2D Deep Video Capsule Network with Temporal Shift for Action Recognition

Theo Voillemin, Hazem Wannous, Jean-Philippe Vandeborre

2D Deep Video Capsule Network with Temporal Shift for Action Recognition

Théo Voillemin
University of Lille
CNRS, IMT Lille Douai
UMR 9189 - CRIStAL
F-59000 Lille, France
Email: theo.voillemin@univ-lille.fr

Hazem Wannous
University of Lille
CNRS, Centrale Lille
IMT Lille Douai
UMR 9189 - CRIStAL
F-59000 Lille, France
Email: hazem.wannous@univ-lille.fr

Jean-Philippe Vandeborre
IMT Lille Douai
University of Lille, CNRS
UMR 9189 - CRIStAL
F-59000 Lille, France
Email: jean-philippe.vandeborre@imt-lille-douai.fr

***Abstract*—Action recognition in continuous video streams is a growing field since the past few years. Deep learning techniques and in particular Convolutional Neural Networks (CNNs) achieved good results in this topic. However, intrinsic CNNs limitations begin to cap the results since 2D CNN cannot capture temporal information and 3D CNN are to much resource demanding for real-time applications. Capsule Network, evolution of CNN, already proves its interesting benefits on small and low informational datasets like MNIST but yet its true potential has not emerged. In this paper we tackle the action recognition problem by proposing a new architecture combining Temporal Shift module over deep Capsule Network. Temporal Shift module permits us to insert temporal information over 2D Capsule Network with a zero computational cost to conserve the lightness of 2D capsules and their ability to connect spatial features. Our proposed approach outperforms or brings near state-of-the-art results on color and depth information on public datasets like First Person Hand Action and DHG 14/28 with a number of parameters 10 to 40 times less than existing approaches.**

I. Introduction

Action recognition and video understanding are hard problems that generally requires a lot of computational cost that make them difficult to process continuous videos. Though, a lot of applications require for efficient and accurate solution of those problems like hand gestures recognition for new Augmented/Virtual Reality interactions.

Deep learning has become one of the most efficient techniques for image and video understanding especially since the appearance of Convolutional Neural Networks (CNNs) [1]. They also proved their benefits for real-time applications, at least for relatively small networks, since the learning phase is particularly time consuming, but the inference consists only of a bunch of calculus. However, 3D CNNs, even if they present the benefit to capture both spatial and temporal information, are generally way to large to be applied in real-time with reasonable computing power [2], [3], [4]. Conversely, 2D CNNs, thanks to their weights sharing, bring small computational cost, perfect for fast training and real-time inference, but in their basic implementation, they cannot extract temporal information since they work with only one frame at a time [5], [6], [7], [8].

To counterbalance this lack of temporal information extraction, previous works presented several approaches taking into account temporal modeling into 2D CNNs [8], [9], [10]. Our attention especially focused on recent Temporal Shift Module (TSM) research by Li *et al.* [11] that provides new efficient video modeling by adding temporal information into 2D CNN by shifting channels in convolutional layers without deteriorating computational costs.

Besides this lack of temporal information, CNNs seize other limitations as invariance because of pooling, or the incapability to understand spatial features relationship between convolutional layers. A solution was proposed by Sabour *et al.* [12] developing a new architecture based on convolution, capsule network that have presented promising results on some datasets like MNIST [13] with a far less deeper network than the common CNN solutions. But still, Capsule Network struggles to provide good performance on complex dataset. Recent intuitions, similar to deep convolutional neural network, pushed some research as Rajasegaran *et al.* [14] to experiment deep capsule network (DCN) by stacking capsule layers. They found that simply stacking those layers conduct to a degradation of performance whether in classification precision or in computational cost because of the new routing algorithm, specific to the good learning of capsule layers. In order to address this issue, they proposed to reduce the number of routing iterations on the initial capsule layers, also adding skip connections to improve learning of middle layers and use of 3D convolutions layers to reduce parameters number. It should also be noted that the usage of capsule network have not been yet explored for video understanding, only Duarte *et al.*'s approach [15] can be found. However, we note in this first network of 3D capsules the same problem as for 3DCNN, that is the significant cost of calculation of these solutions. In our exploration of 2D capsule network solutions for video and action understanding, we found some new complications specific to capsule architecture, the weights divergence when large scale information are passed as input of capsule layer, because frame data from video dataset as HMDB51 [16] or FPAH [17] are much bigger than the 14 x 14 frame of dataset like MNIST [13].

To address those problems, we propose in this paper a new architecture combining deep capsule network and temporal shift module for efficient video and action understanding, called later **2D Deep Video Capsule Network** (**2D DVCN**). We explore three different capsule shifts based on Li *et al.*'s approach [11], illustrated in Fig. 1. Shift of entire first capsules from a frame to the next one (Fig. 1d), shift of first dimension features for every capsules of a layer of a frame to the capsules of the next one (Fig. 1c) and shift of first dimension features of the first capsule of a layer of a frame to the next one (Fig. 1b).

The implementation of temporal shift on capsule network is motivated by the strength of capsules to understand relationship between spatial features extracted by convolution operations. Our goal is to reinforce the temporal information shifted and shared from a frame to the next one by linking the shifted convolutions with the frame's original ones thanks to the capsules and routing algorithm. We also figured out how to avoid weights divergence when processing large image with capsule network by modifying the original DeepCaps network [14] permitting to use input frame 10 times larger than actual frames of datasets usually test on by other capsule networks. The main contributions of our paper are the followings:

- Developing a first 2D capsule network for video understanding (**2D DVCN**) thanks to the temporal information addition of the temporal shift module.
- Proposing three different implementations (Fig. 1) of temporal shift module over capsule layer and a study of their impact on classification rate of our approach.
- Evaluation the performance of our approach over challenging action and gesture video datasets as HMBD51 [16], FPAH [17] and DHG14/28 [18] with outperform or near state-of-the-art results over color and/or depth information but with 10 to 40 times less parameters than other 2D/3D CNN or RNN methods.

The rest of this paper is organized as follows: Discussion of related work over other capsule or 2D/3D convolutional network for video understanding problem in Section II, description of our 2D Deep Video Capsule Network and temporal shift module implementation in Section III, Section IV shows our results and Section V concludes this paper.

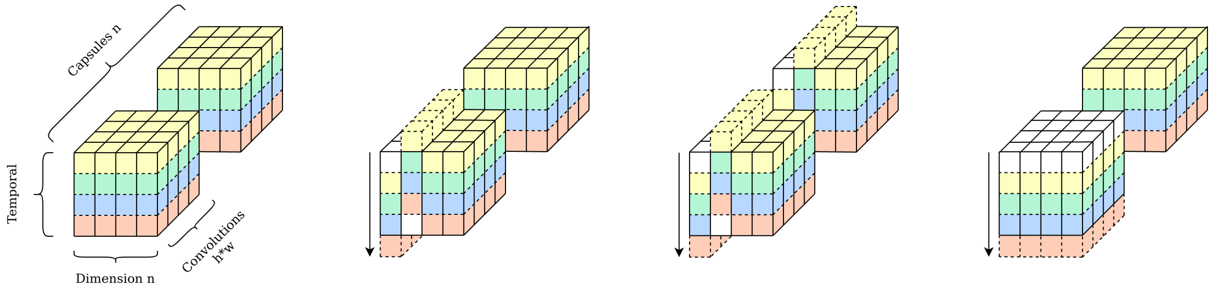
II. Related Work

Since the first implementation of **Capsule Network** by Sabour *et al.* [12], some new capsule architectures have been explored. Ma *et al.* [19] developed deep capsule network joints to LSTM for forecasting transportation. Similar idea for speech recognition but with an LSTM before capsule layer has been explored by Srivastava *et al.* [20]. We can particularly observe that there are few works on capsule network opposite to other deep methods, especially in video understanding. We can find only one implementation of Capsule Network for video understanding in the literature, Duarte *et al.* [15] developed the first 3D Capsule Network especially for action detection and segmentation but we can find the same defect than 3D CNN, the important computational cost compared to 2D Capsule Network.

2D CNN is the lightest method for video recognition and understanding ([5], [6], [7], [8]), but these methods have a natural lack to capture temporal information. Karpathy *et al.* [5] designed a two-streams CNN that takes the same image at different resolutions to capture more spatial information. Temporal Segment Network [7] combines a sparse temporal sampling strategy with video-level supervision. But if 2D CNNs can run efficiently on modest hardware, most of them cannot capture temporal information. 3D CNNs were first developed to offset this lack and most of them surpass 2D CNNs results but for an important computational loss [2], [3], [4]. Molchanov *et al.* [3] implemented a fusion of 3D CNN to capture spatio-temporal information with a Recurrent Neural Network (RNN) for online analysis of video clip per clip, but this method demanded an important computational power.

Recurrent Neural Networks have been also designed for temporal sequences analysis like speech recognition and action or gesture recognition [21], [22], [23], [24]. In their works, Du *et al.* [24] used skeleton joints of each body members on a hierarchical Bidirectional-RNN network to proceed action recognition. RNN based method have generally better classification results since the skeleton information over time is really informative. Still, we decided to develop our 2D DVCN on color, depth or optical flow images since this kind of information is really easy to obtain. It is thus present on a great majority of databases and do not require a extraction pretreatment as skeleton joints needs. Our motivation is therefore to have the more flexible solution.

Many attempts have been made to **model temporal information** into deep learning method. We already discussed above about 3D CNN and their natural handling of temporal information. Attention mechanism have been explored, Guo *et al.* [25] used dynamic weighted sum of local 2D CNN and 3D CNN representations to pass extracted features into an LSTM. Zang *et al.* [26] have implemented Attention-based Temporal Weighted to implant visual attention into a multi-stream CNN. Lin *et al.*[11] proposed the temporal shift module to deal with lack of temporal information in 2D CNN without computational loss, on each kernel of each convolutional layer, a small proportion of channels is shift on the next or previous frame of a same video sequence that will be treat frame per frame by the CNN. This temporal shifting permits to bring information of a part of the spatial extracted features of the previous frame on the current treated frame, or from the next to the current one in case of offline implementation. This new module, in addition to have zero extra computational cost apart of data movement, improved state-of-the-art performance over multiple action datasets like HMDB51 [16] or UCF101 [27] in comparison to other temporal modeling methods. The last work particularly attracted our attention and from which our proposed approach was inspired.



(a) Original convolution capsule layer

(b) Offline shift on first kernels of the first capsule

(c) Offline shift on first kernels of every capsules

(d) Online shift on every kernels of the first capsules

Fig. 1: Visual implementation of our temporal shift over capsule layers. A color horizontal slice of a big cube represent a capsule. A big cube represent the same capsule at different temporality / frame. The two big cubes represent a ShiftConvCapsule layer (first and last capsules). Small cube represent a convolutional kernel of size h*w, channels are represented over a line. A white small cube is for zero-padding since the shift leave empty the first channels. In offline implementation, (a) and (b), channels are shifted in both temporal direction, past and future, since it's impossible in an online solution to get information from the future, an online shift has been implemented to progressively shift only to the future frame.

III. 2D Deep Video Capsule Network (2D DVCN)

2D capsule network architecture for video understanding has not been yet explored. The only works explored to deal with temporal information using capsule used LSTM cells that can only perform well on low dimensional information which is not the case for videos treatment. However, capsule network already proves its potential for spatial-features extraction [12], [14], hence we decided to explore how can we incorporate temporal-features extraction into capsules. We first explain the Temporal Shift Module (TSM) [11] implementation into capsules, then our 2D Deep Video Capsule Network for video understanding.

A. *Shift Convolutional Capsule*

Let's represent the shape of the input of a convolutional capsule as : $R^{(w,h,c,n)}$ with w and h respectively be the width and height of the current spatial features, c the number of capsule tensors and n the dimension of the capsule (number of convolutional kernels). By looking at the shape of a convolutional capsule, we can see that, structurally, capsules are just a set of $c*n$ convolution kernels of size $w*h$. Benefits of capsule network occur by regrouping n convolution kernels into c capsules so that they can represent complex spatial features and with the routing algorithm that produces links between them. We call *shift* the operation that consists of moving channels or entire capsules from one layer to another one. The same for a video sequence, but from the next (or previous in case of offline) inference of the network that treats another frame. In our approach, we implement and test three different temporal shift operations:

First dimension features of the first capsule. This implementation is a naive adaptation of the temporal shift module [11] over capsules, since we can see a capsule layer as a $R^{(w,h,cxn)}$ convolution layer if we unroll the capsules. Hence, if we shift only the first convolution kernels of the layer, once roll again, we can see we shifted only the first dimension features of the first capsule (Fig. 1b).

First dimension features of all capsules. In this implementation, we consider capsules as what they are for, a set of convolutional kernels that, together, represent a more complex spatial feature (Fig. 1c). Our motivation is to consider the sub-spatial features inside each capsules independently to other capsules so that a shift of the first kernels of each capsules permits to capture temporal information of each complex spatial features encapsulated. The temporal shift is then an integral part of the capsule behavior.

Entire first capsules. This implementation is a mixed of the two others. As in the original TSM, they shift a small portion of spatial features extracted by convolutional kernels. In our case, we shift a small portion of complex spatial features extracted by capsules by shifting the entire first capsules (Fig. 1d). Our motivation is then relatively similar, temporal shift is independent to capsules behaviour, we want to share complex spatial features extracted between frames.

Offline vs Online. For offline scenarios, since we can treat an entire video at once, we can shift kernels to future frame, but also from future frame (Fig. 1b and Fig. 1c). This allows us to share a maximum of information inside a video, yet, we cannot get access of future frame for online implementation. Moreover, we can only passed our frame one by one to our network, contrary to offline case where we can pass all video frames in a batch. To address those problems, we can temporally cached kernel of capsule information into a temporal tensor so that we can reuse those information on the next frame.

Every implementation has the same structure in what we called a **ShiftConvCaps** (Fig. 2). As in TSM [11], we added a residual branch inside each convolution capsule where the shift and convolution operations are performed. A naive implementation of shift module, where we linearly shift the input

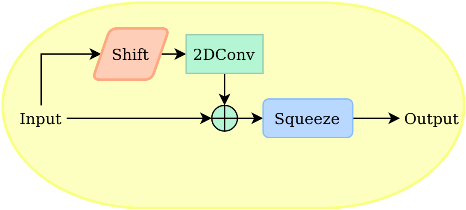


Fig. 2: ConvShiftCapsule representation

and then perform convolution on it, irremediably provoked a loss of information due to the disappearance of spatial features shifted in the current frame. Hence both original input and shifted convolution are fused together before the squeeze activation function of the capsule.

B. *Network Architecture*

The architecture of our 2D Deep Video Capsule Network with ShiftConvCaps, skettched in Fig. 3, is inspired by the DeepCaps model [14]. We conserve the basic structure of a Capsule Network with some first 2D convolution operations followed by convolution capsule and then classification ones. The convolution capsules are replaced with CapsCell [14] that performs three convolution capsules and a residual one. All convolution capsules that performed a convolution stride of $(1 * 1)$ are replaced with ShiftConvCaps.

Since we work on much complex problem with video understanding than image classification, we need to use larger reshape input images $(128 * 112 * 3)$ compared to DeepCaps $(64 * 64 * 3)$ or other 2D capsule networks. However, we observe a quick exploding gradient on the deepest capsule layer using such inputs if they are not enough reduced by the previous convolution layers. The original implementation of DeepCaps does not produce this problem since the first convolution and capsule layers produced very small outputs. However, we consider that these first layers reduced to much the dimension of the input and provoked a loss of information that can be really useful to deal with our video understanding problem. To address this problem, we kept reasonably large capsule dimension on the CapsCell by removing convolution strides in the middle of the network. Then, to offset this large scale data for the classification capsules, we added two extra ConvCaps to have treatable data on the classification capsule layer (Fig. 3). We also decided to augment convolution kernels size of ShiftConvCaps to get nearer from to original Capsule Network vision, observing better results in that way.

Finally, a basic reconstruction module with 2D deconvolution was put at the end of the network to reconstruct input image and assist the training by regularization. We also use the margin loss for loss function as described in Sabour *et al.* [12]:

$$L_k = T_k max(0, m^+ - ||v_k||)^2 + (1 - T_k) max(0, ||v_k|| - m^-)^2 \quad (1)$$

With $m^+ = 0.9$ and $m^- = 0.1$ and where $T_k = 1$ if the sequence belong to class k and $||v_k||$ is the length of the classification capsules.

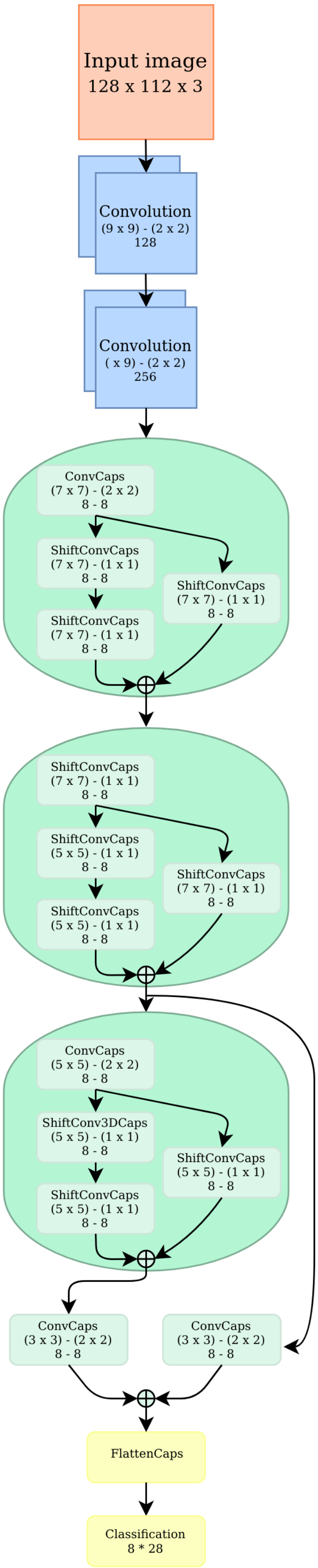


Fig. 3: Our 2D Deep Video Capsule network architecture [14]

IV. Experiments and Results

In this section, we first evaluate our proposed approach on two datasets and compare it with existing state-of-the-art methods based only on color and depth images. We then demonstrate the impact of the temporal shift module over capsule network.

A. Datasets and implementation

We use two datasets to test our new architecture, First Person Hand Action (FPHA) [17] and Dynamic Hand Gesture Dataset (DHG28) [22].

The FPHA dataset is a collection of RGB-D video sequences with more than 100k frames of 45 dialy hand action categories, involving 26 different objects in several hand configurations. It also contained 21 joints of a hand model via 6 magnetic sensors and inverse kinematic that we are not going to use in our experiments.

The DHG dataset contains sequences of 14 hand gestures performed in two ways : using one finger and the whole hand so we can consider 28 class gestures. Each frame of sequences contains a depth image and the coordinates of 22 joints forming a full hand skeleton both captured with The Intel RealSense short range depth camera. We only use depth image in our case.

We used Keras library [28] and Tensorflow [29] as backend for the development of 2D DVCN and trained it on the following configuration: Intel i9 9900k, Nvidia RTX Titan and 32 GB RAM. For all experiments, we used the Adam Optimizer [30] with a starting learning rate of 0.001 and trained the network over 500 epochs on offline shift version.

B. Classification results

On the color data of the FPHA database [17], in order to avoid overfitting, we considerably reduce the size of our network with only 2 CapsCells and 4 capsules of dimension 8 in the remaining ShiftConvCaps. Such a configuration allows us to have a very small network with only 4 millions parameters and still improving results than with a much deeper network. We reduce the action sequences to 16 images per sequence, which is enough because the database provides high definition color images. Indeed, the action performed by the hand is not only a temporal but also a spatial problem. The subject interacts with an object so there is a spatial context that can counterbalance the necessity to deeply analyse the action over the time dimension. We observe that we obtain better classification results over other state-of-the-art methods on color stream with 45 time less parameters (TABLE I). We apply TSM [11] method on FPHA to compare our temporal shift on capsule layers over them on convolution layers. We use their code obtained from Github. We can also see the benefit of capsules over traditional convolution with the 5% better classification than TSM [11] with 5 times less parameters.

Method	Parameters	Accuracy (%)
Two stream-color [31]	46M	61.56
Two stream-flow [31]	46M	69.91
Two stream-all [31]	181M	75.30
TSM [11]	24M	71.57
2D CVNN	**4M**	**76.72**

TABLE I: Comparing 2D DVCN on FPHA dataset [17] over other color stream methods

For the DHG28 dataset [22] we consider the 28 labels cases and use the three CapsCells network as represented in Fig. 3, we obtained a 7M parameters network. Since we have less spatial information, the database being composed of gestures only make in front of the camera, without any object interactions or background context, we decided to consider video sequences of 32 frames to extract as temporal information as possible (TABLE II). We also observe a 10% classification rate improvement compare to TSM [11]. We approach state-of-the art method [32] that uses fusion of 3D CNN and 2D CNN VGG16 [33] and have 20 times more parameters than our method.

Method	Parameters	Accuracy (%)
TSM [11]	24M	58.66
2D-3DCNN Fusion [32]	140M	**74.41**
2D CVNN	**7M**	**68.98**

TABLE II: Comparing 2D DVCN on DHG28 databset [22] over other depth stream methods

To prove the benefits of the temporal shift module over capsule network, we also conducted some experimentation by testing the same network with the same training conditions, same architecture, dataset and hyper-parameters but with or without the temporal shift module applied (TABLE III). We then prove that there is directly an improvement by applying two of the temporal shift operations we implement and that the temporal sharing information have a positive impact for classification with at least a 4% accuracy improvement. However, we observe significant degradation of results when applying the temporal shift on every kernel on the first capsule only (Fig. 1b). We can explain this degradation since this shift is the only one that share partial temporal information. Indeed, in this configuration, every other capsule have to deal, with the routing algorithm, with a capsule that have previous and current frame information ; this inconstant information leads to deterioration of the results. On the contrary, the two other implementations offer real improvement. We can see that temporal shift on every kernels of every capsules (Fig. 1c) and shifting all the first capsules (Fig. 1d) lead to similar improvement by outperforming the classification results. We can explain the improvement since the shift over entire capsule is the most logical way to share temporal information on a capsule network. Indeed, a capsule represents a complex spatial information by encoding in a vector, different characteristics of a same spatial features like the orientation or the scale. The two first implementations then, by shifting the firsts elements of these vectors, do not share pertinent information since they shift very specific and not contextual spatial information. Shifting entire capsules however move temporally all spatial information and explain the best improvement from this implementation.

Shift	Accuracy (%)
No shift	70.01
Shift every kernels first caps	64.14
Shift every kernels every caps	74.33
Shift all first capsules	**76.72**

TABLE III: Comparison of all temporal shift operations and no shift on the FPHA database [17]

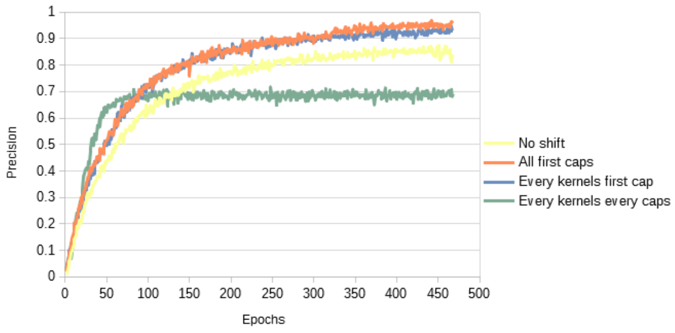


Fig. 4: Evolution of training precision over epochs on different shift operations

We can also see that, regardless of the temporal shift implementation, in addition to the final classification improvement, the training operation is faster to reach same precision in less epochs, even with the temporal shift on every kernels of the first capsule (Fig. 4).

V. Conclusion

In this paper, we proposed a new architecture for video understanding, 2D Deep Video Capsule Network that can deal with large image input without exploding gradient contrary to other capsule network architecture. We show that the integration and implementation of temporal shift module over capsule layer inject relevant and benefit temporal information to process video sequences with no additional computational cost. We also prove that shifting all capsules instead of internal kernels provides better results since we share all complex spatial information captured by capsules. Finally, we are outperforming the state-of-the-art on the FPHA dataset with 45 time less parameters and bring near best results on DHG28 but with 20 time less parameters.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] 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*, vol. 35, no. 1, pp. 221–231, 2012.

[3] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4207–4215.

[4] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[5] 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*, 2014, pp. 1725–1732.

[6] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov, "Exploiting image-trained cnn architectures for unconstrained video classification," *arXiv preprint arXiv:1503.04144*, 2015.

[7] 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*. Springer, 2016, pp. 20–36.

[8] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal relational reasoning in videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 803–818.

[9] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell, "Actionvlad: Learning spatio-temporal aggregation for action classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 971–980.

[10] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[11] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7083–7093.

[12] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in neural information processing systems*, 2017, pp. 3856–3866.

[13] Y. LeCun, "The mnist database of handwritten digits. nec research institute," 1998.

[14] J. Rajasegaran, V. Jayasundara, S. Jayasekara, H. Jayasekara, S. Seneviratne, and R. Rodrigo, "Deepcaps: Going deeper with capsule networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 725–10 733.

[15] K. Duarte, Y. Rawat, and M. Shah, "Videocapsulenet: A simplified network for action detection," in *Advances in Neural Information Processing Systems*, 2018, pp. 7610–7619.

[16] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: a large video database for human motion recognition," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011.

[17] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim, "First-person hand action benchmark with rgb-d videos and 3d hand pose annotations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 409–419.

[18] Q. De Smedt, H. Wannous, and J.-P. Vandeborre, "Skeleton-based dynamic hand gesture recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016, pp. 1–9.

[19] X. Ma, Y. Li, Z. Cui, and Y. Wang, "Forecasting transportation network speed using deep capsule networks with nested lstm models," *arXiv preprint arXiv:1811.04745*, 2018.

[20] S. Srivastava, P. Khurana, and V. Tewari, "Identifying aggression and toxicity in comments using capsule network," in *Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying (TRAC-2018)*, 2018, pp. 98–105.

[21] K. Murakami and H. Taguchi, "Gesture recognition using recurrent neural networks," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, 1991, pp. 237–242.

[22] Q. De Smedt, "Dynamic hand gesture recognition-from traditional handcrafted to recent deep learning approaches," Ph.D. dissertation, 2017.

[23] X. Chen, H. Guo, G. Wang, and L. Zhang, "Motion feature augmented recurrent neural network for skeleton-based dynamic hand gesture recognition," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2881–2885.

[24] 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, 2015, pp. 1110–1118.

[25] Z. Guo, L. Gao, J. Song, X. Xu, J. Shao, and H. T. Shen, "Attention-based lstm with semantic consistency for videos captioning," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 357–361.

[26] J. Zang, L. Wang, Z. Liu, Q. Zhang, G. Hua, and N. Zheng, "Attention-based temporal weighted convolutional neural network for action recognition," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer, 2018, pp. 97–108.

[27] 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.

[28] F. Chollet *et al.*, "Keras: The python deep learning library," *Astrophysics Source Code Library*, 2018.

[29] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 2016, pp. 265–283.

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

[31] 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*, 2016, pp. 1933–1941.

[32] E. Zhang, B. Xue, F. Cao, J. Duan, G. Lin, and Y. Lei, "Fusion of 2d cnn and 3d densenet for dynamic gesture recognition," *Electronics*, vol. 8, no. 12, p. 1511, 2019.

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