# DRONECAPS: RECOGNITION OF HUMAN ACTIONS IN DRONE VIDEOS USING CAPSULE NETWORKS WITH BINARY VOLUME COMPARISONS

*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

Understanding human actions from videos captured by drones is a challenging task in computer vision due to the unfamiliar viewpoints of individuals and changes in their size due to the camera's location and motion. This work proposes DroneCaps, a capsule network architecture for multi-label human action recognition (HAR) in videos captured by drones. DroneCaps uses features computed by 3D convolution neural networks plus a new set of features computed by a novel Binary Volume Comparison layer. All these features, in conjunction with the learning power of CapsNets, allow understanding and abstracting the different viewpoints and poses of the depicted individuals very efficiently, thus improving multi-label HAR. The evaluation of the DroneCaps architecture's performance for multi-label classification shows that it outperforms state-of-the-art methods on the Okutama-Action dataset.

***Index Terms***— Capsule networks, EM Routing, Dynamic Routing, drone videos, Human Action Recognition

## 1. INTRODUCTION

Human Action Recognition (HAR) is one of the most prevalent challenges in the computer vision community. Many research has been conducted to recognise human actions using videos captured in different contexts, such as sports, surveillance, and care for the elderly [1–3]. As expected, the majority of solutions for HAR within these contexts are limited to videos depicting frontal or side views acquired by personal and closed-circuit television (CCTV) cameras. In contrast, only a very limited number of solutions for HAR using aerial videos exist, as this is a relatively new area of research. Specifically, the analysis of aerial imagery has focused on object tracking [4–7], object detection [4, 7–9] and object counting [4, 8] .

HAR using aerial videos acquired by Unmanned Aerial Vehicles (UAVs), or drones, offers the possibility to design new technologies for search and rescue tasks, surveillance, human interaction understanding, and tracking. The latter is particularly important for robotic perception and navigation using drones [10, 11] . However, working with drone videos is also a challenging task due to multiple changes in the pose and size of objects, occlusions and camera motion. The recent introduction of the Okutama-Action dataset [9] has facilitated the development of solutions for HAR using drone videos [12, 13] . This dataset has succeed in integrating videos depicting real-world, aerial-view scenes of multiple human actions. Despite the recent development of HAR methods for drone videos, there is still an important gap between the state-of-the-art performance of these methods and those designed for videos acquired by personal and CCTV cameras.

Capsule networks (CapsNets) have been shown to overcome some of the weaknesses of convolutional neural networks (CNNs) for image classification [14, 15] by preserving the detailed information about an object's location and pose throughout the network [14]. CapsNets have also been able to achieve promising performance for HAR in videos captured by personal and CCTV cameras [16]. Despite these promising results, some of the most important capabilities of CapsNets have not been thoroughly explored and tested on videos, such as dealing with unfamiliar viewpoints and very small objects, particularly for multi-label HAR, which refers to the task of classifying two or more human actions as being performed simultaneously in a scene.

In this work, we propose a CapsNets architecture for multi-label HAR in drone videos. Our network, hereinafter called DroneCaps, uses two types of features: those extracted by traditional 3D-CNNs and those extracted by a novel Binary Volume Comparison (BVC) layer. These two types of features provide key motion and spatial information that can be exploited effectively by the CapsNets for multi-label classification. Specifically, the proposed BVC layer simplifies the scene into basic shapes that carry motion information with very little background noise. Compared to the state-of-the-art multi-label HAR methods tested on the Okutama-Action dataset, our DroneCaps architecture can achieve improvements of up to 13.75% in terms of Total Accuracy.

The rest of the paper is organized as follows. Section 2 describes the related work on CapsNets and HAR using drones videos. Our proposed DroneCaps architecture is described in Section 3. Section 4 presents and discusses the performance evaluation results. Finally, Section 5 concludes this paper.

**Fig. 1**: An action tube is a collection of bounding boxes, one per frame, depicting an individual performing one or more actions across $L$ frames.

## 2. RELATED WORK

CapsNets aim to perform inverse graphics, i.e., finding the constituent objects of the visual data and their instantiation (pose) parameters. A CapsNet comprises several functions, or capsules, that aim to predict the presence and instantiation parameters of a specific object at a particular location [14]. CapsNets are said to be equivariant as they preserve detailed information about an object's location and pose throughout the network, which are not preserved by CNNs. This characteristic helps the network to better deal with pose changes, translation, rotation and scaling. CapsNets can handle objects comprising a hierarchy of parts by employing *Dynamic Routing* or *Expectation-Maximization (EM) Routing*. In Dynamic Routing, 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 determine the instantiation parameters of objects [14]. On the other hand, in EM Routing, capsules in layer $L$ predict the pose matrix and activation of every capsule in layer $L+1$. Only when the prediction of capsules in layer $L$ agrees, will their outputs be routed to the corresponding capsule in layer $L+1$ [15]. EM Routing can then effectively deal with unfamiliar viewpoints, translation, rotation and scaling.

Most research on HAR for drone videos follows a three-stage process. The first stage detects objects (individuals) in all the video frames. The second stage tracks the detected objects over a number of frames to generate an action tube (see Fig. 1). The third stage involves classifying the actions depicted in each action tube. For multi-label HAR, an action tube may depict two or more actions being performed simultaneously. Since our main contribution is on multi-label HAR, we focus on reviewing the third stage of state-of-the-art methods proposed for drone videos.

In [9], the authors use a two-stream approach for HAR: the first stream uses appearance information and the second stream uses optical flow [17]. Their work is limited to single-label HAR, since their detection algorithm, i.e., the Single Shot multi-box Detector (SSD) [18], cannot handle multiple labels. In [12], the authors use a VGG neural network to extract visual features from objects of interest. They subsequently concatenate these features with a bag-of-words representation by using the Visual Question Answering technique [19]. Like [9], this method reports results only on single-label HAR. More recently, Yang et al. [13] propose the Attention Action Recognition Network (AARN) for multi-label HAR. The AARN extracts spatial features from each frame in an action tube by using 2D convolutional (2D-Conv) layers. The extracted feature maps are then stacked together to form 3D feature maps. These features are fed to a Spatio-Temporal Attention Module (STAM), which acts as an autoencoder to generate attention maps that focus on the detected individuals. The authors compare their AARN against the I3D, a two-stream inflated 3D-CNN [20], and Lite ECO, an efficient CNN architecture for online video understanding [21]. The results show that the AARN outperforms I3D and Lite-ECO for multi-label HAR based on several metrics.

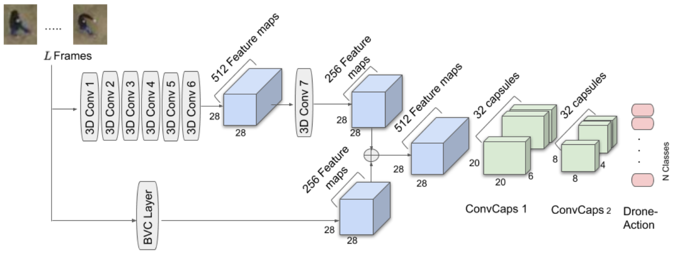

**Fig. 2**: Proposed DroneCaps architecture.

## 3. PROPOSED DRONECAPS ARCHITECTURE

The proposed DroneCaps architecture is depicted in Fig. 2. It comprises two streams with an action tube of $L = 8$ RGB frames as input. The first stream comprises seven 3D-Conv layers. The first six 3D-Conv layers are used for feature extraction. The weights of these layers are initialised based on the weights used for HAR on the Sports-1M dataset. The number of feature maps produced by the sixth 3D-Conv layer is reduced to half by using the seventh 3D-Conv layer with a filter size of $1 \times 1 \times 1$. The second stream is our proposed BVC layer, which comprises three parts: a 3D-Conv layer with 12 non-trainable (i.e., fixed) filters, a non-linear function and a set of learnable weights (see Fig. 3). The final feature maps produced by the BVC layer (see set feature maps, $F$, in Fig. 3) are concatenated with those produced by the first stream along the feature map dimension (see Fig. 2). The concatenated feature maps are used as the input to a convolutional capsule layer, *ConvCaps1*, consisting of 32 capsules. For each capsule, a $4 \times 4$ pose matrix and corresponding activation are obtained by applying a 3D convolution with a filter size of $3 \times 9 \times 9$. The *ConvCaps1* layer is followed by a second convolutional capsule layer, i.e., *ConvCaps2*. Finally, the last layer of the DroneCaps architecture is the *DroneAction* capsule layer, where each capsule represents a class. EM Routing is used to route between each $4 \times 4$ pose matrix and the corresponding activation from the *ConvCaps2* layer to the appropriate capsule in the DroneAction layer. The DroneCaps architecture uses the *spread loss function* [15, 16]:

$$L = \sum_{a_i \neq a_i^t} L_i, \tag{1}$$

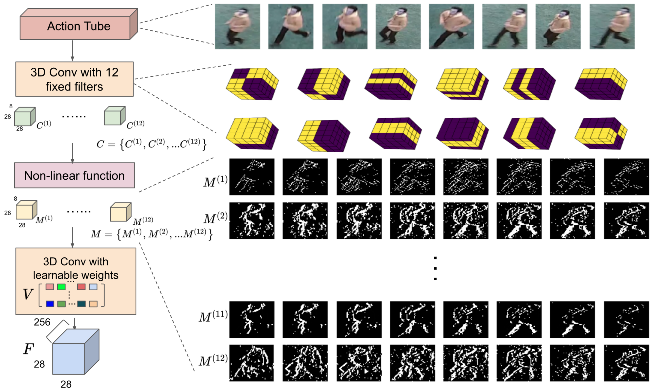

**Fig. 3**: The proposed BVC layer. $V$ is the only set of $12 \times 256$ learnable weights and $F$ is the output set of 256 2D feature maps.

where $L_i = max(0, m - (a_i^t - a_i))^2$ is the loss of the $i^{th}$ capsule at the DroneAction layer, $a_i$ is its predicted output and $a_i^t$ is its corresponding target value. For multi-label HAR, multiple capsules at the DroneAction layer can be simultaneously active and the total loss is the summation of the losses of the erroneous capsules. i.e., those for which $a_i \neq a_i^t$. The margin value, $m$, penalizes more harshly cases when $a_i$ is high, i.e., cases when the prediction indicates that the associated class is present but the ground truth indicates that the associated class is not present ($a_i^t = 0$). This $m$ value is set to 0.2 at the beginning of the training process and increased up to 0.9. The proposed BVC layer consists of a 3D-Conv layer, a non-linear function applied to the sets of feature maps produced by the 3D-Conv layer and a set of learnable weights, $V$ (see Fig. 3). The 3D-Conv layer comprises twelve 3D filters, each with a size of $4 \times 4 \times 4$. These filters have fixed values of -1 or +1 and effectively divide a 3D region into two subregions: one represented by all locations with a value of +1, i.e., region $r_1$, and another by all locations with a value of -1, i.e., region $r_2$. By convolving the input action tube with the $n^{th}$ fixed 3D filter, a region of the action tube is first divided into $r_1$ and $r_2$, and the intensity of the pixels, $p$, in $r_1$ and $r_2$ are summed:

$$s_1 = \sum_{p \in r_1} p; \; s_2 = \sum_{p \in r_2} p. \tag{2}$$

Finally, a total value is computed for the position $\{w, h, l\}$ of the resulting 3D feature map, $C^{(n)}$: $C_{w,h,l}^{(n)} = s_1 - s_2$ (see Fig. 4). Note that value $C_{w,h,l}^{(n)}$ reflects how $r_1$ and $r_2$ compare in terms of intensity sums. For example, if $C_{w,h,l}^{(n)}$ is negative, $s_2 > s_1$, which implies that the intensity sum in $r_2$ is larger value than the intensity sum in $r_1$ [2].

The 3D-Conv layer outputs twelve 3D feature maps, $\mathcal{C} = \{C^{(1)}, C^{(2)}, ...C^{(12)}\}$, one for each fixed 3D filter. Each feature map in $\mathcal{C}$ has $L$ 2D feature maps of spatial dimensions $W \times H$, where $L$ is the number of frames of the input action tube, as defined before. The 3D-Conv layer uses a stride of $1 \times 4 \times 4$ to reduce the spatial dimension of the 3D feature maps with respect to the spatial dimension of the input action tube, while keeping $L$ constant. A non-linear function (ReLU) is applied to the 3D feature maps in $\mathcal{C}$ to map negative values to zero. This produces twelve 3D feature maps with non-negative values, $\mathcal{M} = \{M^{(1)}, M^{(2)}, ...M^{(12)}\}$. Finally, a set of $12 \times 256$ learnable weights, $V$, are used to create a final set of 256 2D feature maps, $F$, by merging the 3D feature maps in $\mathcal{M}$ in different ways. Weights $V$ are learned through the backpropagation of the classification errors through the BVC layer up to this point. The computation of the final set of 2D feature maps, $F$, is implemented by a 3D-Conv layer with a $1 \times 1 \times 1$ filter size [22].

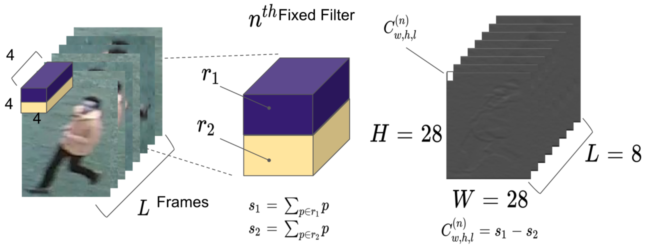

**Fig. 4**: Example computation of the 3D feature map $C^{(n)}$ for the $n^{th}$ fixed 3D filter. Values in $C^{(n)}$ are real numbers.

It is important to highlight one of the main advantages of the BVC layer. The 3D feature maps in $\mathcal{M}$ simplify the content of the input action tube by representing it as a collection of simple structures, e.g., lines and basic shapes, that carry motion information (thanks to the fixed 3D filters) with little noise in the background. These 3D feature maps, after being merged into the final set of 2D feature maps, $F$, can then help the ConvCaps layers of our DroneCaps architecture to extract powerful motion information associated with the objects of interest.

## 4. EXPERIMENTS AND RESULTS

We measure the performance of our DroneCaps architecture for multi-label HAR on the Okutama-Action dataset [9]. The training set consists of 33 videos while the test set consists of 10 videos, as suggested in [9]. Since our work focuses on HAR, we manually extract all action tubes from the training and test sets. A total of $\{572, 117\}$ action tubes are extracted from the training and test sets, respectively. No data augmentation is used for evaluation.

We evaluate six different HAR methods: 1) The AARN in [13], 2) I3D [20], 3) Lite ECO [21], 4) a 3D CapNets architecture with Dynamic Routing (3DCapsNets-DR), 5) a 3D CapsNets architecture with EM Routing (3DCapsNets-EM), and 6) our proposed DroneCaps architecture. 3DCapsNets-DR comprises six 3D-Conv layers followed by a primary capsule layer and a DroneAction layer (see Fig. 5) [23]. 3DCapsNets-EM follows the same architecture as our Drone-Caps architecture,

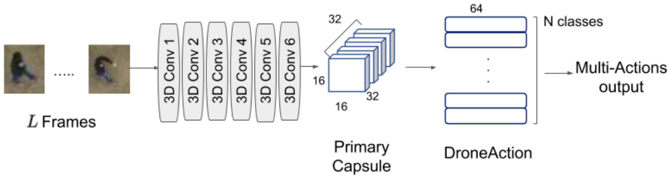

**Fig. 5**: 3D CapsNets architecture with Dynamic Routing.

but without the BVC layer and the $7^{th}$ 3D-Conv layer in the first stream. All evaluated methods use the same action tubes for training and testing.

We use four different metrics to evaluate the results for multi-label HAR: Total Accuracy, Hamming Loss, One Error, and Exact Match Ratio. Total Accuracy (%) is computed as follows:

$$TA = 100 \times \frac{1}{S} \sum_{i=1}^{N} match(\hat{y}^{(i)}, y^{(i)}) \in [0, 100], \quad (3)$$

where $N$ is the number of action tubes tested, $y^{(i)}$ is the ground truth vector for action tube $i$, $\hat{y}^{(i)}$ is the corresponding predicted vector, $S$ is the total number of true classes present in all the tested action tubes, and function *match* returns the number of classes out of the $k$ true classes in $y^{(i)}$ that are present in $\hat{y}^{(i)}$ as the $k$ classes with the highest activation value. The Hamming Loss [13] measures the average Hamming Distance (HD), as a proportion, between $y^{(i)}$ and $\hat{y}^{(i)}$, after rounding the predicted probabilities in $\hat{y}^{(i)}$ to 0 or 1:

$$HL = \frac{1}{NL} \sum_{i=1}^{N} \mathrm{HD}(\hat{y}^{(i)}, y^{(i)}) \in [0, 1], \quad (4)$$

where $L$ is the cardinality of $y^{(i)}$. The One Error metric measures the proportion of action tubes whose predicted class with the highest activation value is not in the set of the true classes:

$$OneError = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[f(\hat{y}_i) \notin g(y_i)] \in [0, 1], \quad (5)$$

where $f(\hat{y}_i)$ returns the class with the highest activation in $\hat{y}_i$ and $g(y_i)$ returns the set of true classes. Finally, the Exact Match Ratio calculates the proportion of action tubes whose predicted probabilities (after rounding values to 0 or 1) are exactly the same as their corresponding ground truth labels:

$$ExactMatchRatio = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[\hat{y}^{(i)} = y^{(i)}] \in [0, 1]. \quad (6)$$

Table 1 tabulates the performance of all tested methods using the test action tubes extracted from the dataset. Note that all CapsNets architectures outperform the AARN [13], Lite-ECO and I3D-RGB, which confirms the capability of CapNets in dealing with challenging cases that may include occlusions, various viewpoints and very small objects. Let us recall that no data augmentation is used for evaluation, hence, the number of training samples is limited. The CapsNets architectures, however, can effectively extract key information from both the spatial and temporal dimensions of the limited number of training samples. This characteristic is especially useful when working with action tubes extracted from high resolution videos acquired by drones, where objects (i.e., individuals) usually appear to be very small in size.

**Table 1**: Results of all tested methods for multi-label HAR on the Okutama dataset. (*) Reproduced results. ↑: the higher the better. ↓: the lower the better

| Method | Total Acc. ↑ | Hamming Loss ↓ | One Error ↓ | Exact Match Ratio ↑ |
|---|---|---|---|---|
| AARN [13]* | 33.75 | 0.158 | 0.658 | 0.179 |
| Lite ECO [21]* | 36.25 | 0.147 | 0.589 | 0.222 |
| I3D (RGB) [20]* | 38.12 | 0.131 | 0.589 | 0.213 |
| 3DCapsNet-DR | 39.37 | 0.172 | 0.581 | 0.247 |
| 3DCapsNet-EM | 41.87 | 0.132 | 0.658 | 0.240 |
| DroneCaps | **47.50** | **0.119** | **0.572** | **0.290** |

3DCapsNets-EM outperforms 3DCapsNets-DR as EM routing separates the activation of capsules from their poses. In Dynamic Routing, pose and activation are jointly represented by a vector. The vector's orientation is the pose, while its length represents the activation. Separating pose from activation helps recognizing objects from different viewpoints (i.e., elevations and angles), which is particular useful when working with drone videos.

Our proposed DroneCaps architecture outperforms all other CapsNets architectures by up to 8.13% in terms of Total Accuracy. Since 3DCapsNets-EM is equivalent to our DroneCaps architecture but with no BVC layer, these results also confirm the advantages of using the BVC layer to improve multi-label classification. Fig. 3 shows a number of sample 3D feature maps from $\mathcal{M}$ as generated by the BVC layer after convolving an action tube with the set of 3D fixed filters and applying a non-linear function. Note that these 3D feature maps describe the motion among a set of frames in terms of basic structures, like lines. Such a simple representation of the input action tube helps the DroneCaps architecture to focus on the object of interest (i.e., the person depicted in the action tube in this case) and disregard the background, which may be a source of noise that can affect the classification results.

## 5. CONCLUSION

In this paper, we proposed the DroneCaps architecture for multi-label HAR in drone videos. This architecture, which is based on CapsNets, uses a novel BVC layer to enhance the 3D-CNN features used by the capsule layer. Specifically, the BVC layer can simplify a scene as a collection of simple structures that carry motion information. The proposed DroneCaps architecture improve the accuracy by 13.75% compared to current state-of-the-art methods as evaluated on the Okutama-Action dataset for multi-label classification.

# References

[1] H. Gammulle, S. Denman, S. Sridharan, and C. Fookes, “Two stream lstm: A deep fusion framework for human action recognition,” in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 177–186, IEEE, 2017.

[2] R. Leyva, V. Sanchez, and T.-L. Chang, “Fast binary-based video descriptors for action recognition,” in *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, IEEE, 2016.

[3] C. Huang, C. Wang, and J. Wang, “Human action recognition system for elderly and children care using three stream convnet,” in *2015 International Conference on Orange Technologies (ICOT)*, pp. 5–9, Dec 2015.

[4] T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye, “A large contextual dataset for classification, detection and counting of cars with deep learning,” in *European Conference on Computer Vision*, pp. 785–800, Springer, 2016.

[5] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, “Learning social etiquette: Human trajectory understanding in crowded scenes,” in *European conference on computer vision*, pp. 549–565, Springer, 2016.

[6] M. Mueller, N. Smith, and B. Ghanem, “A benchmark and simulator for uav tracking,” in *European conference on computer vision*, pp. 445–461, Springer, 2016.

[7] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, “Vision meets drones: A challenge,” *arXiv preprint arXiv:1804.07437*, 2018.

[8] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, “Drone-based object counting by spatially regularized regional proposal network,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4145–4153, 2017.

[9] M. Barekatain, M. Martí, H.-F. Shih, S. Murray, K. Nakayama, Y. Matsuo, and H. Prendinger, “Okutama-action: An aerial view video dataset for concurrent human action detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 28–35, 2017.

[10] M. Obaid, F. Kistler, G. Kasparavičiūtė, A. E. Yantaç, and M. Fjeld, “How would you gesture navigate a drone? a user-centered approach to control a drone,” in *Proceedings of the 20th International Academic Mindtrek Conference*, pp. 113–121, 2016.

[11] A. A. Zhilenkov and I. R. Epifantsev, “System of autonomous navigation of the drone in difficult conditions of the forest trails,” in *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 1036–1039, IEEE, 2018.

[12] A. Soleimani and N. M. Nasrabadi, “Convolutional neural networks for aerial multi-label pedestrian detection,” in *2018 21st International Conference on Information Fusion (FUSION)*, pp. 1005–1010, IEEE, 2018.

[13] F. Yang, S. Sakti, Y. Wu, and S. Nakamura, “A framework for knowing who is doing what in aerial surveillance videos,” *IEEE Access*, vol. 7, pp. 93315–93325, 2019.

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

[15] G. E. Hinton, S. Sabour, and N. Frosst, “Matrix capsules with em routing,” 2018.

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

[17] G. Singh, S. Saha, M. Sapienza, P. H. Torr, and F. Cuzzolin, “Online real-time multiple spatiotemporal action localisation and prediction,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3637–3646, 2017.

[18] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[19] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, “Vqa: Visual question answering,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2425–2433, 2015.

[20] 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*, pp. 6299–6308, 2017.

[21] M. Zolfaghari, K. Singh, and T. Brox, “Eco: Efficient convolutional network for online video understanding,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 695–712, 2018.

[22] F. Juefei-Xu, V. Naresh Boddeti, and M. Savvides, “Local binary convolutional neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 19–28, 2017.

[23] A. M. Algamdi, V. Sanchez, and C.-T. Li, “Learning temporal information from spatial information using capsnets for human action recognition,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3867–3871, IEEE, 2019.