Human-Object Interaction Detection: 1D Convolutional Neural Network Approach Using Skeleton Data

Plamen Hristov
RCVT Department
Technical University of Sofia
Sofia, Bulgaria
plm@tu-sofia.bg

Dimiter Avresky
IRIANC
Munich, Germany
autonomic@irianc.com

Ognian Boumbarov
RCVT Department
Technical University of Sofia
Sofia, Bulgaria
olb@tu-sofia.bg

***Abstract*—Human-object interaction detection is a somewhat recently emerged scientific topic, which is mainly due to the advent of deep learning algorithms. Most current methods are performed on single images, detecting separately humans and objects, using state-of-the-art pose detection and object detection networks. The networks ease the overall task by allowing for learning of the readily inferred features. When adding the time dimension into the equation, this task becomes more complex, as temporal features between frames have to be taken into account.**

The paper aims to show an approach for detecting human interactions in videos, which utilizes several different methods – YOLOv5 for object detection, CSR-DCF and Kalman Filter for object tracking, and 1D Convolutional Neural Network (1D-CNN) for real-time interaction detection. The overall algorithm is purposed for salient and rigid (modern-solid) objects in mind, positioned in closed-door scenes. The dataset and task are privately defined, that is they are relevant to this work only and cannot be compared to other works. The overall algorithm is tested on a subset of the PKU Multi-Modality Dataset (PKUMMD).

***Index Terms*—1DCNN, skeleton, interaction, object, video**

I. Introduction

Human behavior has been the object of study for decades by many researchers. Most of it concentrates on the person themselves, disregarding their surroundings, which could give us additional data of interest. This data could serve as a context in smart-homes and surveillance of vulnerable people like the elderly or handicapped. Of course, this is only scratching the surface of the area of applications, since object interactions amount to a huge part of human activities. To detect an interaction of any kind, the concept of interaction has to be defined first. When dealing with video data, an obvious definition would be a period from time A to time B where an object is in the proximity of a human joint and moves along the time dimension. Since the start and end frames have to be known preliminarily, this would make the problem a supervised one. However, datasets with such labelling are almost non-existent and the process of labelling is very time-consuming. The target data of this method are RGBD images and 3D skeletons, which are extracted from sensors like Kinect v2. Still, as depth sensing and pose estimation, deep neural networks continue to improve performance-wise, in the future we could see this data extracted by something as compact as a smartphone. This work concentrates on the method itself and the data is assumed to be extracted in real-time. Object locations are also needed, therefore separate algorithms are used for that purpose.

978-1-6654-9550-9/21/$31.00 ©2021 IEEE

II. Related Works

Most of the recent research in literature deals with interaction detection in static images, where each human and object is detected beforehand by a deep neural network, like the Convolutional Neural Network, and the relationship between them both is inferred with another supervised learning algorithm. This relationship is generally represented by a human-action-object triplet.

An example of this common approach is [5], where the authors extract human and object bounding boxes from an image, using ResNet-50 and additionally the body parts using AlphaPose, a multi-person pose estimation network. Finally, the inferred data is passed to their proposed pair-wise feature network (PFNet) where three levels of pair-wise features are extracted – instance level (between human and object), body part level (between body part and object), and semantic level (between human, object and the type of object). The first two features are used for the calculation of an action probability, based on appearance, while the last one is used for action probability with a semantic prior. The final HOI score is a product of those two probabilities. [5] achieve 52.8 mAP on the V-COCO dataset, which is commonly used as a benchmark, and 20.05 mAP on the default configuration of the full HICO-DET dataset.

While this works well for static images with conspicuous context, many non-evident interactions can only be found when taking into account the time dimension.

Chiou et al. [3] have taken the task further by improving the conventional method of extracting and combining features of objects and people in still frames into a variant fit for modelling videos. First, they sample videos for keyframes

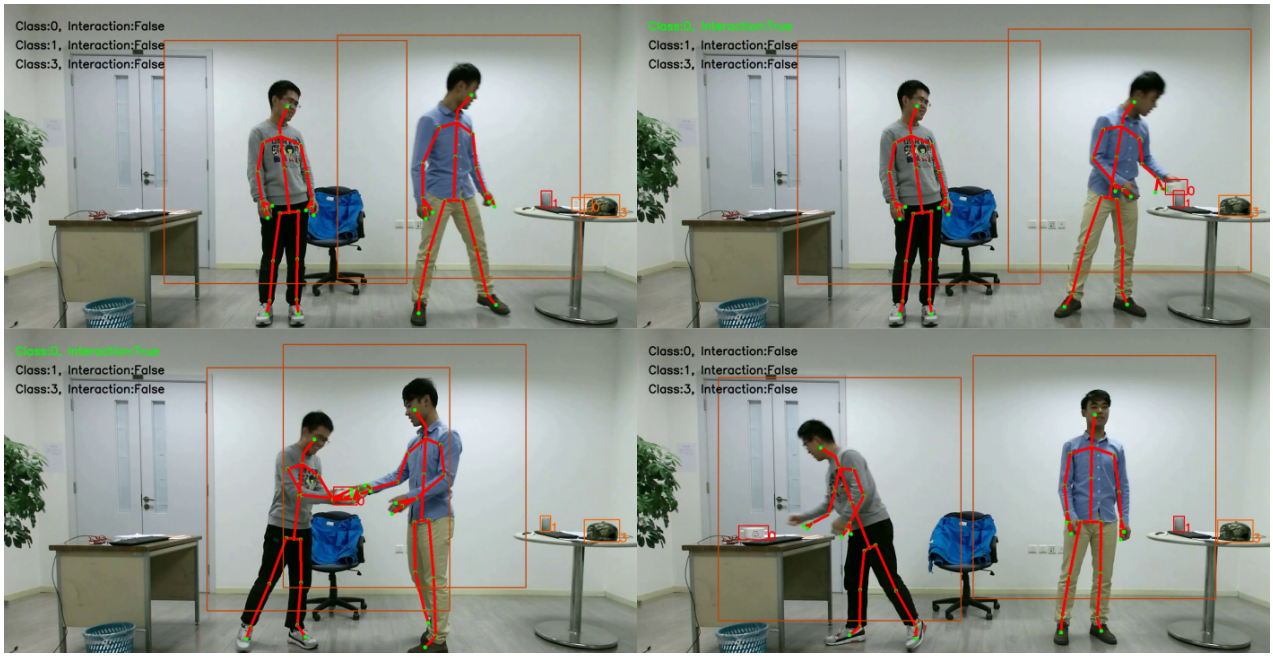

Fig. 1. Sampled frames from a ground-truth labelled interaction segment. An interaction occurs the moment the object moves and is close to a person. Objects in red boxes are detected by YOLOv5, objects in yellow are tracked with CSR-DCF. People in the scene are tracked with a skeleton-pose. The 640x640p boxes around the poses are the region where YOLOv5 is used. The text on the top left shows the current object in the scene and if they are interacted with.

with a rate of 1Hz. The neighboring frames of these key-frames form segments and are then passed to a 3D-CNN to extract Spatio-temporal features for each segment. Second, trajectories get calculated for each key-frame for pooling these extracted feature maps, which are then average pooled along the time-dimension to get correctly localized visual features. Thirdly, three binary masks, corresponding to the human pose, human bounding box, and object bounding box, are calculated and then merged for each human-object pair. Then, they are downsized and passed to two 3D-CNN layers to extract pose-masking features. Finally, the visual and pose features, as well as the trajectories, are concatenated and classified by linear layers. The output size is equal to the number of interaction types.

Chiou et al. [3] achieve 17.6 mAP on their proposed benchmark dataset VidHOI, which consists of 7122 videos, 70 video hours, and 7.3 million annotated frames.

What the above approaches have in common is that they use features extracted from deep learning methods. Other methods are based on the position of objects and the human joints and limbs, which are extracted from depth sensors, such as the Kinect v2, or deep learning models. They, however, often differ in their presented concept of interaction.

Meng et al. [6] use the extracted skeletons from Kinect v2 for each frame and detect the object in it, using the LOP (Local Occupancy Pattern) algorithm. Each frame is represented by all the joint differences in the skeleton. The position of the object is also seen as a joint. A window of a specified number of frames slides over every video, where for each segment of frames, a feature vector is calculated by concatenating all the joint differences from the frames, and then classified by a Random Forest model. The authors achieve an accuracy of 75.8% on the ORGBD dataset, in the cross-subject setting. The aforementioned work deals with classifying whole videos for their observed type of interaction, but could be potentially fitted for online interaction detection on a per-frame basis.

Bruckschen et al. [7] define a prerequisite for an interaction as an intersection between the bounding boxes of an object and a human hand. The boxes are inferred using R-CNN and OpenPose, respectively. Since using only this rule could cause false positives because of occlusions or inaccurate detections, the authors define a likelihood function, using the durations of observed human-object interactions in a university setting, as training data. They further find that the data is best fitted with a gamma probability density function with k = 5 and Θ = 0.9. The cumulative distribution function calculated from this probability density function is used for predicting the likelihood of interaction, depending on its duration. Human-object interaction is then defined as ground truth if at least 5 seconds have elapsed between frames where there are bounding box intersections. The threshold of 5 seconds was selected because it corresponds to a likelihood of 50%.

Bruckschen et al. [7] test their method on a self-collected dataset, comprising 195 human-object interactions, totaling over 27 minutes of video data. They achieve precision and recall rates of 0.82 for a minimum likelihood threshold of 0.22. Fang et al. [4] propose a HOI method in videos for robot understanding. They exploit RGB, depth, and skeleton data from a Kinect v2 sensor. First of all, the joint position of a person's right hand is used to crop an area around it

from the RGB frame – this is passed for object detection to a YOLO model, trained with custom data. Secondly, the person holding an object is segmented using the depth frames, and the intersection between their bounding boxes is used to define the belonging of the object. Finally, a Kernelized Correlation Filter is used for tracking an object between frames. When the object is lost, the YOLO model again is used for detection.

III. Proposed Method

The method starts with preprocessing the whole data, in order to find the ground truth interaction segments. This means finding the objects using an external deep learning model in the video and tracking them. The videos, recorded by Kinect v2, have a resolution of 1920x1080 pixels and a refresh rate of 30Hz. Object locations are detected or tracked over the course of the videos. Combined with the body pose information, provided by Kinect v2, they are transformed into time series, which represents the movement of the human limbs with relation to the object position over time. The 1D CNN is often used for data, which can also be represented as time series (like sensor readings, audio and text), therefore it was chosen for the task of score prediction, based on a sequence.

A. Definition of an interaction

A prerequisite for assessing ground truth interaction segments in a dataset is knowing the start and end times for those segments beforehand. Labelling whole videos is time-consuming and, if done by a group of people, it could result in different subjective interpretations due to the weakly defined nature of the task. Instead, a simple rule for a segment is proposed – the start time is set when an object moves from its previous location, which assumes that all objects are stationary at the beginning of a video. Consequently, the end time has to be declared as the moment the same object stops moving. An additional problem arises from this presumption. This is the definition of stationariness for an object, for which reason two variables have to be declared – a threshold for a position change between frames, which marks the start of an interaction and a number of frames for which the object keeps its position under this threshold – marking the end of the interaction. (Fig. 1) shows an example of the proposed concept in a video.

B. Object Detection

When testing in real time there is no information about the objects, they need to be found for each frame. For that reason, a pretrained model of YOLOv5 [1] is used. YOLO (You Only Look Once) is a recent family of object detection models trained on the COCO [2] dataset (Common Objects in Context). This dataset consists of around 330 000 images, 200 000 of which are labelled, containing 1.5 million object instances, grouped into 80 different classes. It is one of the most common benchmarks for object detection and recognition methods. The output of YOLOv5 is a list of bounding box predictions for each object in the image, with a format of x, y, width, height, which are normalized between 0 and 1, and a confidence score for each prediction – between 0 and 1. There are several available YOLOv5 pretrained models with different sizes, their precision increasing with size, but their inference time decreasing. In this work, the YOLOv5x (extra-large) model is selected.

C. Transfer Learning

When dealing with objects that the model is not trained for, or has difficulty detecting due to resolution size or uncommon object pose, transfer learning has to be performed for them. This is done by sampling frames from the video dataset, which then are used for training. In this work, every 15th frame is selected. Since YOLOv5x was trained on images with a resolution of 640x640 pixels, the sampled frames are cropped from the RGB frame, while fitting inside the person in the scene.

D. Human Tracking

A depth sensor like Kinect v2 exploits a closed-source algorithm for real-time pose estimation of the human body, which is represented as a set of points in 3D space – also commonly known as a skeleton. These points correspond to joints of the human body and for Kinect v2 they are 25. They are extremely useful when applied to Human Activity Recognition tasks, as the feature extraction step is significantly reduced. For each RGB frame, a 640x640 pixel bounding box is extracted in a way that the skeleton of the tracked person is fitted inside, disregarding the lower body joints from the knee and below, as only hands are used in the proposed scenario. This avoids having to predict the objects on the full-frame, as we are only interested in the objects in case they are in proximity to a person.

E. Object Tracking

YOLOv5 achieves very high performance for real-time prediction, even in cheaper GPUs. However, it does so on a per-frame basis, with no tracking functionality. In failure to detect an object, its last known position could be used for tracking it, hence a CSR-DCF tracker [8] is introduced to fill the gaps between measurements and specifically, the algorithm provided by the OpenCV library [9]. Another problem in detection is occlusion, where an assumption has to be made on where the object is situated, based on its previous position, speed, and acceleration. A linear Kalman Filter is used in this regard.

The Kalman Filter is an algorithm for predicting states of variables over time, given prior measurements of those variables for each timeframe. Also, process and measurement noise are taken into consideration. The state vector components in the proposed case contain position, velocity, and acceleration of the bounding box coordinates of the object.

F. Training and Testing

After finding the ground truth data, the training step can begin. For this, a 1DCNN with a sliding window approach will be introduced. The network architecture (Fig. 2) comprises two 1D convolutional layers with 76 filters and kernel sizes

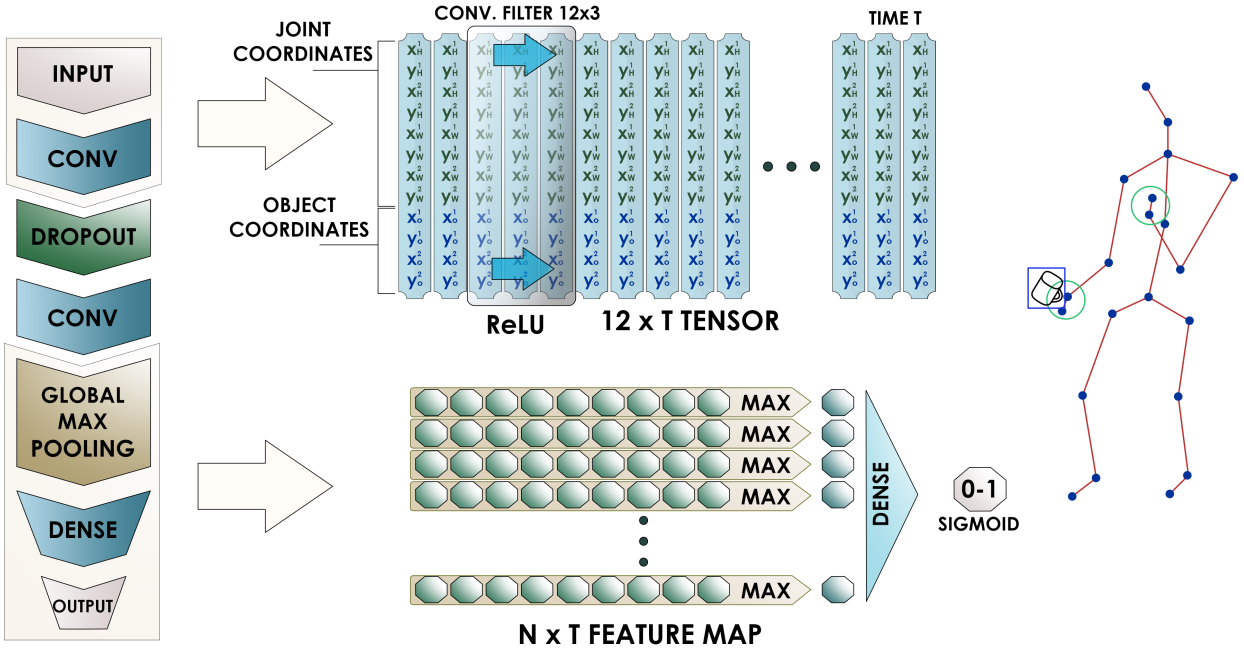

Fig. 2. Overall algorithm. A tensor holding 30 frames is passed to a neural network, containing several layers. Each frame is a vector, consisting of 8 coordinate values for the body joints and 4 coordinates for the object. The output of the network is a single probability value for an interaction of the last frame (based on its 29 preceding frames)

of 3 and 5, with a spatial dropout layer between them. Spatial dropout is a regularization method which discards entire feature maps along the time dimension if neighboring maps are strongly correlated. The dropout rate is set to 0.5 (50% of maps are dropped).

Since only the local context is needed for a sequence, a limit of 30 time steps is set. This way, the 1DCNN will return a label for a frame only and not for the entire sequence. Each time step is a vector, which contains the 3D coordinates of both hand and wrist joints of the skeleton and the 2D bounding box of an object. The problem of multiple objects, as well as people in the scene, is solved by creating a separate stream for every object-person pair.

Each sequence is split into windows of 30 frames and passed to the network. The output is exactly one vector, which is then connected to a dense neuron. This neuron generates a probability of interaction with a sigmoid activation function.

Sigmoid(1) is a mathematical function that exists between 0 and 1, therefore, it is generally used for predicting a probability.

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \tag{1}$$

IV. Experiments

The following experiments are done on a subset of the PKU-MMD dataset [10]. It is a dataset used for the regression task of action detection in videos, where the start and end frames of every action exist as ground truth and have to be found when testing. The full dataset consists of RGB videos, which contain many such action segments and for which the body pose is extracted for every person (skeleton). There also exist three views for the recorded scene - Left, Middle, Right. In this work, only the Middle one is used.

A. Transfer Learning for Object Detection

In order to maximize the detection capabilities for this dataset, the YOLO model has to be trained further on frames from the videos. Frames were sampled from each video with a 15Hz sampling rate and an area of 640x640 pixels was cropped around the person in the scene. The total number of frames totals 7100. Each frame was manually labelled for the objects that are present in it. The objects chosen for training were: “cup“, “bowl“, “smartphone“, “baseball hat“.

The newly formed dataset was split into 85:15 for training and validation.

The YOLO library contains a script for training on new or pre-trained models, which was used in our case. The training was done over 12 epochs, on a server with a GTX 1080 Ti GPU and i7-6700K CPU.

B. Dataset Creation and Labelling

As the full dataset contains non-object related action segments, such were discarded. The remaining segments are: “drink water”, “eat meal/snack”, “giving something to other person”, “answer phone”, “playing with phone/tablet”, “put something inside pocket”, “take out something from pocket”. Some of the segments happen consecutively and so they are combined into one.

Then, all the segments are padded with extra 100 frames in the beginning and end from their corresponding videos, in order to capture the “circumstances” before and after an interaction (reaching for an object and moving away from

it). The resulting dataset contains 624 video segments with skeleton data for every frame. Every segment was labelled for its objects with the object detection/tracking methods and their related interactions. The selected interaction parameters were:

- 3 pixel movement threshold
- 3 frames for an interaction start for this threshold
- 5 frames for an interaction end for this threshold

The new segments are sequences of frames, where each frame contains skeleton and object coordinates and whether an interaction exists (0 or 1). Coordinates of each object were concatenated with the skeleton coordinates for every frame, in order to construct a sequence of timesteps, which can be passed to a 1D CNN.

C. Training a 1D CNN model on the interaction dataset

The dataset is split into 70:20:10 parts for training, validation, and testing, respectively. Each sequence is divided into windows of 30 frames with a stride of 10. The network is then trained on the windows for 10 epochs. The chosen optimizer is Adam, with a learning rate of 0.01. Fig. 3 shows the prediction results for a subset of the full data.

Accuracy, recall, precision and F1 score were chosen as performance metrics and they were calculated based on thresholding the final results with different values. Optimal scores are reached with a threshold of 0.2. (Table. I)

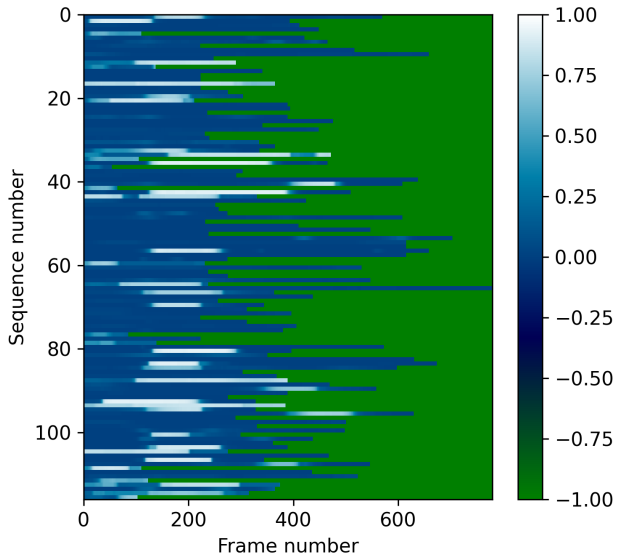

Fig. 3. Prediction results for a subset of the full data, before thresholding.

TABLE I
PREDICTION SCORES

Threshold	Accuracy	Recall	Precision	F1 Score
0.5	0.922	0.627	0.875	0.731
0.35	0.928	0.669	0.878	0.759
0.25	0.934	0.708	0.879	0.784
0.2	0.934	0.732	0.855	0.789

V. CONCLUSION

The results achieved show great promise and a possibility for future real-time implementations of this algorithm.

A few setbacks, however, exist. For example, the false negatives of the object detections (missed detections) force us to fill the gaps by using prediction methods that don't always correctly track the lost object. Assuming a state-of-the-art method for object detection like YOLO is used, this could be fixed by increasing the resolution of the dataset.

Another problem is obtaining coordinates for an occluded object – predicting the position based on previous data is a simple, but a limited, solution. Upgrading the system to capture several views of a scene would solve this issue and even provide more accurate data.

ACKNOWLEDGEMENT

This work was supported by the contract KP-06-H37/8 from 06.12.2019 for research project: "Inference algorithms for semantic knowledge extraction based on deep architectures for context-aware holographic communication" of the Bulgarian Research Fund of the Ministry of Education and Science, Bulgaria.

REFERENCES

[1] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, Ayush Chaurasia, TaoXie, Liu Changyu, Abhiram V, Laughing, tkianai, yxNONG, Adam Hogan, lorenzomammana, AlexWang1900, Jan Hajek, Laurentiu Diaconu, Marc, Yonghye Kwon, oleg, Francisco Ingham. (2021). ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations (v5.0). Zenodo. https://doi.org/10.5281/zenodo.4679653

[2] Lin TY. et al. (2014), Microsoft COCO: Common Objects in Context, In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48

[3] Meng-Jiun Chiou, Chun-Yu Liao, Li-Wei Wang, Roger Zimmermann, and Jiashi Feng. 2021. ST-HOI: A Spatial-Temporal Baseline for Human-Object Interaction Detection in Videos. In Proceedings of the 2021 Workshop on Intelligent Cross-Data Analysis and Retrieval (ICDAR '21). Association for Computing Machinery, New York, NY, USA, 9–17. DOI:https://doi.org/10.1145/3463944.3469097

[4] Fang Z, Yuan J, Magnenat-Thalmann N. Understanding human-object interaction in RGB-D videos for human robot interaction. Paper presented at: Proceedings of Computer Graphics International; 2018. p. 163–167

[5] Liu, H., Mu, TJ., Huang, X. Detecting human—object interaction with multi-level pairwise feature network. Comp. Visual Media 7, 229–239 (2021). https://doi.org/10.1007/s41095-020-0188-2

[6] Meng Meng, Hassen Drira, Mohamed Daoudi, Jacques Boonaert. Human-Object Interaction Recognition by Learning the distances between the Object and the Skeleton Joints. Face and Gesture, 2015, Ljubljana, Slovenia. ffhal-01703222 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] Bruckschen L., Amft S., Tanke J., Gall J., Bennewitz M. (2019) Detection of Generic Human-Object Interactions in Video Streams. In: Salichs M. et al. (eds) Social Robotics. ICSR 2019. Lecture Notes in Computer Science, vol 11876. Springer, Cham. https://doi.org/10.1007/978-3-030-35888-4_11

[8] Alan Lukežič, Tomáš Vojíř, Luka Čehovin, Jiří Matas and Matej Kristan. "Discriminative Correlation Filter with Channel and Spatial Reliability." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[9] Bradski, G. The OpenCV Library. *Dr. Dobb's Journal Of Software Tools*. (2000)

[10] Liu, C., Hu, Y., Li, Y., Song, S., & Liu, J (2017). PKU-MMD: A Large Scale Benchmark for Continuous Multi-Modal Human Action Understanding. ArXiv, abs/1703.07475.