Sustained Self-Supervised Pretraining for Temporal Order Verification

Himanshu Buckchash(✉) and Balasubramanian Raman

Machine Vision Lab, Department of Computer Science and Engineering,
Indian Institute of Technology Roorkee, Roorkee, India
hbuckchash@cs.iitr.ac.in, balarfma@iitr.ac.in

Abstract. Self-Supervised Pretraining (SSP) has been shown to boost performance for video related tasks such as action recognition and pose estimation. It captures important spatiotemporal constraints which act as an implicit regularizer. This work seeks to leverage upon temporal derivatives and a novel sampling algorithm for sustained (long term) SSP. Main limitations of our baseline approach are – its inadequacy to capture sustained temporal information, weaker sampling algorithm, and the need for parameter tuning. This work analyzes the Temporal Order Verification (TOV) problem in detail, by incorporating multiple temporal derivatives for temporal information amplification and using a novel sampling algorithm that does not need manual parameter adjustment. The key idea is that image-only tuples contain less information and become virtually indiscriminating in case of cyclic events, this can be attenuated by fusing temporal derivatives with the image-only tuples. We explore a few simple yet powerful variants for TOV. One variant uses Motion History Images (MHI), others use optical flow. The proposed TOV algorithm has been compared with previous works along with validation on challenging benchmarks – HMDB51 and UCF101.

Keywords: Self-Supervised pretraining · Temporal order verification · Action recognition

1 Introduction

SSP leverages the colossal amount of unlabeled data to provide an initial weight configuration which noticeably improves the performance of a model during its supervised fine-tuning. Applications of SSP can be found in different domains such as Action Recognition (AR), Natural Language Processing (NLP), and so forth. SSP ensures that the weights are not domain-specific, they readily generalize on closely related domains as well. For example: SSP for action recognition does well for pose estimation [13]; similarly, a model pretrained for question answering does well for commonsense reasoning, textual entailment, semantic similarity [16], use of pretrained word embeddings for multiple NLP tasks [2,14,15].

B. Deka et al. (Eds.): PReMI 2019, LNCS 11941, pp. 140–149, 2019.
https://doi.org/10.1007/978-3-030-34869-4_16

SSP for action recognition can be posed as a Temporal Order Verification (TOV) task [13]. TOV requires an unsupervised algorithm to generate a tuple of frames such that few of them are in valid temporal order (positive tuple) and few are out of order (negative tuple). These tuples are used to train a deep learning model that uses a binary log loss that helps to learn the pose information while trying to assert whether the order of the tuple is valid or not. As an example, Fig. 1 shows a positive tuple sampled for cartwheel action. In this figure, swapping the second and third frame will result in a negative tuple.

Fig. 1. A tuple sampled (at frame number – 5, 16, 23) for the cartwheel action.

In a recent study by researchers at OpenAI, it is shown that SSP boosts the performance of supervised tasks, and the learning is transferable to multiple related domains [16]. Similarly, it is shown by Wang *et al.* that SSP boosts performance for supervised action recognition and pose estimation [13,21]. These works provide convincing results for pairing supervised learning with SSP. This work builds upon the TOV work done by Misra *et al.* [13], and explores the challenges of SSP of deep models in the context of action recognition.

Psychologically, it is proved that the spatiotemporal signals provide significant information for answering questions based on the temporal ordering of spatial data [3,17]. Information can be sampled from spatiotemporal data, retaining the temporal order, and can be utilized for reasoning about the pose or trajectory of some object. This idea has been utilized by previous researchers [7,9,13,18,21]. The main emphasis is on sampling data in order of temporal constraints and using a discriminative model to learn the distribution of spatiotemporal information by auto-generating the positive and negative labels for data. A recurrent theme is to use the frames sampled at appropriate time-steps and using them for training a neural network, which can then infer about the sequence or ranking of the frames [13,21]. This approach can be further leveraged for solving action recognition or pose estimation problem [5,8,10,12,18,22,23].

The concept of tuple order verification has been recently applied for learning the context in videos and images [6,13,20,21]. Doersch *et al.* used the context of images for learning parts and object categories. They model the SSP problem as teaching a classifier about the relative placement of object patches in an image. This pretrained representation is then used to discover several categories of objects without any supervision [4]. However, their work cannot be directly applied to videos. Misra *et al.* have adapted their idea from images to videos [13]. They model action recognition task as – learning to order the temporal information. They sample frames from high motion instances in a video and

then train a triplet CNN with shared weights in a Siamese fashion to learn the order of the sampled frames (Fig. 1).

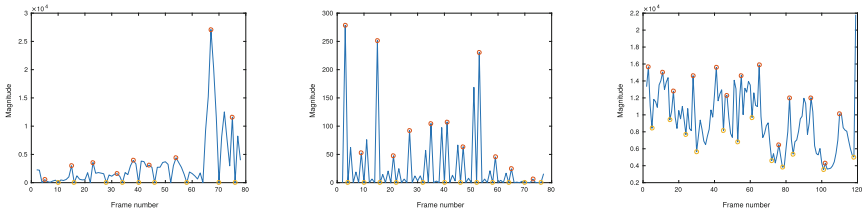
The main contribution of [13] is that unlike [21] they do not consider insertion of random samples for constructing positive and negative tuples, instead they sample frames from high motion window. This appears logically correct. However, they use image-only tuples, coupled with their sampling algorithm which does not capture *valleys* in the flow (as discussed under Sect. 2), which leaves room for degeneracy in performance. This work targets these two issues. First, we present temporal derivative fusion with image-only tuples for tighter temporal constraints, second, we provide a novel sampling algorithm that takes into account both the *valleys* (regions with low optical flow magnitude) and the *peaks* (regions with high optical flow magnitude). The sampling algorithm does not require manual parameter adjustment.

After performing an in-depth study on temporal derivatives and tuple sampling, the main contributions of this work are (1) Algorithm for fusion of multiple temporal derivatives with image-only tuples for persistent temporal dependencies. (2) A novel sampling algorithm that considers both *peaks* and *valleys* in optical flow. We found that considering both – peaks and valleys – improves results for cyclic events such as dribbling, clapping *etc.* (3) Proposed approach is made independent of manual parameter tuning, this increases the generalizability of our work. In addition to these, proposed work has been empirically and qualitatively validated on the two challenging action recognition datasets – HMDB51 [11], UCF101 [19].

Fig. 2. Example of tuples in case of sampling with and without valleys.

2 Analysis of Tuple Order Verification

It was observed by [13] that if the temporal windows are very far apart then there is a high probability of repetition of the same pose, especially in case of cyclic events. For cyclic actions such as clap, dribble, pullup, situp *etc.*, there



(a) Cartwheel (b) Clap (c) Dribble

Fig. 3. Optical flow signatures calculated with Frobenius norm for different actions.

exists a pause and movement cycle, where the action starts and finishes and then repeats itself. Unlike [13], instead of sampling from high motion windows (peaks), we found that it is also useful to consider the low motion windows (valleys). These motion windows are observed by taking Frobenius norm of optical flow. Frobenius norm for cartwheel action can be seen in Fig. 3, the corresponding frames are shown in Fig. 1. It can be seen in Fig. 2 first row, sampling frames from a dribbling action clip results similar information in consecutive frames, however, considering valleys during sampling helps alleviate the problem. Some events do not give enough time to capture valleys, such as chewing. Discrimination of the training tuples for these events is very confusing even for humans.

2.1 Proposed Method

Misra and Wang [13,21] observed that the triplet network performs as a better constraint on the latent representation of data by avoiding convergence of two points (in latent space) on to a single point. It was also noticed that taking up more than three frames does not provide any performance boost. Hence, a triplet Siamese model has been considered in this work (Fig. 4).

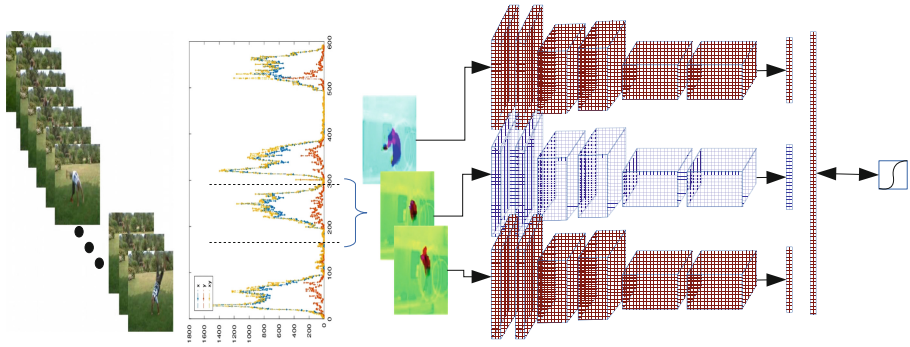


Fig. 4. Model of the proposed SSP technique showing how the Siamese-triplet network is trained on an action clip.

Algorithm 1. Self-supervised pretraining

Input: Dataset $\mathcal{D}$, Siamese model $\mathcal{M}$
Output: Set of pretrained weights $\mathcal{W}$
$\mathcal{P}, \mathcal{N} \leftarrow SampleTuple(\mathcal{V}) \mid \forall$ video $\mathcal{V} \in \mathcal{D}$;
Split $\mathcal{P}, \mathcal{N}$ as S_{train}, S_{val}, S_{test} sets ; `// 70:10:20`
`// Train using` S_{train}
foreach *epoch* **do**
  foreach *mini_batch* **do**
    $\mathcal{Y} \leftarrow \mathcal{M}(batch\_size, TempoDeriv(a, b, c))$; `// TempoDeriv() appends the specific temporal derivative of tuple` $-(a, b, c)$
    $\mathcal{L} \leftarrow -\frac{1}{N} \sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$;
    $Backprop(\mathcal{M}, Mean(\mathcal{L}))$;
  end
end

Temporal derivatives, such as Motion History Images (MHI) or Optical Flow (OF), alone provide much discriminatory information, this was especially observed on UCF101 by [5]. The entire SSP algorithm by [13] depends upon the choice of three frames (a tuple). Each frame can be seen as a window of information; it follows that we have three windows. Choosing the right frame at each of the three windows is crucial for temporal inference. A single frame resembles a very narrow sized temporal window. This fact coupled with the cyclic nature of events and arbitrary motion spikes, makes this small window very vulnerable, and a slight miss in sampling can make room for an invalid training tuple. The key idea in this work is to make the information persist (a little longer) at each of these three temporal windows. A larger temporal window serves as a tighter temporal constraint by capturing more information at each sampling step. Following this idea, we have fused the temporal derivatives with static pose information for constructing a sustained SSP algorithm.

Unlike [13] the proposed SSP approach samples the sets of positive and negative tuples from both peaks and valleys as described by Algorithm 2. We first sample quintuplets, and for every quintuplet (a, b, c, d, e), — (b, c, d), (d, c, b) are considered positive and (b, a, d), (b, e, d), (d, a, b), (d, e, b) are considered negative. These tuples are then fed to the weight-shared Siamese network, whose outputs from the fully connected layers of individual CNNs are concatenated prior to training with binary cross-entropy loss (Algorithm 1).

2.2 Sampling

The positive/negative tuple sampling is performed according to the Algorithm 2. First we find the peaks and valleys from the flow magnitude. Next, the peaks are clustered, and the ones falling in the lowest cluster are removed. Only one peak is kept in a radius of size Minimum Peak Distance (MPD). This results in a sequence of local peaks and valleys. Subsequently, sum of squared distance (SSD) is used for pruning consecutive similar frames, using a threshold σ, which

Algorithm 2. Tuple sampling

Input: Video $\mathcal{V}$
Output: Set of tuples, positive $\mathcal{P}$ and negative $\mathcal{N}$
$\mathcal{F} \leftarrow f_1, f_2, \ldots, f_n$ frames from $\mathcal{V}$ with stride of 2;
$\mathcal{F}_{of} \leftarrow OpticalFlow(\mathcal{F})$;
$\mathcal{F}_{peak} \leftarrow Peaks(\mathcal{F}_{of})$ with MPD $\leftarrow 4$;
$\mathcal{F}_{valley} \leftarrow Valleys(\mathcal{F}_{of})$;
Cluster $\mathcal{F}_{peak}$ with k-means, drop lowest magnitude cluster;
$i \leftarrow 1$; `// Index of first peak`
$k \leftarrow 2$; `// Index of second peak`
while $k \leq End(\mathcal{F}_{peak})$ **do**
  $\mathcal{S} \leftarrow j \mid \forall\, j \in (\mathcal{F}_{valley}) \wedge \mathcal{F}_{peak}(i) \prec \mathcal{F}_{valley}(j) \prec \mathcal{F}_{peak}(k)$;
  Append $CentralPeak(S)$ to $\mathcal{F}_{valley-new}$;
  $i \leftarrow k$;
  $k \leftarrow k + 1$;
end
$\mathcal{F}_{pv} \leftarrow \mathcal{F}_{peak} + \mathcal{F}_{valley-new}$; `// Combine`
$a \leftarrow 1$; `// Index of first frame in` $\mathcal{F}_{pv}$
$b \leftarrow 2$; `// Index of second frame in` $\mathcal{F}_{pv}$
while $b \leq End(\mathcal{F}_{pv})$ **do**
  For consecutive frame pair $(a, b) \in \mathcal{F}_{pv}$;
  if $S(a, b) < \sigma$ **then**
    $b \leftarrow b + 1$; `// Sum of squared distance less than threshold` σ
  else
    $a \leftarrow b$; $b \leftarrow b + 1$;
  end
end
foreach *quintuplet* $-(a, b, c, d, e)$ *of consecutive frames* a, b, c, d, e *from* $\mathcal{F}_{pv}$ **do**
  $\mathcal{P} \leftarrow (b, c, d), (d, c, b)$;
  $\mathcal{N} \leftarrow (b, a, d), (b, e, d), (d, a, b), (d, e, b)$;
end

is determined empirically. Following this, peaks and valleys are combined and quintuplets are formed. For every quintuplet (a, b, c, d, e), — (b, c, d), (d, c, b) are considered positive and (b, a, d), (b, e, d), (d, a, b), (d, e, b) are considered negative.

3 Experiments

All experiments have been performed on split 1 of UCF101 [19] and HMDB51 [11]. [13] is used as a baseline for our experiments. Due to the high computational requirement of Siamese networks, a non-bulky CaffeNet like architecture is considered for the proof of concept. It is trained from scratch for all experiments with variable learning rate, dropout, momentum, and a batch size of 64. 650K tuples were sampled for UCF101, 350K tuples were sampled for HMDB51. We experimented with three variants of temporal derivatives. tRGB uses only RGB tuples, tBW uses only gray-scale image tuples, tOF-BW early-fuses gray-scale

Table 1. Results of tuple prediction on HMDB51 and UCF101 datasets.

Dataset	tRGB	tBW	tOF-BW	tOF-RGB	tMHI-RGB	[13]	[21]
UCF101	69.1	67.3	71.2	**74.8**	28.6	67.3	61.2
HMDB51	35.6	33.6	35.4	**38.2**	15.9	34.1	30.8

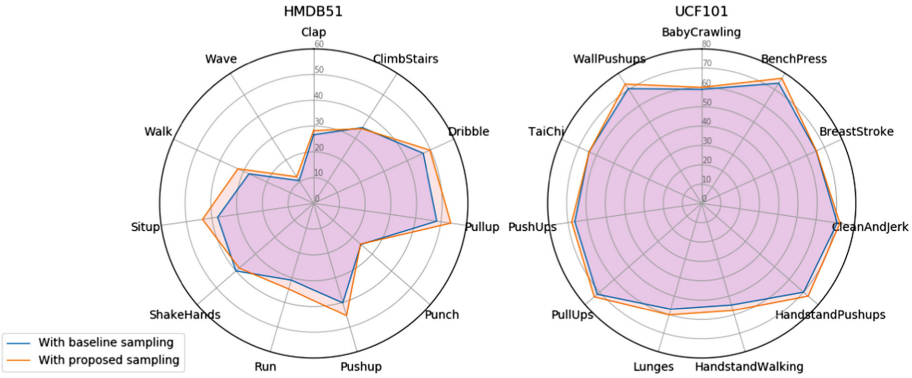


Fig. 5. Comparison of the proposed sampling algorithm with the baseline sampling on two challenging datasets – HMDB51 & UCF101.

frames with the flow derivatives, tOF-RGB early-fuses RGB frames with the flow, tMHI-RGB uses Motion History Images (MHI) with short temporal window [1]. All of these variants are tested with the proposed sampling algorithm.

3.1 Results Analysis

The reversed order of the tuples is also deemed as valid since it preserves temporal constraints. It was also observed that forward and backward tuples are naturally discovered during the sampling of cyclic actions. Semantic level fusion of flow captures details complementary to the spatial data. Tuple verification by Misra *et al.* captures pose, however this variant of tuple verification forces the model to learn motion transformation (Fig. 7). It can be inferred from the Table 1 that fusion of temporal derivatives (tOF-RGB, tOF-BW), significantly boosts the overall performance of SSP. When combined with spatial data, the flow acts as a saliency by preserving the motion information about the parts (Fig. 7). The best results are obtained by tOF-RGB model which outperforms

Table 2. Comparison of sampling methods for TOV.

Dataset	[13]	tRGB
UCF101	67.3	**69.1**
HMDB51	34.1	**35.6**

Fig. 6. Results of retrieved nearest neighbors against four query images. Top row results by tOF-RGB model, bottom row for a model initialized with random weights.

[13] by over 7% and [21] by over 13%, as reported in the Table 1. tMHI-RGB performs miserably because it tends to clutter the image with a lot of information which reduces meaningful pose information. tBW and tRGB have small performance difference; it suggests that the model learns pose and is invariant to color. tOF-BW model not only learns pose information but also learns the flow transformation parameters. Table 2 shows the difference between the baseline sampling algorithm [13] and the proposed sampling for RGB tuples. For further clarity on the performance of the tuple sampling algorithm, class-wise comparison of cyclic actions is reported in Fig. 5. It can be seen that for both of the datasets, the proposed sampling algorithm performs better than the baseline sampling.

Figure 6 shows the frames retrieved for a nearest-neighbor query on the tOF-RGB model, in comparison to the baseline model which is initialized with random weights. We see that the tOF-RGB model captures the pose information better than the model with random weights. This establishes the significance of SSP.

(a) Boxing

(b) Cartwheel

Fig. 7. Triplets having fusion of flow and spatial data for creation of positive tuples.

4 Conclusion

The whole purpose of having TOV is to be able to order or rank the data. The ability to rank or order the temporal data is in proportion to the ability to discriminate. To achieve this, each element of tuple-to-be-ordered should store information which is as discriminatory as possible. It was noticed that the spatial information could be augmented with temporal derivatives for each input to the Siamese-triplet. To attain this, multiple approaches were explored in this work. tOF-RGB achieved best results using proposed sampling along with flow and RGB information fusion. It could be concluded that the temporal derivatives provide a better representation for estimation of the pose. In the future, focus can be on the elongation of the temporal windows. To achieve this, 3D convolutions can be explored. The main takeaways of this work are: (1) Temporal derivatives are a strong prior for ordering (2) The combination of flow and spatial information is better than each considered individually, as we see that it acts as a salient pair (3) For sampling, both peaks and valleys should be considered for capturing the cyclic actions.

References

1. Ahad, M.A.R.: Motion History Images for Action Recognition and Understanding. Springer (2012)
2. Chen, D., Manning, C.: A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 740–750 (2014)
3. Cleeremans, A., McClelland, J.L.: Learning the structure of event sequences. J. Exp. Psychol. Gen. **120**(3), 235 (1991)
4. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1422–1430 (2015)
5. Feichtenhofer, C., Pinz, A., Zisserman, A.: Convolutional two-stream network fusion for video action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1933–1941 (2016)
6. Goroshin, R., Bruna, J., Tompson, J., Eigen, D., LeCun, Y.: Unsupervised learning of spatiotemporally coherent metrics. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4086–4093 (2015)
7. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: Null, pp. 1735–1742. IEEE (2006)
8. Jain, M., Jegou, H., Bouthemy, P.: Better exploiting motion for better action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2555–2562 (2013)
9. Jayaraman, D., Grauman, K.: Learning image representations equivariant to ego-motion. In: Proceedings of ICCV (2015)
10. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
11. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: a large video database for human motion recognition. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2556–2563. IEEE (2011)

12. Le, Q.V.: Building high-level features using large scale unsupervised learning. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8595–8598. IEEE (2013)
13. Misra, I., Zitnick, C.L., Hebert, M.: Shuffle and learn: unsupervised learning using temporal order verification. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 527–544. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_32
14. Pennington, J., Socher, R., Manning, C.: Glove: global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
15. Qi, Y., Sachan, D.S., Felix, M., Padmanabhan, S.J., Neubig, G.: When and why are pre-trained word embeddings useful for neural machine translation? arXiv preprint arXiv:1804.06323 (2018)
16. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf (2018)
17. Reber, A.S.: Implicit learning and tacit knowledge. J. Exp. Psychol. Gen. **118**(3), 219 (1989)
18. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems, pp. 568–576 (2014)
19. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: a dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
20. Srivastava, N., Mansimov, E., Salakhudinov, R.: Unsupervised learning of video representations using LSTMS. In: International Conference on Machine Learning, pp. 843–852 (2015)
21. Wang, X., Gupta, A.: Unsupervised learning of visual representations using videos. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2794–2802 (2015)
22. Wang, X., Gupta, A.: Generative image modeling using style and structure adversarial networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 318–335. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_20
23. Yang, W., Gao, Y., Cao, L., Yang, M., Shi, Y.: mPadal: a joint local-and-global multi-view feature selection method for activity recognition. Appl. Intell. **41**(3), 776–790 (2014)