Acta Cybernetica **24** (2019) 157–172.

Combining Common Sense Rules and Machine Learning to Understand Object Manipulation*

András Sárkány[ab], Máté Csákvári[ac], and Mike Olasz[c]

Abstract

Automatic situation understanding in videos has improved remarkably in recent years. However, state-of-the-art image processing methods still have considerable shortcomings: they usually require training data for each object class present and may have high false positive or false negative rates, making them impractical for general applications. We study a case that has a limited goal in a narrow context and argue about the complexity of the general problem. We suggest to solve this problem by including *common sense rules* and by exploiting various state-of-the art *deep neural networks* (DNNs) as the detectors of the conditions of those rules.

We want to deal with the manipulation of unknown objects at a remote table. We have two action types to be detected: 'picking up an object from the table' and 'putting an object onto the table' and due to remote monitoring, we consider monocular observation. We quantitatively evaluate the performance of the system on manually annotated video segments, present precision and recall scores. We also discuss issues on machine reasoning. We conclude that the proposed neural-symbolic approach a) diminishes the required size of training data and b) enables new applications where labeled data are difficult or expensive to get.

Keywords: situation understanding, event recognition, computer vision

1 Introduction

When we talk about situation understanding in AI, we usually imagine a scenario with people acting in front of a camera and the task of the computer system is to assign categories to the ongoing events. Our work presented in this paper is about detecting one particular example of such events: an object being lifted up from or put down on a table. In most cases the need to recognize such a scenario does not

*This research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

[a]Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary.

[b]Present address: Argus Cognitive Inc., USA

[c]Argus Cognitive Inc., USA

DOI: 10.14232/actacyb.24.1.2019.11

stand alone, rather it comes from a higher level goal or some kind of application logic. Looking at this problem from a top-down perspective, it is generally true that fulfilling one such higher level goal requires multiple lower level recognition tasks to be solved and this hierarchy can go multiple levels.

We start from the concept of spatio-temporal pattern recognition that we will use to conceptualize the complexity of such tasks. Pattern recognition in general, is the process of analyzing data and making a decision, such as classifying a sample to different categories, with the help of known regularities in the data [5]. In our case, pattern recognition concerns spatio-temporal ones that we shall call event recognition in the following.

In the case of classification we expect the data space to have 2 properties: 1) the data points corresponding to a category are confined to a region of the space possibly having lower effective dimensionality 2) the data space is locally smooth around these confined regions, meaning that data points close to each other usually belong to the same category. These concepts are analogous to the generalization property. However the data space that we work in not necessarily holds these properties but with the help of highly nonlinear transformations such a space can be constructed. This is especially true in real-world applications where data comes from sensors and it is represented in high dimensional space, like images.

Known regularities encoded in the machine can have two sources: a) human knowledge about the data and b) a model trained by data with an algorithm. Humans can comprehend and give rules for categories for a data space of maximum 2-3 effective dimensions if it has the confinement and smoothness properties. Unsupervised machine learning algorithms can perform nonlinear dimension reduction, but for most real world problems supervised methods are needed, which require training labels. (Also high dimensional data requires more advanced algorithms.)

Annotated training data is the most expensive component of the pattern recognition process and its size rests on two main factors. First, high dimensionality of the input exponentially increases the required training data, also known as the curse of dimensionality [4].

Secondly, special target categories are less frequent in real life therefore collecting the necessary amount of data requires more resources (time, money). Besides, finding the regularities that characterize a special category in comparison to a more general one from the same input requires more training data.

Consequently end-to-end machine learning from high dimensional data to very special categories requires the most expensive kind of data, and the most advanced algorithms.

Our scenario, and other examples of situation understanding, fit the description above: the video recordings we work with, consisting of $\geq$1 million 3-channel pixels for 25 fps, make a very high dimensional input. Collecting appropriate training data that contains the variance of the possible samples requires many subjects in many different scenarios from different viewpoints.

In our scenario the scene is recorded by a monocular RGB camera. Algorithm development for such devices is motivated by the inexpensiveness and commonness of them and it opens a path for a wide range of applications.

Lifting up or putting down an object viewed from an arbitrary (but not maliciously) placed camera is exposed to ambiguity even for human viewers of the scene because of a) occlusion of body parts and objects (to add to the confusion occlusion is caused by other body parts and objects) b) other type of hand movements similarly executed c) illusory perceptions caused by the information loss of the camera projection. So as data is the biggest obstacle how can we save on the costs of data acquisition?

We propose a process of breaking down a difficult event recognition task, which is only solvable with very expensive data, into smaller problems that we can solve with less resource such as freely available data and human knowledge. To the best of our knowledge, this work is the first one that offers a non-obtrusive automated solution to the quantification of picking up and putting down objects.

We describe our approach and the actual recognition pipeline implemented in this paper in Section 2. We compared our method to ground truth annotations on videos provided by Rush University Medical Center showing Autism Diagnostic Observation Schedule, Second Edition (ADOS-2) sessions with different patients. We report the results of our evaluation in Section 3. We discuss the results and provide possible improvements for future work in Section 4.

2 Method

In this section we describe our methodology for designing our pattern recognition pipeline. We refer to the related work Section 2.1. We outline the theoretical approach in Section 2.2. We used multiple machine learning models, algorithms and data trained or developed by other researchers, we describe these in Section 2.3. Our event recognition pipeline is depicted in Section 2.5.

2.1 Related work

There have been many advances in situation understanding and activity recognition in recent years and many datasets have been created for sets of action categories that can be used as benchmarks. In the early days the task was set as a classification task [9] but it moved to a detection task as methods evolved [18]. Recently most methods first extract sparse or dense timestamps and represent events by various descriptors, then use these to recognize similar space-time intervals. Some use strong video features such as histogram of gradients (HOG), motion history images (MHI) [2], others utilize pose estimation [18] or object detection [21]. There are supervised [8] and self-organizing solutions [3] as well, also wide usage of deep learning methods [19]. It is common in them that they use training data and they learn from samples of the event classes they aim to detect.

In contrast, our approach combines several detectors, put together by exploiting human knowledge and it doesn't require samples of the target event class, making it a viable alternative.

2.2 Approach

Our key idea is to transform our pattern recognition task to the composition of multiple "smaller" tasks. With the help of our human intuition we include a more general concept between the input data and our special category. The training data for this general concept is cheaper to manufacture than what we would have needed for our original task. The representation of this general concept has significantly less dimensions than the dimensionality of the original input data, lowering the required training data size for the second recognition task, which is identifying the special concept in the general one.

For example our recognizable action, lifting up an object, is a special case of a pose time series data of a person, so by utilizing a pose estimation algorithm like the convolutional PoseMachine [23], we include a general concept, Pose series, and we can define a new pattern recognition task where the input is the representation for this new concept and our target category is our original target category. Pose series will be represented by 17 joints, each with 2 dimensions multiplied by the length of the series in image frames which is an enormous decrease compared to the dimensionality of the original video data space.

This decomposition can be done arbitrary number of times introducing multiple intermediate concepts. Each decomposition replaces a pattern recognition task with two easier one in terms of data acquisition. Since information is lost when adding an intermediate concept, "sibling" concepts that are between the same input- and target data representations are necessary components. Eventually intermediate concepts will comply to the important factors for the original recognition task.

Each recognition task can be solved in any possible way outlined previously: a) supervised machine learning b) unsupervised dimension reduction c) human rules to assign set of data to a target category. Only a) requires training data where a very beneficial possibility is to use off-the-shelf data (or even trained models) created by others for a general concept that can be used for one's target category as well. From the perspective of expensive data acquisition the coupling of b) and c) can turn the tide from impossible to possible.

We stated previously that humans can formulate rules for comprehensible 2-3d spaces and this knowledge acquisition can save the collection of training data. Dimension reduction on its own can't really solve pattern recognition but can produce the representation that is usable for human rule creation. If we can decompose the original task in a way that such intermediate recognition task that is solvable with dimension reduction and human rules emerges, we can omit a data acquisition step.

The development of such a pipeline is an evolutionary process: we investigate the effect of including and excluding components through quantified metrics and qualitative analysis of samples of false positive and false negative recognitions. Consequently our pipeline have many other variations with other intuitive ideas.

2.3 External components

We briefly list the algorithmic components used in our pipeline. For further details please refer to the literature:

- **Convolutional Pose Machine**[23]: A deep learning method for human pose estimation. It provides estimation for key anatomical points of the human body. We use this method mainly to extract elbow and wrist coordinates so that we can further analyze the lower arm movement of the subject.
- **Mask-RCNN**[11]: A variant of the popular Faster-RCNN[17] object detection method. Other than providing bounding boxes for many object categories it also provides fine-grained instance segmentation of the underlying object.
- **Hand detector**: We used an R-FCN[6] based hand detector trained on a hand dataset [16].
- **Optical flow**: We used FlowNet2.0[12] as optical flow algorithm.

2.4 Target event description

We consider a scenario where a person interacts with objects on a flat surface, most likely a table viewed from a stationary camera. We decompose the event into three required subevents.

Putting down an object:

1. the person holds an object in her hand away from the future release location
2. the person puts it down on the surface
3. the person releases the object and moves her hand away from the release location

Picking up an object:

1. the object is on the table, and the person's hand is away from the object location
2. the person moves her hand towards the object and grabs it
3. the person takes the object away from its original position on the table

We introduced some limitations with these event definitions. We add three more to the list:

1. only events shorter than 10 seconds are taken into consideration
2. the object needs to be visible in the video for at least a few frames at the appropriate step (Put down 3., Pick up 1.)

3. the location of the object (without the object being there) needs to be visible for at least for a few frames at the appropriate step (Put down 1., Pick up 3.)

4. the required distance of the hand from the object location (pixel-wise on the video) is approximately the same size as a hand.

These limitations makes our task more special which means that there will be events that would fall into the category of picking up an object to a human observer but we don't consider the recognition of those.

2.5 Pipeline

As we described in Section 2.2 our goal is to build a pattern recognition pipeline that minimizes data acquisition which is extremely expensive in the case of a very special target category. We relied on our human intuition to decompose the original task and developed it in an evolutionary fashion.

Our main idea is twofold: when an object is moved there are two frames from the video of the stationary camera where the only difference of these images relevant parts determine the object's pixel representation; it "appears" or "disappears". We also note that grabbing or releasing an object involves a temporary stop of the hand's movement. We combine these two ideas to an algorithm where we first search for points in time when the hand is stopped, select these as candidates, then search one frame preceding and one following the candidate point, where the difference of the two frames shows the (dis)appearing object.

If a candidate is false, for example the person just rested her hand on the table for a second without moving any object, then the difference of the retrieved images will be zero, as both images show the part of the table unchanged.

We specify common sense assumptions and derive algorithmic components to:

1. select candidate positions

2. refine candidate positions by hand detection

3. select the two relevant frames

4. process the images to neglect irrelevant differences caused by interfering actions in the scene

5. take the image difference

In the following sections we describe each step in detail.

2.5.1 Candidate selection

We find timestamps of object grabbing by assuming that this action requires that the hand stops for at least an instant. Thus we look for such changes in the speed of the forearm which we measure by optical flow at regions estimated from elbow and wrist joint coordinates. If the magnitude of average velocity in that region

is at a local minima, then the timestamp is selected as a candidate for change detection. The local minimas are found by using gaussian filter on the velocity magnitude signal and then using a peak finding algorithm[7]. For each of the candidate position we assume that the appearing/disappearing object is occluded by the hand in the instant of releasing/grabbing.

2.5.2 Candidate refinement

We refine the candidates obtained in the previous step by using an R-FCN based hand detection algorithm[6]. If the hand detection fails, meaning the hand cannot be find where the movement stopped then we discard this candidate position. A possible situation where this can happen is that the hand is under the table. We are using the bounding boxes obtained here as a RoI (region of interest) in the following steps.

2.5.3 Image selection

In this step we select two frames, I_{t_1} and I_{t_2}, that will be used at the image differencing step. We search backwards and forward in time to find frames where the object is not occluded by the hand, to find I_{t_1} and I_{t_2}. This is done by simply checking if the bounding boxes of the hand moved significantly since the frame in which the object was grabbed. The search has a max duration parameter in both directions, if the hand doesn't leave the RoI in that time span, the candidate position is discarded. This parameter is set to 120 frames (4.8 seconds for a 25 fps video).

2.5.4 Interference removal

We found that the selected frames and RoI given by previous steps contains differences other than the object we were looking for. These differences come from different sources: effect of actors interfering in the RoI (e.g. body parts, shadow, other manipulated object). To neutralize these effects we create a binary mask on the RoI that neglects pixels that belongs to these phenomena. In fact we estimate multiple binary masks with different strategies and take the union of the relevant pixels found by each method. These are the following:

2.5.5 Interference removal - Body occlusion

We found that in many cases there were body parts inside the RoI which accounted for many of the changes that could be found between the two frames. We used Mask-RCNN[1][11] for filtering out pixels corresponding these parts.

2.5.6 Interference removal - Long-term optical flow

There are changes between I_{t_1} and I_{t_2} that correspond to small movements over a longer period of time. A possible example is when the edge of the paper moved on

which the object was placed. To account for these differences we use optical flow between I_{t_1} and I_{t_2}.

2.5.7 Interference removal - Short-term optical flow

There can be ongoing actions inside RoI in I_{t_1} and I_{t_2} which we would also like to filter out. These events are typically hand movements or some other object movement. The optical flow between a frame at time instant t and $t-1$ helps in removing any these effects inside the RoI (e.g. hand is still there but moving). This could be done for any t and $t-k$ time instants, however we found $k=1$ to be sufficient.

2.5.8 Image difference

We combine the binary masks obtained from previous steps to I_{t_1} and I_{t_2} then taking their difference as follows [13]:

$$I_{final}(x,y) = \| I_{t_1}(x,y) - \left(\frac{\sigma_1}{\sigma_2}(I_{t_2}(x,y) - \mu_2) + \mu_1 \right) \|_2$$

where μ_1, σ_1 and μ_2, σ_2 are the mean and standard deviation of I_{t_1} and I_{t_2} respectively.

After calculating the difference image, we perform the following steps to produce the final mask:

1. threshold I_{final} to keep changed part of the images
2. drop small continuous blobs of the difference image for noise reduction

If the remaining covered area of the final mask is larger than 1% of the image area then we keep the difference as the object threshold, otherwise we discard the candidate position.

3 Evaluation

3.1 Data

We evaluated our pattern recognition pipeline on video segments that show a child taking part in an ADOS-2 diagnostic interview. This test includes different types of playful activities conducted by a clinician who is also present in the same room. The clinician interacts with the child during these games. Some of these tests contain objects that are moved from one place to another. We selected two activities "Puzzle game" and "Storytelling with toys" from three subjects. Segments for these games lasts 2-3 minutes and 10 minutes respectively. The dimensions of the puzzle pieces are around 5 cm x 3cm x 0.5cm colored blue and purple. The toys used for storytelling are dolls, plastic animals and tools of various sizes, their largest dimension ranges from 5 cm to 25 cm and colored diversely (examples can

(a) Selected image I_{t_1} (b) Selected image I_{t_2} (c) Final mask applied to I_{t_1}

Figure 1: A few examples from our evaluations. The algorithm starts by finding instants when a hand stops , see, column (b). Frames selected before and after the time instants of the hand stops are shown in columns (a) and (c), respectively. Column (d): masked images after applying the derived filters and taking the image differences. (See text for more information).

be seen on Figure 1). The room (size: 2.5 m x 2.5 m) has the same layout in all recordings with a single table, two chairs and optionally some cabinets holding the different tools for the different tasks. A single light source illuminates the room to standard indoor lighting.

Videos were recorded with a resolution of 1920x1080 pixels. Ground truth annotations of the precise temporal extent of the events were created with our own video annotator tool. Annotations were created and verified by our colleagues using a detailed description (Section 2.4), basically a set of rules about the relevant events, i.e. "picking objects up" and "putting objects down". We would like to emphasize that the ADOS-2 interviews were designed, conducted and recorded independently from our work, meaning that the properties of our proposed system had no effect on how the data was collected.

Statistics of each video segment's length and ground truth coverage is on Table 1.

Table 1: Length of input videos and coverage of ground truth annotations in our data

Subject		1	2	3	All
Puzzle	Length (s)	150	120	120	390
	Ground Truth coverage	31.7%	53.6%	39.3%	40.8%
Storytelling with toys	Length (s)	640	540	615	1795
	Ground Truth coverage	6.9%	20.1%	14.4%	13.4%

3.2 Methodology

We report the performance of our pipeline on these videos with precision and recall metric scores. We show each component's added value for the pipeline with metrics to justify its need in the pipeline by running different versions of our pipeline that include different combinations of components. We name and describe them as follows:

1. P1: Each handstop candidate is considered a positive detection (Section 2.5.1)
2. P2: The candidates of P1 are filtered by candidate refinement (Section 2.5.2)
3. P3: Before and after image is retrieved. The remaining candidates are considered a positive detection (Section 2.5.3).
4. P4a: Filter the candidates of P3 by the image difference of I_{t_1} and I_{t_2} (Section 2.5.8)
5. P4b: Only apply long-term optical flow filtering described in (Section 2.5.6) and image difference (Section 2.5.8)
6. P4c: Only apply short-term optical flow filtering described in (Section 2.5.7) and image difference (Section 2.5.8)
7. P4d: Only apply body-occlusion filtering described (Section 2.5.5) and image difference (Section 2.5.8)
8. P4: Apply all inference removing filters and image difference utilizing the full pipeline.

P1 selects candidate time instances from the video and all other pipeline versions filter these initial candidates. Each pipeline's output is a list of candidate time instance that can be compared to the ground truth annotations with pattern recognition metrics, like precision and recall.

Since certain pairs of pipeline versions differ only in one algorithmic component (P1-P2, P2-P3, P3-P4a, P3-P4b, P3-P4c, P3-P4d, P3-P4) the difference in their performance shows the effect of adding said component to the pipeline. Each

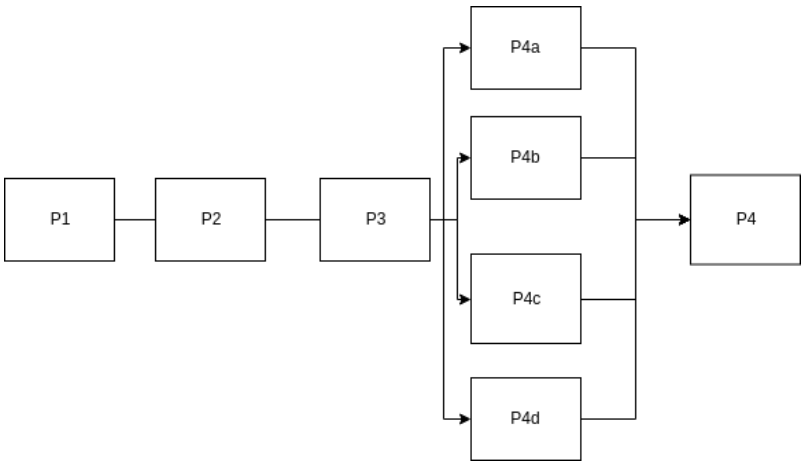

Figure 2: Overview of the stages described in Section 3.2. The stages until P3 are built linearly. Stages after this correspond to the different inference removal algorithms and are considered to be run in parallel. Finally the results are combined and image difference is taken in P4.

components filters the output of the previous one and we can treat it and evaluate it as a binary classification tasks.

The pipeline versions P1-P3 are built linearly by adding components, P4a-P4 are combinations of the 3 inference removal methods (Section 2.5.4).

3.3 Results

We report precision and recall scores for our evaluations (Table 2). Since taking an object happens over time, the ground truth annotations consist of intervals. We do not differentiate between left and right hand events, but we consider parallel annotations. On the other hand our method detects time instants.

We consider all positive predictions that fall into a ground truth interval to be true positives, and all ground truth interval that contain a positive prediction is handled as recalled.

P1 starts with a large number of candidate points measuring in good recall but poor precision scores. Progressing to P3, the candidates are filtered and we can see that precision increases and recall decreases. Comparing P4a-P4 we can see that the combinations of the 3 interference removal components has the best precision and precision and recall values vary when we remove only one type of inference (P4b-P4d).

Our full pipeline (P4) shows reasonable recognition capability when the dataset of the two activities are aggregated (precision 0.51, recall 0.49). When we consider the datasets separately, the "Puzzle" activity dataset has 0.7 precision with 0.68 recall, and "Storytelling with toys" has 0.41 precision with 0.41 recall.

Table 2: Precision and recall scores for each of pipeline variations for our datasets. The pipelines differ on which algorithmic components they contain, explained in Section 3.2.

ADOS-2 activity	Metric	Evaluation pipelines							
		P1	P2	P3	P4a	P4b	P4c	P4d	P4
Puzzle	Precision	0.45	0.62	0.60	0.61	0.65	0.61	0.68	**0.70**
	Recall	0.95	0.92	0.77	0.77	0.75	0.75	0.71	**0.68**
Storytelling with toys	Precision	0.16	0.19	0.21	0.22	0.26	0.22	0.38	**0.41**
	Recall	0.92	0.87	0.62	0.62	0.56	0.62	0.41	**0.41**
All	Precision	0.20	0.25	0.28	0.29	0.35	0.29	0.49	**0.51**
	Recall	0.93	0.89	0.67	0.67	0.62	0.67	0.51	**0.49**

(a) I_{t_1} (b) I_{t_2} (c) Finals mask applied to I_{t_1}

Figure 3: A semi-failure case from our evaluations. (a) and (b) shows the images selected before and after the hand stop time instant respectively. (c) shows the masked image after applying our filters. The object is correctly discovered, but also many other area are also segmented, most of them corresponding to shadow.

Furthermore, we carried out an analysis of the performance of each component by estimating their ability to correctly classify candidates as true or false positives in their respective place in the pipeline.

4 Discussion and outlook

For any '*detector*', low precision and high recall means that it is more general than intended, covering other events as well, while high precision and low recall means that it is too specific, covering only a subset of the targeted events. In the first case, one needs to integrate more knowledge into the pipeline. In the high precision case semi-supervised machine learning, where the annotated samples are collected with the help of the high precision detector on available non-annotated data, may become feasible. This made possible by tuning the individual deep neural network

components providing lower scores with the samples gained via consistence seeking [14].

We can see from the results that our pipeline's intermediate and final performance has higher metric values on the "Puzzle" dataset. We must take into account the ground truth to video length ratio (Table 1) for the two dataset. On "Puzzle" one has 40.8% chance with a random time instance to find a ground truth interval while the same chance is 13.4% for the other dataset. Thus the detection of our target events in the "Puzzle" scenario are easier.

We emphasize that adding each component increases precision, conclusively we are successful in adding knowledge to our pipeline by combining rules and trained deep neural networks. Our components still have shortcomings which we analyzed qualitatively. First, one of the limitations for our method is how we deal with shadows. See the example in Figure 3. This case is not covered by any of our filters at this point. Secondly, we found that there are multiple cases when the detected time instance of hand stop (interpreted as grabbing or releasing) is very close to a ground truth interval but outside of it. In future we consider representing the machine detections as intervals between the before-after images instead of time instances, and use IoU (Intersection-over-Union) measure for evaluations. This would make the detections easier to cluster or drop unlikely ones resulting in less false positives and less multiple detections of the same ground truth interval.

We also minimize the collection of new training samples as proposed in [14]. Our example is the monitoring of the manipulation of unknown objects and detecting "picking up" and "putting down" these objects. In turn the problem treated belongs to the family of '*zero-shot learning*' tasks. First, we considered the general driving principles of the process and transformed them into concrete rules by taking into account trained deep neural networks dealing with hand detection, body pose estimation and motion information extracted from optical flow estimation.

We carried out a detailed analysis of the proposed method on real world scenarios. We found that considerable complexity arises in this relatively simple problem and that it can be overcome by applying rules. We also note that information pieces, e.g., information about depth are missing and could be included. Due to novel developments in deep learning technologies, such as

1. the estimation of 3D distance of objects [10] and that of
2. 3D body configurations [15, 22], as well as the
3. precise estimation of hand configurations from monocamera recordings [20]

our approach on zero shot learning guided by neural-symbolic approach and the belonging self-training capabilities will become more precise and fit the stringent requirements of remote monitoring in the near future.

We plan to conduct a larger scale experiment in the future where we can address the mentioned issues and analyze the algorithm further on more data. We will also try to improve our pipeline by adding depth information estimated from RGB recordings using the listed deep learning algorithms.

Acknowledgements

We are thankful to our supervisor Dr. habil. András Lőrincz and also to Zoltán Tősér for their great advices. Special thanks are due to Szilvia Szeier and Kevin Hartyáni for creating the ground truth annotations and to Judit Fülöp for verifying them. We would also like to thank Dr. Erzsébet Csuhaj Varjú as professional leader. The project has been supported by the European Union, co-financed by the European Social Fund EFOP-3.6.3-16-2017-00002.

Author contributions

A.S. designed the theoretical background, M.Cs. designed the method, M.O. and A.S. helped to further improve upon it. A.S., M.Cs. and M.O. all took part in designing and executing the computational analyses. A.S. and M.Cs. wrote the manuscript.

References

[1] Abdulla, Waleed. Mask r-cnn for object detection and instance segmentation on keras and tensorflow, 2017.

[2] Ahad, Md Atiqur Rahman. *Motion history images for action recognition and understanding*. Springer Science & Business Media, 2012.

[3] Alayrac, Jean-Baptiste, Sivic, Josef, Laptev, Ivan, and Lacoste-Julien, Simon. Joint discovery of object states and manipulation actions. *arXiv preprint arXiv:1702.02738*, 2, 2017.

[4] Bellman, Richard E. *Adaptive control processes: a guided tour*, volume 2045. Princeton university press, 2015.

[5] Bishop, Christopher. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

[6] Dai, Jifeng, Li, Yi, He, Kaiming, and Sun, Jian. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

[7] Du, Pan, Kibbe, Warren A, and Lin, Simon M. Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching. *Bioinformatics*, 22(17):2059–2065, 2006.

[8] Duan, Kun, Parikh, Devi, Crandall, David, and Grauman, Kristen. Discovering localized attributes for fine-grained recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3474–3481. IEEE, 2012.

[9] Everingham, M, Van Gool, L, Williams, C, Winn, J, and Zisserman, A. The pascal action classification taster competition. *International Journal of Computer Vision*, 88:303–338, 2011.

[10] Godard, Clément, Mac Aodha, Oisin, and Brostow, Gabriel J. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017.

[11] He, Kaiming, Gkioxari, Georgia, Dollár, Piotr, and Girshick, Ross. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[12] Ilg, Eddy, Mayer, Nikolaus, Saikia, Tonmoy, Keuper, Margret, Dosovitskiy, Alexey, and Brox, Thomas. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE conference on computer vision and pattern recognition (CVPR)*, volume 2, page 6, 2017.

[13] İlsever, Murat and Ünsalan, Cem. Pixel-based change detection methods. In *Two-Dimensional Change Detection Methods*, pages 7–21. Springer, 2012.

[14] Lőrincz, A, Csákvári, Máté, Fóthi, Áron, Milacski, Z Ádám, Sárkány, András, and Tősér, Z. Towards reasoning based representations: Deep consistence seeking machine. *Cognitive Systems Research*, 47:92–108, 2018.

[15] Mehta, Dushyant, Sotnychenko, Oleksandr, Mueller, Franziska, Xu, Weipeng, Sridhar, Srinath, Pons-Moll, Gerard, and Theobalt, Christian. Single-shot multi-person 3d body pose estimation from monocular rgb input. *arXiv preprint arXiv:1712.03453*, 2017.

[16] Mittal, Arpit, Zisserman, Andrew, and Torr, Philip HS. Hand detection using multiple proposals. In *BMVC*, pages 1–11. Citeseer, 2011.

[17] Ren, Shaoqing, He, Kaiming, Girshick, Ross, and Sun, Jian. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[18] Rohrbach, Marcus, Rohrbach, Anna, Regneri, Michaela, Amin, Sikandar, Andriluka, Mykhaylo, Pinkal, Manfred, and Schiele, Bernt. Recognizing fine-grained and composite activities using hand-centric features and script data. *International Journal of Computer Vision*, pages 1–28, 2015. DOI: `10.1007/s11263-015-0851-8`.

[19] Simonyan, Karen and Zisserman, Andrew. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[20] Spurr, Adrian, Song, Jie, Park, Seonwook, and Hilliges, Otmar. Cross-modal deep variational hand pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 89–98, 2018.

[21] Teo, Ching L, Yang, Yezhou, Daumé, Hal, Fermüller, Cornelia, and Aloimonos, Yiannis. Towards a watson that sees: Language-guided action recognition for robots. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 374–381. IEEE, 2012.

[22] Véges, Márton, Varga, Viktor, and Lőrincz, András. 3d human pose estimation with siamese equivariant embedding. *arXiv preprint arXiv:1809.07217*, 2018.

[23] Wei, Shih-En, Ramakrishna, Varun, Kanade, Takeo, and Sheikh, Yaser. Convolutional pose machines. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4732, 2016.