LEARNING TEMPORAL INFORMATION FROM SPATIAL INFORMATION USING CAPSNETS FOR HUMAN ACTION RECOGNITION

Abdullah M. Algamdi⋆*, Victor Sanchez*⋆*, and Chang-Tsun Li*†

⋆ Dept. of Computer Science, University of Warwick, UK
† School of Information Technology, Deakin University, Australia

ABSTRACT

Capsule Networks (CapsNets) are recently introduced to overcome some of the shortcomings of traditional Convolutional Neural Networks (CNNs). CapsNets replace neurons in CNNs with vectors to retain spatial relationships among the features. In this paper, we propose a CapsNet architecture that employs individual video frames for human action recognition without explicitly extracting motion information. We also propose weight pooling to reduce the computational complexity and improve the classification accuracy by appropriately removing some of the extracted features. We show how the capsules of the proposed architecture can encode temporal information by using the spatial features extracted from several video frames. Compared with a traditional CNN of the same complexity, the proposed CapsNet improves action recognition performance by 12.11% and 22.29% on the KTH and UCF-sports datasets, respectively.

Index Terms— CapsNet, Human Action Recognition, CNN, routing-by-agreement

1. INTRODUCTION

Convolutional Neural Networks (CNNs) have been shown to attain outstanding results for human action recognition (HAR) [1] by employing either 2D/3D convolutional kernels [2], [3], or multi-stream approaches that learn features from the spatial (frames) and temporal dimensions (optical flow) of the video data [4] [5]. Unlike local feature descriptors [6] [7], which rely on hand-crafted features, CNNs learn features automatically and use translated replicas of these learned features to translate knowledge visual data. This has been shown to be very efficient in visual data interpretation.

Despite their outstanding performance, CNNs do not encode the spatial relationship among the learned features. The multiple pooling layers commonly found in CNNs force the network to ignore valuable information about the precise location and pose of objects, thus discarding any correlation among the learned features. This is a significant drawback for segmentation and object detection tasks.

Capsule Networks (CapsNets) have been recently introduced [8] to overcome some of the weaknesses of CNNs. A CapsNet is a network that aims to perform inverse graphics, i.e., finding the constituent objects displayed on the visual data and their instantiation parameters. CapsNets are equivariant as they preserve detailed information about an object's location and pose throughout the network. This helps the network to learn all the part-whole relationships, determine the precise location of the extracted features, and build a hierarchical representation of objects composed of a hierarchy of parts [9].

Based on the promising results that CapsNets have attained for image classification [8] [10], we propose a 2D architecture based on CapsNet for HAR. Our architecture uses capsules to learn the location and pose of important objects depicted in the input video frames to classify actions without the need to explicitly extract motion information from the data (i.e., optical flow). We also introduce a weight pooling algorithm to reduce the number of predictions to be made by the capsules and improve the classification accuracy. Compared to a CNN of similar complexity and architecture, our CapsNet achieves more accurate recognition results on the KTH and UCF-Sports datasets with improvements of 12.11% and 22.29%, respectively.

The rest of the paper is organised as follows. Section 2 briefly describes the related work on CapsNets and HAR using Deep Neural Networks (DNNs). Our proposed architecture is described in Section 3. Section 4 presents and discusses the performance. Finally, Section 5 concludes this work.

2. RELATED WORK

CapNets: These networks comprise several capsules. A capsule is any function that aims to predict the presence and instantiation parameters of a specific object at a particular location. Capsules output activation vectors, whose length, $l \in [0, 1]$, represents the probability that the object of interest is present in the location associated with the capsule. Alternatively, capsules can output matrices [10], which can better deal with different viewpoints of the visual data. Capsules can be easily created by re-arranging feature maps extracted by traditional convolutional (Conv) layers into vectors (or matrices) representing features extracted from a specific location.

978-1-5386-4658-8/18/$31.00 ©2019 IEEE

CapsNets can handle objects comprising a hierarchy of parts by employing *routing-by-agreement*. To this end, every capsule in layer L predicts the output of every capsule in layer $L+1$, and only when the prediction of capsules in layer L *agrees*, will their outputs be *routed* to the corresponding capsule in layer $L+1$ to accurately determine the instantiation parameters of objects [8].

HAR with CapsNets: Recently, CapsNets have been designed for image classification [11] [12] and object segmentation [13] tasks. To the best of our knowledge, CapsNets have not been previously used exclusively for HAR. CapsNets have been used, however, for human action detection, where HAR is performed as an additional task [14]. That particular work uses 3D convolutional layers to extract the features maps needed to create the capsules, which drastically increases the number of capsules and thus, the computational cost of routing. This is partially addressed by employing capsule pooling.

In a more general framework, DNNs have been successfully used for HAR [15], where 2D convolutions are key components to classify the action depicted in each frame [2]. The work in [16, 17] uses region-based approaches to detect the action across multiple frames. The extracted features are then merged to obtain spatial-temporal information. The work in [18] proposes a two-stream network, one stream takes video frames to analyze the spatial dimension and the other stream takes the optical flow of two consecutive frames to analyze the temporal dimension. Despite capturing the temporal dimension, the optical flow has to be computed beforehand, which may be computationally expensive. The work in [19, 20] uses 3D CNNs to extract both spatial and temporal features from the video frames and then classify actions accordingly. To benefit from the pre-trained weights of ImageNet in 3D CNNs, [1] proposes the two-stream I3D, which inflates the 2D convolution to 3D.

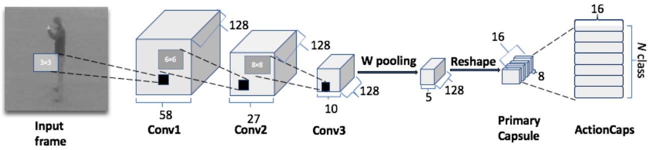

Fig. 1: Proposed CapsNet architecture.

3. PROPOSED CAPSNET FOR HAR

The proposed CapsNet architecture for HAR is depicted in Fig. 1. The input to the network are gray-level video frames. Three 2D ReLU Conv layers with a kernel size of $\{3\times3, 6\times6, 8\times8\}$ and a stride of $\{1, 2, 2\}$, respectively, first extract 128 feature maps, each. To reduce the computational costs of routing, the architecture employs a novel weight pooling approach on the feature maps generated by Conv3. It then employs a primary capsule layer composed of capsules created from the pooled feature maps as 8-dimensional vectors. These primary capsules are fully connected to the ActionCaps, a collection of N capsules, where N is the number of action classes. Each ActionCap is a 16-dimensional vector that interacts with the primary capsules via routing-by-agreement [8].

To reduce overfitting, the architecture uses dropout [21] and batch normalization [22]. Dropout is used after each Conv layer with a 75% drop rate and batch normalisation is used after the input, before each dropout layer, and before the primary capsule layer. These techniques help the network to better generalise and classify more accurately. The network uses cross entropy as the loss function for classification:

$$H(p, q) = -\sum_{x} p(x)\log(q(x)), \tag{1}$$

where p is the true class of the input frame x and q is the predicted value. The ActionCap with the largest magnitude corresponds to the detected class.

It is important to mention that capsules are originally designed to classify images, using e.g. the MNIST [23] or the smallNORB datasets [10], which requires extracting less complex features compared to the case of HAR. Video frames depicting human actions tend to be more challenging with complex backgrounds and illumination changes. HAR then requires extracting more complex features. To tackle this challenge without drastically increasing the computational complexity, our CapsNet employs three Conv layers to extract more complex features with fewer feature maps (128 each) compared to CapsNets used for image classification [8].

Weight pooling: To reduce the high dimensionality of the feature maps produced by Conv layers, large stride values and pooling layers are commonly used. On the one hand, large stride values may prevent the network from extracting all the important features. On the other hand, employing several pooling layers may result in losing important information about the precise location of objects.

In this work, instead of increasing the stride value of the Conv layers or adding several pooling layers, we propose weight pooling, which is inspired by stochastic pooling [24]. Differently from max pooling, which outputs the largest value within a window of values, our weight pooling computes weighted contributions for each value within the window to produce the output value. This is described in Algorithm 1.

Our CapsNet architecture uses a single weight pooling layer with a 2×2 window size and a stride of 2. This guarantees that the pooling operation is only applied once to each 2×2 region. After weight pooling, each of the 128 features maps produced by Conv3 has a dimension of 5×5, which results in 400 primary capsules after re-arranging the pooled feature maps into 8-dimensional vectors.

4. EXPERIMENTS AND RESULTS

We measure the performance of our CapsNet architecture for HAR on the KTH and UCF-sports datasets. The KTH dataset

```
Algorithm 1: Weight pooling algorithm
  Input: feature map, $F$
  Output: pooled feature map, $P$
  for every window $m \times m$ in $F$
  do
      $M \leftarrow m \times m$ area to be pooled;
      $avg \leftarrow$ average of $M$;
      $S \leftarrow avg \times M$;
      $\hat{S} \leftarrow$ normalise $S$;
      $P \leftarrow P \cup$ sum of values of $(M \times \hat{S})$;
  end for
  return $P$
```

comprises videos with simple actions captured in a controlled environment. The UCF-sports dataset, however, comprises realistic videos with very challenging environment conditions, including complex backgrounds with camera motion and illumination changes. All frames are converted to grey scale and re-sized to 60 × 60.

For the KTH dataset, we use the same training and testing split used by [25]. For the UCF-sports dataset, we use the Leave-One-Out (LOO) scheme as in [26] to split the data into training and testing.

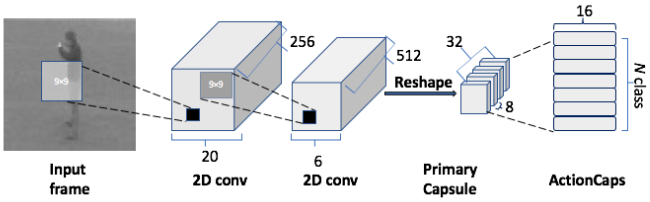

Fig. 2: Baseline Capsule Network.

We evaluate four distinct approaches: 1) a baseline CNN; 2) a baseline CapsNet; (see Fig. 2) 3) our proposed CapsNet with no pooling; 4) our proposed CapsNet with max pooling; and 5) our proposed CapsNet with weight pooling. The baseline CNN replaces the last two layers (primary capsules and ActionCaps) of the proposed CapsNet in Fig. 1 by two fully connected (FC) layers (see Fig. 3). The last fully connected layer is connected with an N class softmax layer. Note that the baseline CNN extracts the same amount of features as compared to our proposed CapsNet. Therefore, evaluating the performance of this network allows determining any improvements introduced by the capsules.

Table 1 tabulates average Correct Classification Rates (CCR) over the splits. These results show that the baseline CNN does not perform very well. State-of-the-art CNNs are usually very deep networks that extract a vast amount of features. Therefore, it is expected that this baseline CNN, which has only three Conv layers, performs poorly.

Our proposed CapsNet with weight pooling outperforms the baseline CNN by 12.11% and 22.29% on the KTH and UCF-sports datasets, respectively. This confirms the power of capsules and routing-by-agreement to classify actions from single video frames compared to using FC layers.

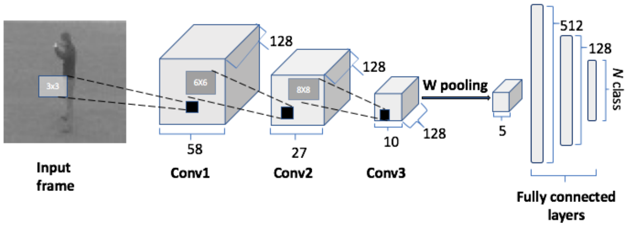

Fig. 3: Baseline CNN.

Table 1: Frame-level CCR (%) and # of parameters (millions) of various networks on the KTH and UCF-sports datasets

Network	# of parameters	KTH	UCF
Baseline CNN	3.3 M	61.41	68.18
Baseline CapsNet	∼21.4 - 31.4 M	64.34	84.91
Proposed CapsNet (no pooling)	∼2.8 - 3.8 M	68.25	81.42
Proposed CapsNet (max pooling)	∼1.9 - 2.2 M	71.71	89.61
Proposed CapsNet (weight pooling)	∼1.9 - 2.2 M	**73.52**	**90.47**

Our proposed CapsNet with weight pooling also outperforms the baseline CapsNet. We find that extracting more feature maps tend to confuse the network, as more capsules tend to be activated in the background and not in those regions depicting the action. Therefore, by reducing the number of features maps and pooling the last set of feature maps, our CapsNet with weight pooling ensures that the most significant features are extracted, thus reducing the chance of confusing. Reducing the number of features maps also reduces the number of training parameters, which helps to reduce training times. The results in Table 1 suggests that the extra features extracted by the baseline CapsNet make the training process computationally expensive (see number of training parameters).

The results in Table 1 also confirm the advantage of using our weight pooling. Our CapsNet with weight pooling outperforms the cases of using no pooling or max pooling. Let us recall that pooling plays an important role in reducing the dimensionality of the extracted feature maps, which help to reduce computational complexity. Unfortunately, this reduction is achieved at the expense of discarding possibly important features. Although our weight pooling discards features, it does this in an appropriate manner by accounting for the values within the windows to be pooled. This pooling helps to enhance performance by reducing noisy features since it computes the contribution of each value within the pooled window to the output average value.

Discussions: Although our proposed CapsNets outperforms the baseline CNN on both the KTH and UCF-sports datasets, it is still far from the results achieved by some of the state-of-art CNNs, which are usually DNNs with several pooling layers and very complex architectures. It is important to

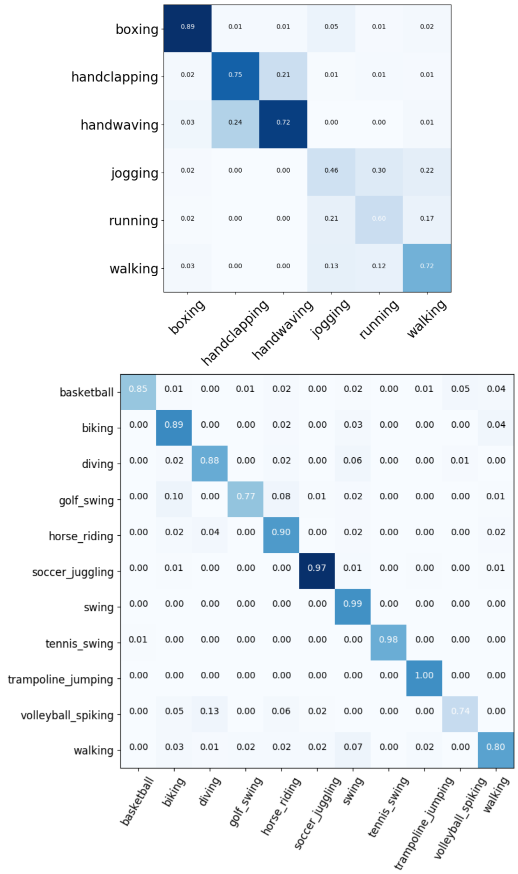

Fig. 4: Confusion matrices of the proposed CapsNet with weight pooling on the KTH (top) and UCF-sports (11 actions - bottom) datasets.

note, however, that our CapsNet is a very simple architecture with only three Conv layers. The CCR improvements tabulated in Table 1 indeed give a notion of the power of capsules and routing-by-agreement for HAR even with a very simple architecture.

Fig. 4 shows the confusion matrices of our CapsNet with weight pooling on the KTH and UCF-sports (for 11 actions) datasets. These matrices show that the majority of the actions are classified with high accuracy. It is important to note that the complex and nosy backgrounds of the frames of these datasets may affect the results in a positive and negative manner. On the one hand, the background can help to classify a frame based on a particular background entity. For example, *playing basketball* in the UCF-sports dataset can be correctly classified not only by the actual human action but also by the basketball ring, the backboard, and the court. The hierarchical composition and location in the scene of these background entities are indeed learned by the capsules. On the other hand, strong noise in the background can activate some of the capsules forcing them to ignore regions where the actual human action is depicted.

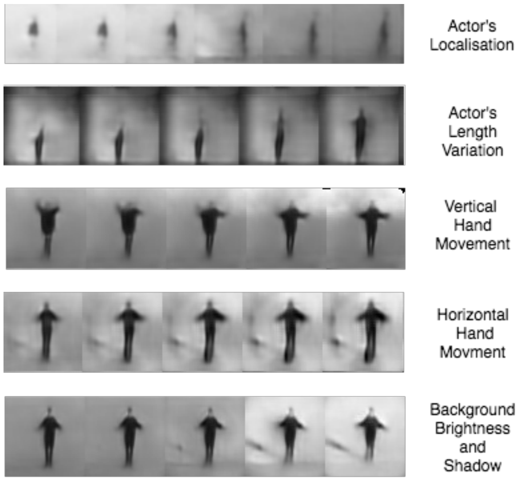

Fig. 5: Manipulation of ActionCaps for the KTH dataset. Each row represents 5 frames reconstructed after modifying one of the 16 dimensions of the output vector by $+/-0.05$.

To better understand the instantiation parameters learned by the capsules, we graphically depict what the ActionCaps encode for the KTH dataset in terms of the space of variations in the way an action is instantiated. To this end, we pass the encoding of only one ActionCap to a reconstruction network by zeroing out other classes [8]. We use our CapsNet in Fig. 1 with five Conv layers without pooling to extract features, and six deconvolutional layers for reconstruction. Fig. 5 shows the reconstructed frames when one of the 16 dimensions of the ActionCap is tweaked by $+/-0.05$. It can be seen that the ActionCaps can indeed encode some of the temporal information of the class by using just video frames. For example, by encoding the multiple positions of the hands, the vertical/horizontal hand movement is encoded by one of the dimensions of the ActionCaps. Similarly, by encoding the multiple actor's positions in the frame, the action of running/walking is encoded by one of these dimensions. The ActionCaps can also encode different instantiation parameters such as the actor's height, background brightness and actor's shadow.

5. CONCLUSIONS

In this paper, we investigated the power of CapsNets for HAR by proposing a simple architecture that can accurately classify actions from video sequences without explicitly extracting temporal information. We also introduced weight pooling to improve the classification accuracy compared to no pooling and max pooling. Our proposed CapsNet outperforms a traditional CNN of similar complexity by 12.11% and 22.29% on the KTH and UCF-sports datasets, respectively, when both networks extract the same features. This confirms the great potential of capsules for HAR. Finally, we showed the ability of capsules to encode the temporal information from the information extracted from individual video frames by encoding the multiple positions of important objects across a number of frames.

6. REFERENCES

[1] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," *CoRR*, vol. abs/1705.07750, 2017.

[2] 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.

[3] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "C3D: generic features for video analysis," *CoRR*, vol. abs/1412.0767, 2014.

[4] 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.

[5] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., pp. 568–576. Curran Associates, Inc., 2014.

[6] R. Leyva, V. Sanchez, and C. Li, "Fast detection of abnormal events in videos with binary features," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2018, pp. 1318–1322.

[7] R. Leyva, V. Sanchez, and C. Li, "Video anomaly detection with compact feature sets for online performance," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3463–3478, July 2017.

[8] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in Neural Information Processing Systems*, 2017, pp. 3859–3869.

[9] A. Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, 2017.

[10] G. E. Hinton, S. Sabour, and N. Frosst, "Matrix capsules with EM routing," in *International Conference on Learning Representations*, 2018.

[11] P. Afshar, A. Mohammadi, and K. N. Plataniotis, "Brain Tumor Type Classification via Capsule Networks," *ArXiv e-prints*, Feb. 2018.

[12] A. Mobiny and H. Van Nguyen, "Fast CapsNet for Lung Cancer Screening," *ArXiv e-prints*, June 2018.

[13] R. LaLonde and U. Bagci, "Capsules for object segmentation," *arXiv preprint arXiv:1804.04241*, 2018.

[14] K. Duarte, Y. S. Rawat, and M. Shah, "Videocapsulenet: A simplified network for action detection," *arXiv preprint arXiv:1805.08162*, 2018.

[15] Y. Kong and Y. Fu, "Human action recognition and prediction: A survey," *CoRR*, vol. abs/1806.11230, 2018.

[16] X. Peng and C. Schmid, "Multi-region two-stream r-cnn for action detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 744–759.

[17] Z. Yang, J. Gao, and R. Nevatia, "Spatio-temporal action detection with cascade proposal and location anticipation," *arXiv preprint arXiv:1708.00042*, 2017.

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

[19] 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, 2013.

[20] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[24] M. D. Zeiler and R. Fergus, "Stochastic pooling for regularization of deep convolutional neural networks," *CoRR*, vol. abs/1301.3557, 2013.

[25] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*. IEEE, 2004, vol. 3, pp. 32–36.

[26] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.