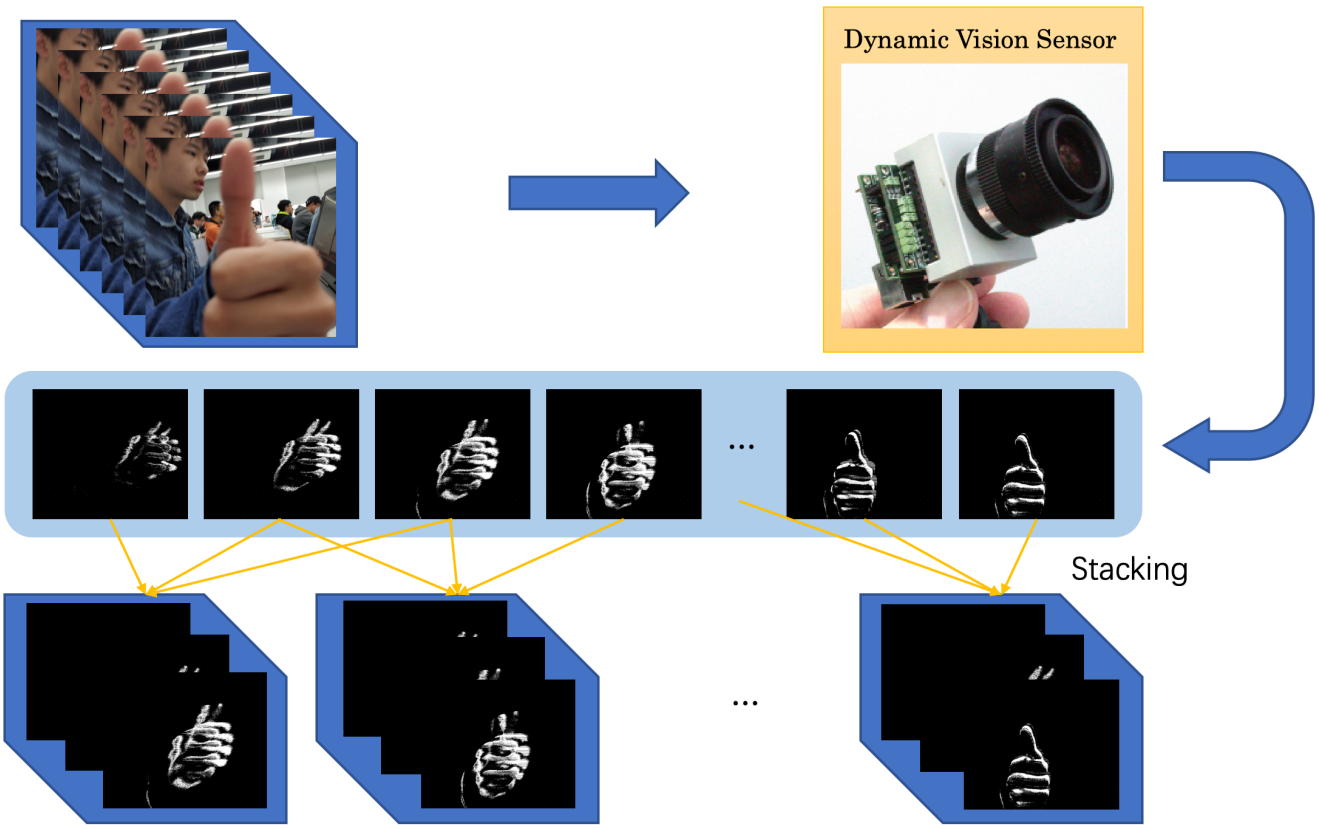

Figure 4. The procedure of frame stacking

3. Image Recognition Models with DVS Data

3.1. Frame Stacking Strategy

To let the image recognition models be able to take motion and spatio-temporal data in, we designed a frame stacking strategy as figure 4 shows.

Firstly we transform the DVS binaries into single frame images whose size is 640 by 768. This kind of image contains only 2 values: high(light pixels in figure 1) and low(dark pixels in figure 1). Then, we stack continuous 3 frames into a 3-frame image, which make it able to be feed with traditional 3-frame-input image recognition models.

3.2. Transfer Learning Strategy

After stacking the input frames into several 3-frame blocks. We are going to train the pre-trained models using the 3-frame blocks.

Traditional transfer learning approaches commonly lock the weights of former layers, and update the weights of last several layers. This kind of methods greatly accelerated the speed of training, as there won't be too much backpropagation as most of the weights are locked. However in our task, it may fail easily, as traditional approaches use images with channels meaning RGB. And our image channels do not mean the RGB components, but motion and spatio-temporal information. Hence, in our case, we believe locking the weights of former layers will not be a good strategy and it may easily run into bad results.

To prove our perspective that frozen parameters based transfer learning strategy is weak, we designed experiments to see whether weight locking is good for such transfer learning task. We will compare their accuracy and convergence rate in the 2 cases using a small dataset extracted from UCF-50 DVS dataset.

Also, to prove that DVS based data is not only fast, but also good for boosting the accuracy, we will cover the comparison part among different input features.

4. Experimental Results

4.1. Implementation Details

We extracted 3 categories of datasets from UCF-50 DVS dataset:

- Playing guitar
- Diving
- Basketball

Hyper Parameter List	
Item	Value
Learning rate	1×10^{-3}
Optimizer	Adamax[11]
Weight decay	1×10^{-3}
Batch size	32
Epochs	3

Table 1. Table of hyper parameters when training the pre-trained models.

Methods	Accuracy
DVS	**97.7%∼100%**
IDT	91.2%
DT	84.5%
HOG	76.1%
HOF	64.3%

Table 2. Comparison of different input features. The experimetal data except "DVS" comes from [15]. DVS performance was tested by partial samples of UCF-50 DVS dataset, while others were tested by all samples of UCF-50 dataset[19].

After stacking the frames as shown in figure 4, we got 4031 samples for training and 1728 samples for validation(training : validation $\approx$ 7:3). We feed the pre-trained models these data. When training, we trace the loss value(Cross entropy loss) of models with frozen parameters and those don't. As to the frozen strategy, we made the output linear layer of the models to 3 and froze all parameters except the that of the output layer.

Our hyper parameters in our experiment is shown in table 1. We used the same hyper parameters when training.

4.2. Results

With the same hyper parameters, the training results of each state-of-art method is shown in table 3.

From table 3 we can see that:

- The models without frozen parameters tend to have higher performance than the ones with frozen parameters.
- The loss of unfrozen models are difficult to converge along the epochs.
- The experimental accuracy of the trained pre-trained models indicates a high usability in industry.

The loss value of each model series over 378 iterations(3 epochs), are shown in figure 5, 6, 7 and 8.

As to the comparison of different input features, we can see the results in table 2. We can see that:

- DVS data tends to help the models to convergence quickly while obtain a high level of accuracy.

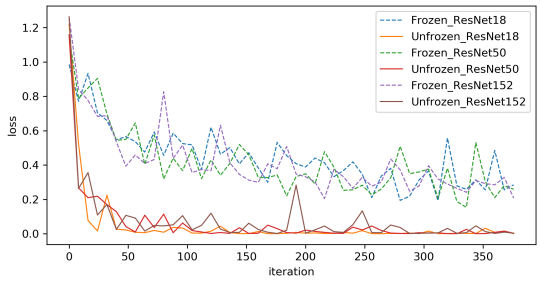

Figure 5. The loss value of ResNet models over 378 iterations

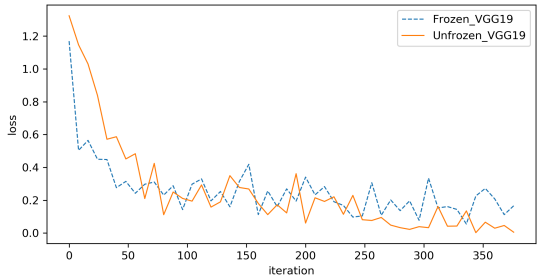

Figure 6. The loss value of VGG models over 378 iterations

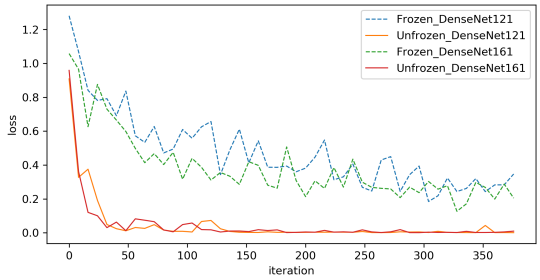

Figure 7. The loss value of densenet models over 378 iterations

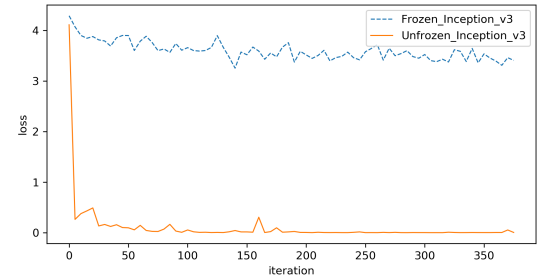

Figure 8. The loss value of inception v3 model over 378 iterations

- Apart from the high accuracy, we need to mention that

Experimental List		
Methods	Accuracy with frozen parameters	Accuracy without frozen parameters
ResNet-18[7]	92.2%	**100.0%**
ResNet-50[7]	94.0%	98.9%
ResNet-152[7]	92.7%	99.5%
VGG-19[18]	**94.3%**	97.7%
DenseNet-121[9]	91.7%	99.9%
DenseNet-161[9]	94.0%	99.8%
Inception v3[21]	90.3%	99.6%

Table 3. Table of each model's accuracy rate on our validation set.

the also-high-performance method – IDT, needs a lot of preprocessing time, which is hard to do real-time job. However, DVS data captured by DVS is amazingly fast and memory-saving(the former one needs hundreds milliseconds with the preprocessing while the frame rate of DVS is always on the million level.).

5. Conclusion

In this paper, we have following conclusions:

- Single-frame DVS data can be stacked to 3-frame and be fed to 3-frame image recognition models.
- Image recognition models with DVS data can do great jobs in the action recognition field.
- Freezing parameters series of transfer learning is not suitable when we want to train with DVS data, as most pre-trained models are good at extracting RGB data not DVS data.
- DVS data is very simple and therefore friendly with tiny models, tending to have better performance with them.
- ResNet18 has the best performance in our experiment.

References

[1] M. Asadi-Aghbolaghi, A. Clapes, M. Bellantonio, H. J. Escalante, V. Ponce-López, X. Baró, I. Guyon, S. Kasaei, and S. Escalera. A survey on deep learning based approaches for action and gesture recognition in image sequences. In *2017 12th IEEE international conference on automatic face & gesture recognition (FG 2017)*, pages 476–483. IEEE, 2017.

[2] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In T. Pajdla and J. Matas, editors, *Computer Vision - ECCV 2004*, pages 25–36, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[4] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015.

[5] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[6] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning precise timing with lstm recurrent networks. *Journal of machine learning research*, 3(Aug):115–143, 2002.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Y. Hu, H. Liu, M. Pfeiffer, and T. Delbruck. Dvs benchmark datasets for object tracking, action recognition, and object recognition. *Frontiers in Neuroscience*, 10:405, 2016.

[9] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[10] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

[11] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[12] A. Krizhevsky. Convolutional deep belief networks on cifar-10.

[13] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb51: A large video database for human motion recognition. pages 2556–2563, 11 2011.

[14] J. A. Leñero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco. A 3.6μs latency asynchronous frame-free event-driven dynamic-vision-sensor. *IEEE Journal of Solid-State Circuits*, 46(6):1443–1455, 2011.

[15] J. Miao, X. Jia, R. Mathew, X. Xu, D. Taubman, and C. Qing. Efficient action recognition from compressed depth maps. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 16–20. IEEE, 2016.

[16] L. Pigou, A. Van Den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre. Beyond temporal pooling: Recurrence and temporal convolutions for gesture recognition in video. *International Journal of Computer Vision*, 126(2-4):430–439, 2018.

[17] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[19] 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.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[22] V. Veeriah, N. Zhuang, and G.-J. Qi. Differential recurrent neural networks for action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 4041–4049, 2015.