Temporal Dropout of Changes Approach to Convolutional Learning of Spatio-Temporal Features

Dubravko Culibrk
University of Trento
Italy
dubravko.culibrk@unitn.it

Nicu Sebe
University of Trento
Italy
sebe@disi.unitn.it

ABSTRACT

The paper addresses the problem of learning features that can account for temporal dynamics present in videos. Although deep convolutional learning methods revolutionized several areas of multimedia and computer vision, there have been relatively few proposals dealing with ways in which these methods can be enabled to make use of motion information, critical to the extraction of useful information from video. We propose a temporal dropout of changes approach for this, which allows us to consider temporal information over a series of frames without increasing the number of training parameters of the network.

To illustrate the potential of the proposed methodology, we focus on the problem of dynamic texture classification. Dynamic textures represent an important form of dynamics present in video data, so far not considered within the framework of deep learning.

Initial results presented in the paper show that the proposed approach, based on a well-known deep convolutional neural network, can achieve state-of-the-art performance on two well-known and challenging dynamic texture classification data sets (DynTex++ and UCLA dynamic texture).

Categories and Subject Descriptors

I.5.4 [**Pattern Recognition**]: Applications—*computer vision*

Keywords

Deep Learning; convolutional neural networks; spatio-temporal features; dynamic texture; temporal dropout

1. INTRODUCTION

The dominant methodology for visual recognition from images and video until recently relied on hand-crafted features [17][7]. Today, we are witnessing a paradigm shift and a growing interest in methods that learn features in both unsupervised and supervised settings.

Current research on deep learning suggests that there is significant potential in using large-scale Neural Networks (NNs) to address machine learning and, in particular, computer vision problems. The Google Brain project showed how an unsupervised AutoEncoder NN with 1 billion connections was able to learn to recognize common objects just by looking at a week's worth of YouTube videos [12]. In 2012, Krizhevsky *et al.* [11] showed how a Convolutional NN (CNN) with 650,000 neurons can be used to classify 1.2M images in the ImageNet Large Scale Visual Recognition data set into 1,000 classes [3], significantly advancing the state of the art. Their approach has recently been successfully extended to object detection achieving beyond-state-of-the-art results on the PASCAL VOC challenge data [7]. Deep learning has also seen several successful applications in the domain of image classification [23][21] and content-based retrieval [19].

When it comes to learning from video data, using deep (convolutional) NNs, few approaches exist [17][12]. However, arguably the most prominent, 3D CNN [9], achieved the best performance in three human action recognition tasks of the TRECVID 2009 Evaluation for Surveillance Event Detection challenge [16], showing the significant potential for such approaches, when large amount of labeled data is available.

Dynamic Textures (DT) represent a set of phenomena occurring in nature, where the perceived changes in the appearance of a system of large number elements are consistent, although the individual elements undergo stochastic changes in theirs. Typically the changes are due to motion (e.g. turbulent water, smoke, vegetation in the wind, insect swarms), but may be the result of the changing intensity of light emitted (e.g. fire). Zhao and Pietikäinen consider such phenomena extensions of the static texture to the temporal domain [22], since the effect is that of a textured object undergoing transformations. Derpanis and Wildes [4], however, point out that the term can apply equally well to simpler phenomena when analyzed in terms of aggregate regional properties (e.g., orderly pedestrian crowds and vehicular traffic).

The ability to recognize DTs based on visual processing is of significance to a number of applications, including, video indexing/retrieval, surveillance and environmental monitoring where they can serve as keys, isolate background clutter (e.g., fluttering vegetation) from activities of interest and detect various critical conditions (e.g., fires), respectively. It comes as no surprise that a significant amount of research effort has been directed toward solving this problem [2][22][13][4][5][20][6]. So far, to the best of our knowledge, no one has attempted to solve the problem of learning high-level features and recognizing DTs using deep NN architectures. Although, as a stochastic spatio-temporal phenomenon, DTs represent a basic domain for testing and evaluating spatio-temporal features.

In this paper we make several contributions: Dynamic texture classification using CNNs is considered for the first time. The proposed methodology exploits changes due to motion to extract

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM'14, November 3–7, 2014, Orlando, Florida, USA.

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-3063-3/14/11 ...$15.00.
http://dx.doi.org/10.1145/2647868.2655048.

dynamic-texture related temporal features and introduces a novel approach to taking into account sequences of frames, without increasing the number of training parameters. We evaluated the proposed methodology on two public, widely used and challenging data sets: DynTex++ [6] and UCLA [5] dynamic texture datasets. The results presented show that we can achieve state of the art classification results.

The rest of the paper is organized as follows: Section 2 deals with the relevant published work. Section 3 describes the proposed methodology. Section 4 discusses experiments performed and results achieved. Section 5 is dedicated to our conclusions.

2. RELATED WORK

2.1 Dynamic Texture Classification

The research into the classification and recognition of dynamic textures continues unabated [4][22][20][6]. A large number of approaches have been proposed over the last ten years. In their 2005 survey Chetverikov and Péteri [2] divided the existing approaches into five classes: methods based on optical flow, methods computing geometric properties in the spatio-temporal domain, methods based on local spatio-temporal filtering, methods using global spatio-temporal transforms and model-based methods that use estimated model parameters as features. Regardless of the type of the approach, they attempt to extract features descriptive of the dynamic texture and classify them by either defining a suitable distance measure and creating a simple distance-based algorithm for comparison or training a machine learning algorithm to achieve the task.

Volume local binary patterns (VLBP) have been proposed by Zhao and Pietikäinen as features to describe dynamic textures [22]. The VLBPs are an extension of the LBP operator widely used in ordinary texture analysis, that combine motion and appearance. They tested their approach using videos generated by extracting parts of the sequences in the DynTex database [13], creating a data set that had 10 examples of a certain class derived from single DynTex sequences. Their classifier is a simple nearest neighbor classifier, based on the log-likelihood statistic that allows them to compare VLBPs, and they used leave-one-group-out (i.e. n/m fold cross-validation [18]) to measure performance, where m corresponds to the number of examples extracted for a single dynamic texture and n is the total number of examples. Various classification rates were achieved depending of whether or not the features used were shift-invariant and how long the feature vector was. Their best result is an accurate classification rate of 95.71%, achieved for a shift-invariant VLBP and a fairly large feature vector $(4,176$ bins$)$.

Chan and Vasconcelos [1] model the dynamic texture as a linear dynamic system (LDS) and achieve good classification using the Martin distance to compare the models. They evaluated both nearest neighbor and support vector machine (SVM) classifiers and showed that the use of a machine learning algorithm such as SVM can improve the classification significantly. Through the use of the SVM classifier they achieved accurate classification rate of 97.5% on the UCLA database [15]. More recently (2009) their work has been extended by Ravichandran *et al.* [14] to use bags of LDSs to achieve improved view-invariant texture classification, when eight classes of textures are concerned.

Derpanis and Wildes [4] proposed new features based on spatio-temporal oriented energy filters to describe dynamic textures and classify them. They identified 7 semantic categories in the UCLA database (flames, fountain, smoke, turbulence, waves, waterfall, vegetation) and achieved a comparatively low classification rate of 92.3%, on sequences derived from this database. However, they specifically considered shift-invariant recognition, and report improved performance under these conditions.

Recently, an approach based on Dynamic Fractal Spectrum (DFS) of temporal gradient (changes) and intensity, derived from a sequence of DT images, has been proposed [20]. Once the features are extracted, an SVM classifier is used to perform DT classification. To the best of our knowledge, the DFS approach achieves state-of-the-art results on the UCLA and DynTex++ public databases, achieving 100% accuracy on the UCLA for 50-class problem and 89.9% for DynTex++.

As far as we are aware no one has considered the problem of using deep neural networks to learn high-level features and use them as basis for dynamic texture classification/recognition.

2.2 Spatio-Temporal Features for Deep Learning

Several approaches have been prosed that attempt to learn the spatio-temporal features using deep CNNs. The focus of these approaches has been mainly on human action recognition.

The approach proposed by Taylor *et al.* [17] can be viewed as a convolutional extension of the Gated Restricted Boltzmann Machine. The authors showed that the architecture proposed is able to learn correspondences between pairs of images. Based on such features they build a multistage neural network classifier that achieved performance comparable to the state of the art in the domain of human action recognition.

More recently, Le *et al.* [12] proposed another approach that uses unsupervised learning of basic features using independent subspace analysis. They stacked Independent Subspace Analysis networks as subunits to form the final convolutional network. Replacing handcrafted features with the learned ones yielded state-of-the-art performance on human action recognition. The training is done on 3D blocks of (10) (16×16 pixel) patches from consecutive frames fed as input to the net. They trained on 200,000 video blocks. It is interesting to note that the performance of their approach drops by 10% when only two frames were considered instead of 10.

The 3D CNN [9] is a supervised approach. In contrast with the two previously discussed approaches, the input to the network are not just raw frames. Instead, the input is comprised of greyscale frames, the gradient and optical flow in x and y directions. Buffers of 7 frames centered on the current frame are used as input to 3D convolution kernels. The authors state that blocks of 5-7 frames have empirically been shown to yield the same performance as taking into account the whole sequence. The size of the patches used for the TRECVID data set was 60×40.

In the proposed approach we also consider the grayscale raw information of the frames, the temporal gradient instead of optical flow, but represent the block of frames of interest with a single randomly selected frame. In contrast to other approaches, our approach does not suffer from the increase in the number of training parameters with the expansion of the block (time-interval) considered.

3. TEMPORAL DROPOUT OF CHANGES APPROACH

The proposed approach, shown in Fig. 1 is simple. Similar to the DFS [20] we use greyscale images and temporal gradient (change) information. However, rather than considering only the temporal gradient between the current and previous frame, we would like to consider longer-term motion across a larger block of frames (Group of Pictures - GOP), as typically done in other deep learning approaches [9]. To do so without increasing the number of train-

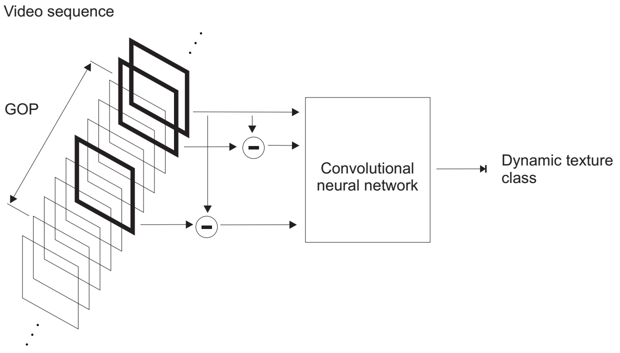

Figure 1: Proposed change detection and temporal dropout approach: The input to the convolutional network is comprised of the greyscale frame, temporal gradient and the gradient between the current frame and a frame randomly selected from a preceding block of frames - Group of Pictures (GOP).

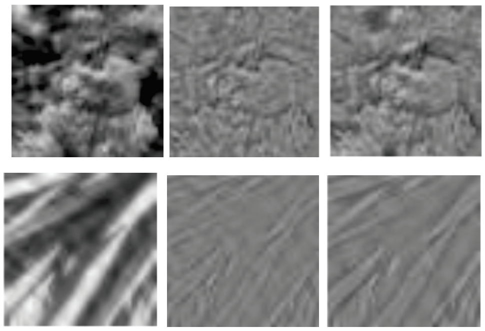

Figure 2: Features from sample frames: Left to right - greyscale, gradient previous frame, gradient random frame. Top - DynTex++ cloth class, bottom - UCLA plant-t-near class.

able parameters, which is a limiting factor in the other proposed appoaches, we use a temporal dropout strategy, whereby a single frame is selected at random from the GOP and a gradient is computed between that frame and the current one.

Our approach is motivated by the dropout mechanism used to reduce overfitting in deep NNs [8], by preventing co-adaptation. In our experiments we used GOP size of 25 frames, as it is represents a second of a 25 fps sequence. This enables us to handle 2.5 times larger sequence of frames than the largest previously considered, without increasing the size of the network.

Fig. 2 shows features extracted for sample frames from the DynTex++ and UCLA data sets. The gradients were scaled to the [0,255] range.

While the proposed approach can be used with any convolutional network architecture, the convolutional neural network used in our experiments is the Krizhevsky *et al.* 650,000 neurons architecture [11], as adapted in [7]. The network contains eight layers, five are convolutional and the remaining three are fully connected. The output of the network is a 128 class softmax, allowing us to use the same architecture to produce the distribution over the classes from both DynTex++ and UCLA datasets (total of 86). The rest is unused, but kept as it does not effect the computational efficiency. The input to the network was reduced by 2 pixels when compared to the original size of the sequences in the databases considered. This allows us to to follow the standard training procedure where

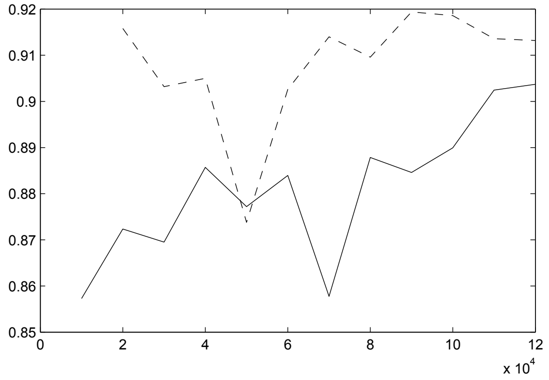

Figure 3: Accuracy during training over iterations: for UCLA data set - dashed line and DynTex++ data set - solid line.

10 patches are created for each input frame. The size of the input was therefore 48×48 pixels in the DynTex++ case and 46×46 in the UCLA. We train our network on a NVIDIA Tesla K40 GPU.

4. EXPERIMENTS AND RESULTS

Three main public DT datasets exist that have been widely used for DT analysis: the UCLA dataset [5], the DynTex dataset [13] and the DynTex++ dataset [6]. Since the DynTex++ dataset is derived from DynTex and seems to be the more challenging dataset of the two, we test the proposed approach on DynTex++ and UCLA datasets and compare with the state-of-the-art results for the DFS method reported in [20].

The UCLA dynamic texture dataset consists of 50 DT classes, each with four grayscale video sequences captured from different viewpoints. Each sequence contains 75, 48×48 pixel frames. In our experiments we used 25% of the data for testing, while the rest was used for training. The split corresponds to the first fold as provided with the dataset.

The DynTex++ dataset contains 36 classes of dynamic texture, each of which contains 100 sequences of a fixed size $50 \times 50 \times 50$. In our experiments we use a random selection of 20% of the sequences as a test set, while the rest form our training set.

We use a GOP of 25 frames and generate a single data instance for each frame after the initial 25 frames. Thus, each sequence of the DyntTex++ was represented with 25 instances per sequence and each UCLA sequence with 50. Each instance is classified separately by the NN and the final classification for a sequence is done by majority voting.

The initial training is done for 20,000 iterations on the larger DynTex++ data set. We than continue training the network either on DynTex++ or UCLA data for a subsequent 100,000 iterations. The training procedure is the same as that used in [7]. The training for our approach takes 24 hours. The classifier accuracy over the training iterations for both datasets is shown in Fig. 3. In all the experiments we used the Caffe implementation of convolutional neural networks [10].

We report two types of accuracy. The per-frame accuracy considers all the frames of the sequence separately and assumes that the decision about the type of the dynamic texture can be made based on a single frame. The sequence-level refers to the accuracy achieved by taking the mode of the per-frame labels as the label of the whole sequence.

The sequence-level accuracy our approach achieved on the UCLA dataset is 98% (a single sequence was misclassified). The top per-

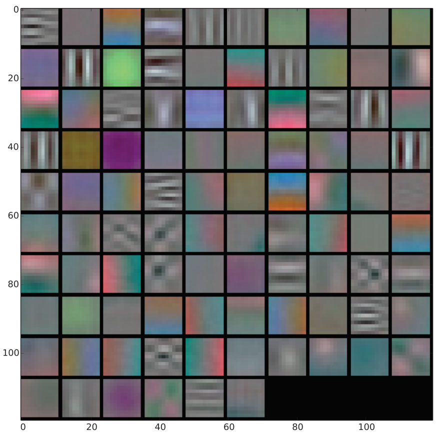

Figure 4: Some of the first convolutional layer filters learned by the network.

frame accuracy achieved over 100,000 iterations was 91.94%. The proposed method's sequence-level classification result is second only to DFS features combined with an SVM classifier. Both the per frame and sequence-level accuracy are above all other methods reported in the literature.

The top per-frame accuracy achieved for the DynTex++ data set is 90.37%, which, even though done at the level of single frames, surpasses the state of the art reported in [20] for sequence-level.

The network is clearly able to learn the features that enable efficient classification of the different DT classes. This can also be seen in Fig. 4, which shows some of the filters from the first layer of the network. The colored information in the figure corresponds to the temporal features learned, while the greyscale represents static features.

5. CONCLUSION

Although the deep learning methods revolutionizes several areas of computer vision, they have not, up to now, been considered for the task of dynamic texture classification. In this paper we show that a deep convolutional neural network can be efficiently used to achieve state-of-the-art DT classification results.

Our approach to learning features relies on temporal gradients and a novel temporal dropout technique, which allows us to consider larger time intervals without increasing the number of learning parameters.

6. ACKNOWLEDGMENTS

This project has received funding from the European Union's Seventh Framework Programme under xLiMe and QoSTREAM, Grant numbers 611346 and 295220.

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

7. REFERENCES

[1] A. Chan and N. Vasconcelos. Classifying video with kernel dynamic textures. In *CVPR*, pages 1–6. IEEE, 2007.

[2] D. Chetverikov and R. Péteri. A brief survey of dynamic texture description and recognition. *Computer Recognition Systems*, pages 17–26, 2005.

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

[4] K. Derpanis and R. Wildes. Dynamic texture recognition based on distributions of spacetime oriented structure. In *CVPR*, pages 191–198. IEEE, 2010.

[5] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.

[6] B. Ghanem and N. Ahuja. Maximum margin distance learning for dynamic texture recognition. In *ECCV*, pages 223–236. Springer, 2010.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013.

[8] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[9] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *PAMI*, 35(1):221–231, 2013.

[10] Y. Jia. Caffe: An open source convolutional architecture for fast feature embedding. `http://caffe.berkeleyvision.org/`, 2013.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, volume 1, pages 4–9, 2012.

[12] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, pages 3361–3368. IEEE, 2011.

[13] R. Péteri, S. Fazekas, and M. Huiskes. DynTex: A comprehensive database of Dynamic Textures. *Pattern Recognition Letters*, 31(12):1627–1632, 2010.

[14] A. Ravichandran, R. Chaudhry, and R. Vidal. View-invariant dynamic texture recognition using a bag of dynamical systems. In *CVPR*, pages 1651–1657. IEEE, 2009.

[15] P. Saisan, G. Doretto, Y. Wu, and S. Soatto. Dynamic texture recognition. In *CVPR*, pages II–58–II–63. IEEE, 2001.

[16] A. F. Smeaton, P. Over, and W. Kraaij. Evaluation campaigns and trecvid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330. ACM Press, 2006.

[17] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV*, pages 140–153. Springer, 2010.

[18] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques, 2nd Edition*. Morgan Kaufmann, San Francisco, 2005.

[19] P. Wu, S. Hoi, H. Xia, P. Zhao, D. Wan, and C. Miao. Online multimodal deep similarity learning with application to image retrieval. In *ACM MM*, pages 153–162, 2013.

[20] Y. Xu, Y. Quan, H. Ling, and H. Ji. Dynamic texture classification using dynamic fractal analysis. In *ICCV*, pages 1219–1226. IEEE, 2011.

[21] Z. Yuan, J. Sang, and C. Xu. Tag-aware image classification via nested deep belief nets. In *ICME*, pages 1–6, 2013.

[22] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *PAMI*, (28):915–928, 2007.

[23] S.-h. Zhong, Y. Liu, and Y. Liu. Bilinear deep learning for image classification. In *ACM MM*, pages 343–352, 2011.