Pillar Networks for action recognition

B Sengupta
Cortexica Vision Systems Limited
Imperial College London
London, UK
b.sengupta@imperial.ac.uk

Y Qian
Cortexica Vision Systems Limited
30 Stamford Street SE1 9LQ
London, UK
yu.qian@cortexica.com

***Abstract*—Image understanding using deep convolutional network has reached human-level performance, yet a closely related problem of video understanding especially, action recognition has not reached the requisite level of maturity. We combine multi-kernels based support-vector-machines (SVM) with a multi-stream deep convolutional neural network to achieve close to state-of-the-art performance on a 51-class activity recognition problem (HMDB-51 dataset); this specific dataset has proved to be particularly challenging for deep neural networks due to the heterogeneity in camera viewpoints, video quality, etc. The resulting architecture is named pillar networks as each (very) deep neural network acts as a pillar for the hierarchical classifiers.**

I. Introduction

Video understanding is a computer vision problem that has attracted the deep-learning community, notably via the usage of the two-stream convolutional network [12]. Such a framework uses a deep convolutional neural network (dCNN) to extract static RGB (Red-Green-Blue) features as well as motion cues from another network that deconstructs the optic-flow of a given video clip. Notably, there has been plenty of work in utilising different types of network architectures for factorising the RGB and optical-flow based features. For example, an inception network [15] uses 1×1 convolutions in its inception block to estimate cross-channel corrections, which is then followed by the estimation of cross-spatial and cross-channel correlations. A residual network (ResNet), on the other hand, learns residuals on the inputs [5].

There are obvious problems that have impeded high accuracy of deep neural networks for video classification. Videos unlike still images have short and long temporal correlations, attributes that single frame (image) convolutional neural network fail to discover. Therefore, the first hurdle is designing recurrent networks and feedforward networks that can learn this latent spatio-temporal structure. Nonetheless, there has been much progress in devising novel neural network architecture since the work of [7]. Another problem is the large storage and memory requirement for analysing moderately sized video snippets. One requires a relatively larger computing resource to train ultra deep neural networks that can learn the subtleties in temporal correlations, given varying lighting, camera angles, pose, etc. It is also difficult to utilise classical image augmentation techniques on a video stream. Additionally, video-based features (unlike in static images) evolve with a dynamics across several orders of time-scales. To add to this long list of technical difficulties, is the problem of the semantic gap, i.e., whether classification/labelling/captioning can lead to "understanding" the video snippet?

We improve upon existing technology by combining Inception networks and ResNets using a Support-Vector-Machine (SVM) classifier that is further combined in a multi-kernel setting to yield, to the best of our knowledge, an increased performance on the HMDB51 data-set [9]. Notably, our work makes the following contributions:

- We introduce pillar networks that are deep as well as wide (depending on use-case), enabling horizontal scalability
- Ability to classify video snippets that have heterogeneity regarding camera angle, video quality, pose, etc.

II. Methods

In this section, we describe the dataset, the network architectures and the multi-kernel learning based support-vector-machine (SVM) setup that we utilise in our four-stream dCNN pillar network for activity recognition. We refer the readers to the original network architectures in [18] and [10] for further technical details. While we do not report the results here, classification methodologies like AdaBoost, gradient boosting, random forests, etc. have classification accuracy in the range of 5-55% for this dataset, for either the RGB or the optic-flow based features.

A. Dataset

The HMDB51 dataset [9] is an action classification dataset that comprises of 6,766 video clips which have been divided into 51 action classes. Although a much larger UCF-sports dataset exists with 101 action classes [14], the HMDB51 has proven to be more challenging. This is because each video has been filmed using a variety of viewpoints, occlusions, camera motions, video quality, etc. anointing the challenges of video-based prediction problems. The second motivation behind using such a dataset lies in the fact that HMDB51 has storage and compute requirement that is fulfilled by a modern workstation with GPUs – alleviating deployment on expensive cloud-based compute resources.

All experiments were done on Intel Xeon E5-2687W 3 GHz 128 GB workstation with two 12GB nVIDIA TITAN Xp GPUs. As in the original evaluation scheme, we report accuracy as an average over the three training/testing splits.

BS and YQ contributed equally to this paper.

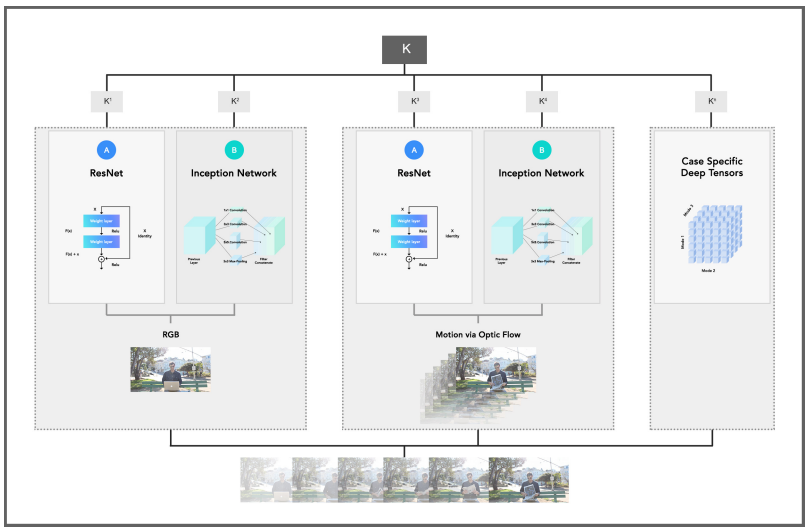

Fig. 1. **The Pillar Network framework:** In this specific instantiation, there are two types of networks, namely ResNets and Inception networks that factorise static (RGB) and dynamic (optic flow) inputs obtained from the video. Whilst we do not use additional case specific deep tensors for the present work, under such a multi-kernel learning framework, additional feature tensors (hand-crafted or otherwise) can be learnt, according to the specific need of the problem. K^i refers to the individual kernels for the input video that are subsequently combined to yield a single kernel K for the SVM.

B. Inception layers for RGB and flow extraction

We use the inception layer architecture described in [18]. Each video is divided into N segments, and a short sub-segment is randomly selected from each segment so that a preliminary prediction can be produced from each snippet. This is later combined to form a video-level prediction. An Inception with Batch Normalisation network [6] is utilised for both the spatial and the optic-flow stream. The feature size of each inception network is fixed at 1024. For further details on network pre-training, construction, etc. please refer to [18].

C. Residual layers for RGB and flow extraction

We utilise the network architecture proposed in [10] where the authors leverage recurrent networks and convolutions over temporally constructed feature matrices as shown in Fig. 1. In our instantiation, we truncate the network to yield 2048 features, which is different from [10] where these features feed into an LSTM (Long Short Term Memory) network. The spatial stream network takes in RGB images as input with a ResNet-101 [5] as a feature extractor; this ResNet-101 spatial-stream ConvNet has been pre-trained on the ImageNet dataset. The temporal stream stacks ten optical flow images using the pre-training protocol suggested in [18]. The feature size of each ResNet network is fixed at 2048. For further details on network pre-training, construction, etc. please refer to [10].

D. Support Vector Machine (SVM) with multi-kernel learning (MKL)

The basis of the second stage of our classification methodology rests on a maximum margin classifier – a support vector machine (SVM). Given training tuples (x_i, y_i) and weights w, under a Hinge loss, a SVM solves the primal problem [11],

$$\begin{aligned} \min_{w,b,\zeta} \frac{1}{2} w^T w \quad &+ \quad C \sum_{i=1}^{n} \zeta_i \\ s.t. \quad & y_i \left(w^T \phi\left(x_i\right) + b \right) \geqslant 1 - \zeta_i \\ & \zeta_i \geqslant 0, i = 1, \ldots, n \end{aligned} \tag{1}$$

As is customary in kernel methods, computations involving ϕ are handled using kernel functions $k\left(x_i, x_j\right) = \phi\left(x_i\right) \cdot \phi\left(x_j\right)$. In all of our experiments, a Radial Basis Function (RBF) based kernel has been used. C (fixed at 100) is the penalty parameter and ζ is the slack variable.

For multiple kernel learning (MKL), we follow the recipe by [13] (*cf.* [19]) and formulate a convex combination of sub-kernels as,

$$\kappa\left(x_i, x_j\right) = \sum_{k=1}^{K} \beta_k k_k\left(x_i, x_j\right) \tag{2}$$

where, $\beta_k \geqslant 0$ and $\sum_{k=1}^{K} \beta_k = 1$. As shown in [13], we then formulate Eqn. 2 as a semi-infinite linear optimization problem. The value of β is obtained using a linear programming (LP) solver.

III. Results

We use 3570 videos from HMDB51 for training the SVMs under a multiple kernel learning (MKL) framework. Utilising four networks yield four features tensors that are fused in steps, to form a single prediction (Figure 1). The feature tensors for both RGB and Flow are extracted from the output of the last connected layer with 1024 dimension for the

TABLE I
SVM ACCURACY RESULTS

	Inception Network			ResNet			Kernel Fusion
	optical flow	RGB	MKL	optical flow	RGB	MKL	
split-1	61%	54%	68.1%	58.5%	53.1%	63.3%	71.7%
split-2	62.4%	50.8%	69.2%	57.5%	48.6%	62.2%	72.5%
split-3	64%	49.2%	69.5%	57.2%	48%	62%	71.2%
Average	62.5%	51.3%	68.9%	57.7%	49.9%	62.5%	71.8%

Inception network and 2048 for the ResNet network. Four separate SVMs are trained on these feature tensors. Results have been shown for the two networks used – Inception (Table I). We then fuse multiple kernels learnt from the individual classifiers using a semi-infinite linear optimisation problem. Average result from three splits is displayed in Table I. It is apparent that combining kernels from various stages of the prediction process yields better accuracy. It is indeed possible to fuse hand-crafted features, such as iDT [17], to the features generated from a dCNN – although not reported in this work we anticipate features such as iDT will boost the accuracy of the pillar networks. Such additional features take the place of 'case-specific tensors' in Figure 1.

Table II compares our method to a few other methods in the literature. Of notable mention, are the TS-LSTM and the Temporal-Inception methods that form part of the framework that we use here. In short, synergistically, utilising multiple kernels boosts the performance of our classification framework, and enable state-of-the art performance on this dataset.

IV. DISCUSSION

Our main contribution in this paper is to introduce **pillar networks** that are deep as well as wide (by plugging in other deep networks, horizontally) enabling horizontal scalability. Combining different methodologies allow us to reach close to the current state-of-the-art in video classification especially, action recognition.

We utilised the HMDB-51 dataset instead of UCF101 as the former has proven to be difficult for deep networks due to the heterogeneity of image quality, camera angles, etc. As is well-known videos contain extensive long-range temporal structure; using different networks (2 ResNets and 2 Inception networks) to capture the subtleties of this temporal structure is an absolute requirement. Since each network implements a different non-linear transformation, one can utilise them to learn very deep features. Utilising the distributed architecture then enables us to parcellate the feature tensors into computable chunks (by being distributed) of input for an SVM-MKL classifier. Such an architectural choice, therefore, enables us to scale horizontally by plugging in a variety of networks *as per requirement*. While we have used this architecture for video based classification, there is a wide variety of problems where we can apply this methodology – from speech processing (with different pillars/networks) to natural-language-processing (NLP).

Our framework rests on two stages of training – one for training the neural networks and the other for training the multiple kernels of the support vector machine (SVM). Since both of the training stages are decoupled, it allows for scalability wherein different networks can operate on a plug-and-play basis. Indeed, there has been some work in combining deep neural networks with (linear) SVMs [16] to facilitate end-to-end training.

It would be useful to see how pillar networks perform on immensely large datasets such as the Youtube-8m data-set [1]. Additionally, recently published Kinetics human action video dataset from DeepMind [8] is equally attractive, as pre-training, the pillar networks on this dataset before fine-grained training on HMDB-51 will invariably increase the accuracy of the current network architecture.

REFERENCES

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. YouTube-8M: a large-scale video classification benchmark. 2016.

[2] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Spatiotemporal multiplier networks for video action recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[3] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[4] B. Fernando and S. Gould. Discriminatively learned hierarchical rank pooling networks. *arXiv preprint arXiv:1705.10420*, 2017.

[5] 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.

[6] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[8] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The Kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[9] H. Kuehne, H. Jhuang, R. Stiefelhagen, and T. Serre. HMDB51: a large video database for human motion recognition. In *High Performance Computing in Science and Engineering 12*, pages 571–582. Springer, 2013.

[10] C.-Y. Ma, M.-H. Chen, Z. Kira, and G. AlRegib. TS-LSTM and Temporal-Inception: Exploiting spatiotemporal dynamics for activity recognition. *arXiv preprint arXiv:1703.10667*, 2017.

[11] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[12] 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.

[13] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7(Jul):1531–1565, 2006.

[14] 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.

TABLE II

ACCURACY SCORES FOR THE HMDB51 DATA-SET. NOTICE THAT OUR METHOD IS THE CURRENT STATE-OF-THE-ART WERE WE TO IGNORE THE HAND-CRAFTED IDT FEATURES.

Methods	**Accuracy [%]**	**Reference**
Two-stream	59.4	[12]
Rank Pooling (ALL)+ HRP (CNN)	65	[4]
Convolutional Two-stream	65.4	[3]
Temporal-Inception	67.5	[10]
TS-LSTM	69	[10]
Temporal Segment Network (2/3/7 modalities)	68.5/69.4/71	[18]
ST-ResNet + hand-crafted iDT	70.3	[10]
ST-multiplier network	68.9	[2]
Pillar Networks + SVM-MKL	71.8	this paper
ST-multiplier network + hand-crafted iDT	72.2	[2]

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pages 2818–2826, 2016.

[16] Y. Tang. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*, 2013.

[17] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.

[18] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016.

[19] Z. Xu, R. Jin, I. King, and M. Lyu. An extended level method for efficient multiple kernel learning. In *Advances in neural information processing systems*, pages 1825–1832, 2009.