Human Recognition and Behavior Analysis from RGB Camera

INTERNSHIP FINAL PRESENTATION

TOMASZ STAŃCZYK (6209867)

Motivation

- Human behavior analysis
- Well-being importance
  - Schools [1]
  - Kindergartens [2]
  - Elderly centers [3]
- Automation advantages
  - Saving the time of teachers/caregivers
  - Continuity of the analysis, capturing all (important) events

Problem statement

- Automatic
  - Identification of the subject
  - Recognizing the performed action with the time of it occurrence
  - Recognizing the subject's emotion

- System consisting of particular components solving the problem

Research questions – part 1

(1) How such a system can be created and can it be properly integrated?

(2) Can the while system run in real-time?

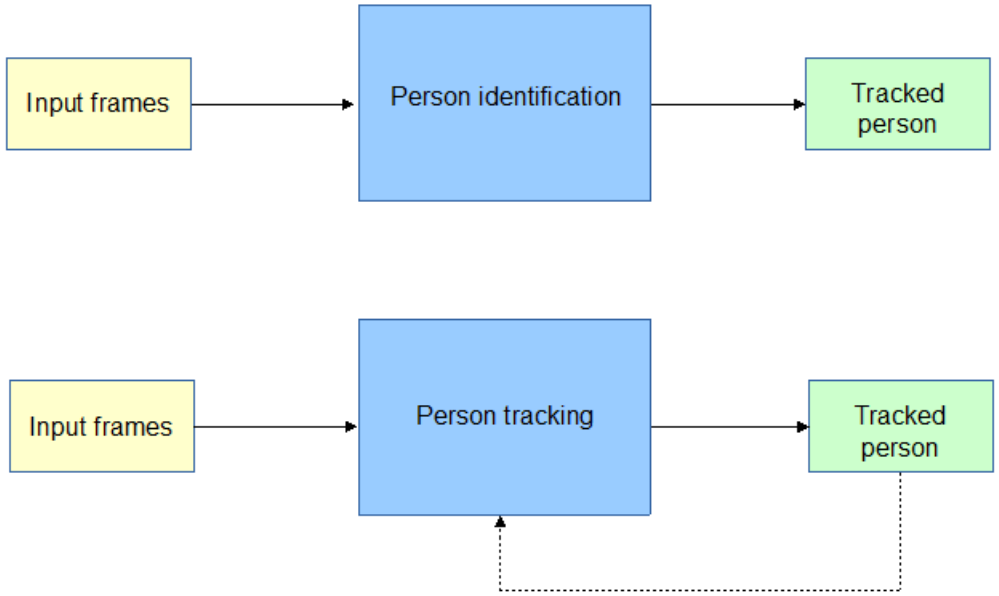
(3) Is high-performance PC hardware needed for this system?

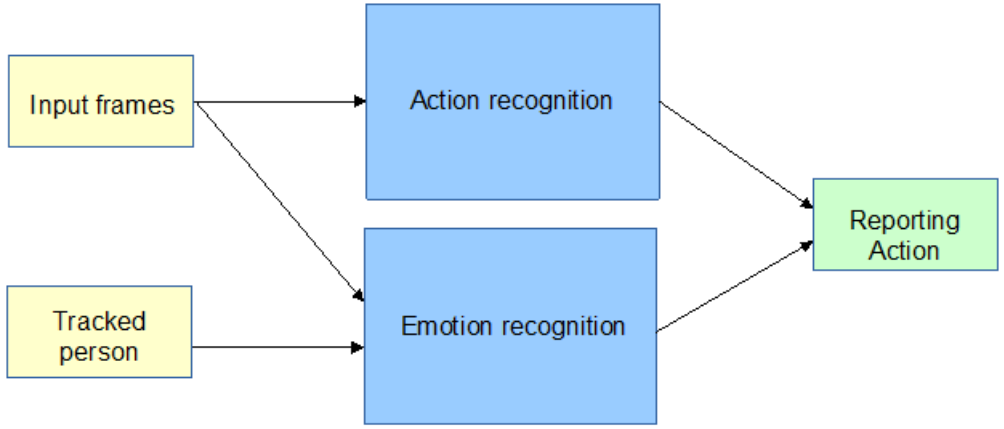
Main approach

- Using several state-of-the-art Computer Vision techniques
- Integrating these into one application
- RGB video stream as the only modality

Main approach

Initial overview of the considered system architecture:





Main approach

- Initial orientation over required solutions/components
- No suitable ready-to-use solution for action recognition found
- A good action recognition component working relatively fast is considered for the development

Research questions – part 2

(4) Can the required and ready to use action recognition component be created using existing solutions?

(5) How good action prediction accuracy score can be reached using existing solutions?

(6) Can this action recognition component run in real time?

Outline

- Part I – Action recognition
  - Research questions (4) and (5)
- Part II – Integrated system
  - Research questions (1), (2), (3) and (6)

(1) How such a system can be created and can it be properly integrated?

(2) Can the while system run in real-time?

(3) Is high-performance PC hardware needed for this system?

(4) Can the required and ready to use action recognition component be created using existing solutions?

(5) How good action prediction accuracy score can be reached using existing solutions?

(6) Can this action recognition component run in real time?

Part I – Action recognition

State-of-the-art

Action recognition datasets

- UCF101 [5]
- HMDB51 [6]
- Sports-1M [7]
- Kinetics-400 [8]
- Kinetics-skeleton [11]
- NTU RGB+D [4]

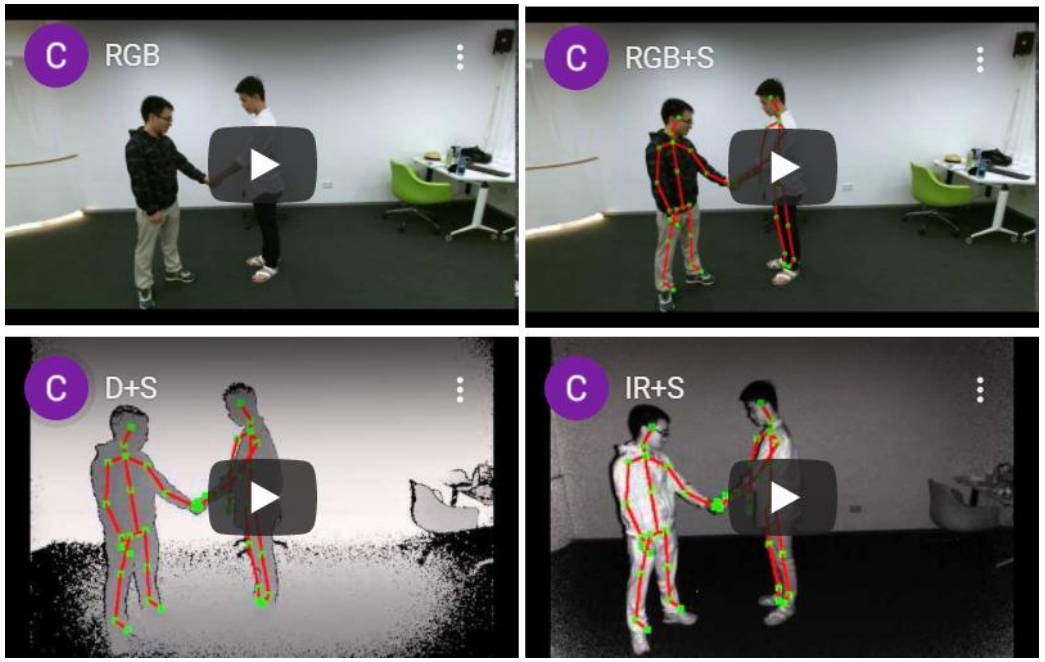
State-of-the-art

NTU RGB+D dataset [4]

- 60 fine-grained action classes
  - daily actions
  - mutual actions
  - medical conditions
- E.g. Drink Water, Eat Meal, Read, Write, Put On a Shoe, Put On Glasses
- Focus on subjects
- Multiple subjects and multiple views are included
- Data samples in several formats,
  - 3D joint data registered by Kinect v2 sensors
  - RGB videos with resolution of 1920x1080

State-of-the-art

NTU RGB+D dataset [4]



State-of-the-art

- Action recognition existing architectures
  - Recurrent convolutional networks [13]
  - Temporal segment networks [14]
  - Spatio-temporal convolution [16]
  - 3D convolutions [17], [18], [19], [20], [21]
  - …
- Utilizing UCF101, HMDB51, Sports-1M or Kinetics-400 datasets
- No performance results on the desired NTU RGB+D dataset

State-of-the-art

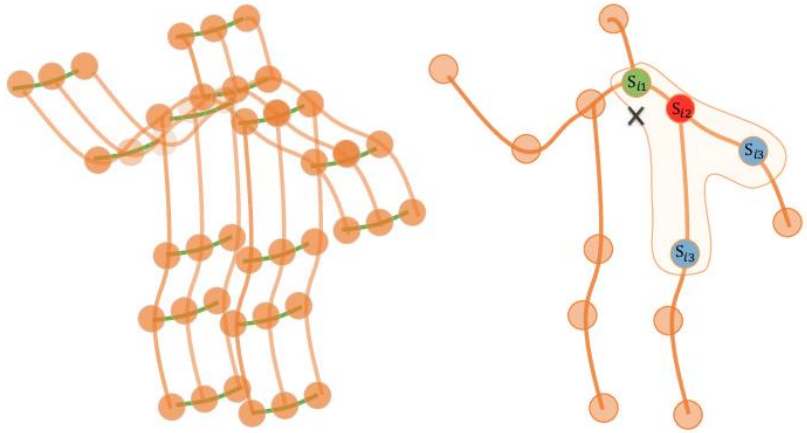
- Skeleton-based action recognition
  - Invariance to viewpoint and appearance [22]
  - Context can be incorporated at posterior processing stages
  - Great performance reported for Deep Learning in skeleton-based action recognition [23], [24]

- Exisiting graph-based architectures:
  - Spatio-Temporal Graph Convolutional Networks [26]
  - Two-Stream Adaptive Graph Convolutional Networks [27]
  - Multi-Stream Adaptive Attention-Enhanced Graph Convolutional Networks (MS-AAGCN) [24]
  - …

State-of-the-art

MS-AAGCN (Multi-Stream Adaptive Attention-Enhanced Graph Convolutional Networks) [24]

- Processing human pose skeletons as graphs
- Learning the graph topology and adapting to various data samples
- Spatial-temporal channel attention module - enabling focusing important
  - joints
  - frames
  - Features



- Skeleton based action recognition with accuracy up to 96.2% on NTU RGB+D (3D)
- …yet up to 37.8% on Kinetics-Skeleton (2D)

Figure courtesy of Shi et al. [24]

Methods

- Performance difference on two datasets: NTU RGB+D and Kinetics-skeleton (accuracy of 96.2% vs. 37.8%)
- Two possible reasons:
  - Dimensionality difference (3D vs. 2D)
  - Type of dataset
- NTU RGB+D: restricted environment, stable camera(s), human subjects always present at the scene
- Kinetics-400 (base for Kinetics-skeleton): large number of YouTube videos, unrestricted environment, unstable camera, human subjects possibly absent from the scene
- A few studies showing informativeness of 2D skeleton data [38], [39], [40]

Methods

Hypothesis:

An architecture with good performance on 3D joints, which is capable of handling 2D joint data, also performs relatively well (meaning only slightly worse) when it is trained and evaluated on a 2D joint dataset instead, yet with type and data analogous to the 3D joint dataset used.

- Using RGB videos coming from the NTU RGB+D dataset and extracting 2D skeletons
- Applying Lightweight OpenPose [41]
  - Faster version of the human pose estimation toolbox OpenPose [12]
- Processing each sample separately
- Saving the output files in the same format as Kinetics-skeleton data

- New dataset created, referenced as NTU 2D

Methods

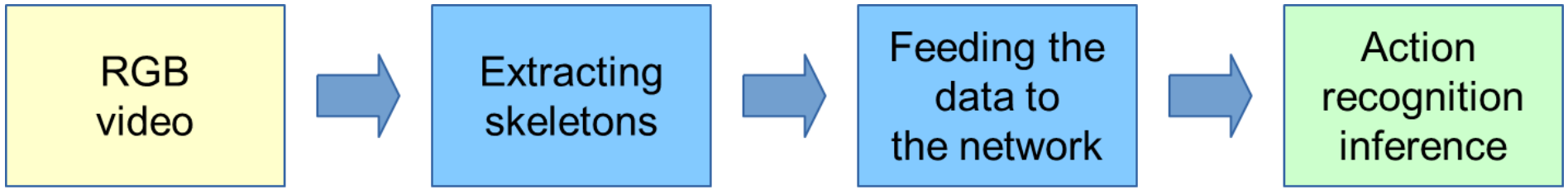
Developing action recognition component

- Using and expanding existing MS-AAGCN implementation [37]
- Performing adaptations where needed
  - Appropriate data pre-processing (with approach for the NTU dataset)
  - Adjusting output layer of the network
- Training the architecture on NTU 2D dataset, with the variations included
  - Generating bone data, motion data (following MS-AAGCN authors' work)
  - Fusing scores

Methods

Developing action recognition component (continued)

- Enabling running predictions on a single custom sample
- Adding Lightweight OpenPose to the prediction mechanism
  - Analogous to creation of NTU 2D
- Creating the whole pipeline:



Experiments

Training multiple models on NTU 2D dataset

- (1) Joint data without attention (xview)
- (2) Joint data without attention (xsub)
- (3) Joint data with attention (xview)
- (4) Joint data with attention (xsub)
- (5) Bone data without attention (xview)
- (6) Bone data with attention (xview)
- (7) Joint motion data with attention (xview)
- (8) Bone motion data without attention (xview)
- (9) Joint padded data with attention (xview)
- (10) Bone padded data without attention (xview)

Experiments

Creating custom samples

- Recording videos with performed actions
  - Performing single subject class actions from NTU RGB+D and NTU 2D datasets
  - Imperfect actions
  - Potentially more behavior included
- Extracting Skeletons with Lightweight OpenPose
- Using the samples for additional evaluation

Experiments

Creating custom samples

Hand waving

Clapping

Sitting down

Experiments

(First) Model evaluation

(1) Joint data without attention (xview)

(2) Joint data without attention (xsub)

Model	Custom samples top 1 accuracy	Custom samples top 5 accuracy	NTU 2D top 1 accuracy	NTU 2D top 5 accuracy
(1)	55.21%	87.12%	87.56%	99.03%
(2)	52.76%	84.05%	82.31%	97.75%

Experiments

(The best) Model evaluation

(3) Joint data with attention (xview)

(9) Joint padded data with attention (xview)

(5) Bone data without attention (xview)

(10) Bone padded data without attention (xview)

Model (fused scores)	Custom samples top 1 accuracy	Custom samples top 5 accuracy	NTU 2D top 1 accuracy	NTU 2D top 5 accuracy
(3)+(5)+(9)+(10)	69.94%	87.73%	93.46%	99.78%
0.5*(3) + 0.5*(5) + 1*(9) + 0.8*(10)	70.55%	87.12%	93.26%	99.78%

Experiments

Model predictions 0.5*(3) + 0.5*(5) + 1*(9) + 0.8*(10)

(Reading)

1. Reading
2. Writing
3. Playing with phone/tablet
4. Typing on a keyboard
5. Taking a selfie

(Take off jacket)

1. Wear jacket
2. Take off jacket
3. Tear up paper
4. Reading
5. Brushing hair

(Drinking water)

1. Drink water
2. Make a phone call/answer phone
3. Wear on glasses
4. Eat meal/snack
5. Taking a selfie

(Touch chest (stomachache/heart pain))

1. Wear on glasses
2. Rach into pocket
3. Sneeze/cough
4. Eat meal/snack
5. Take off glasses

Discussion

- Accuracy difference when testing on NTU 2D test spilt and custom samples
- 70.55% (custom samples) vs. 93.26% (NTU 2D) for the best performing model
- Possibly due to different nature of the datasets
  - Camera view angles
  - Relative size of the subject
- Imperfect actions inside custom samples
  - More behavior (preceding and proceeding the main action)
- Fusing more models
- Incorporating the context information

Conclusion

Answering the following research question:

(4) Can the required and ready to use action recognition component be created using existing solutions?

- An action recognition component developed
  - taking RGB video (from a file or camera stream),
  - extracting human pose skeletons,
  - feeding the skeleton subsequences to a graph convolutional network
- Using solutions for human pose estimation and 3D skeleton-based action recognition
- Applying necessary adaptations and deriving suitable dataset
- Ready-to-use action recognition component is obtained

Conclusion

Answering the following research question:

(5) How good action prediction accuracy score can be reached using existing solutions?

- On the derived dataset test split:
  - top 1 prediction accuracy up to 93.46%
  - top 5 prediction accuracy up to 99.78%
- On the custom samples with imperfectly performed actions and additional behavior included:
  - top 1 prediction accuracy up to 70.55%
  - top 5 prediction accuracy up to 88.96%

Part II – Integrated system

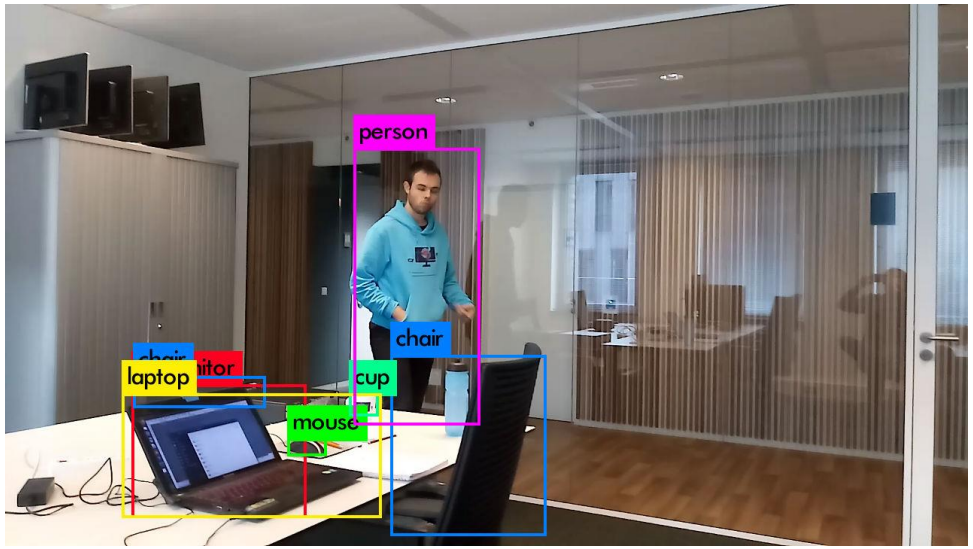
State-of-the-art

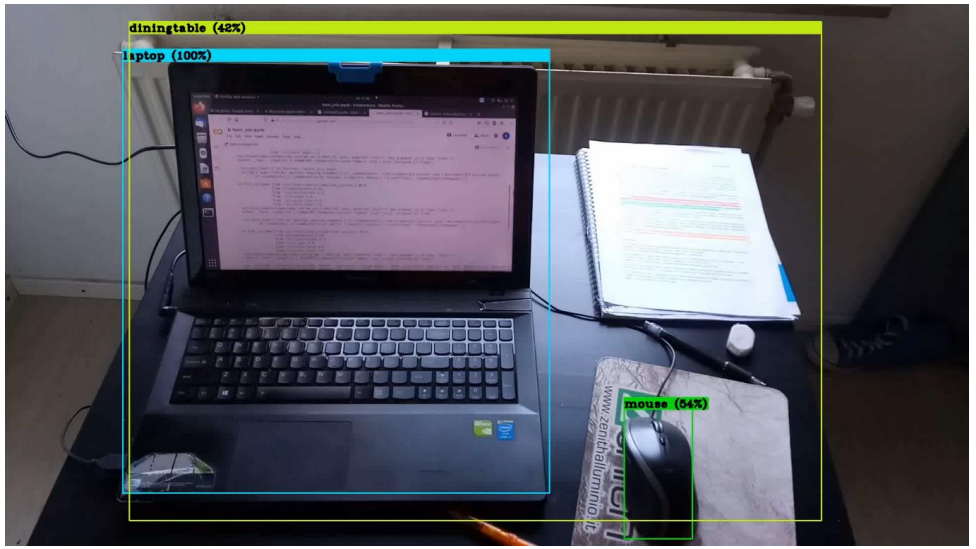
Object detection

- YOLO (You Only Look Once) [44]
- Fast and accurate object detection results
- Bounding boxes with prediction label
- Pre-trained model available [45]
- PyTorch implementation chosen [47]
  - Easy to use and integrate

State-of-the-art

Object detection – examples

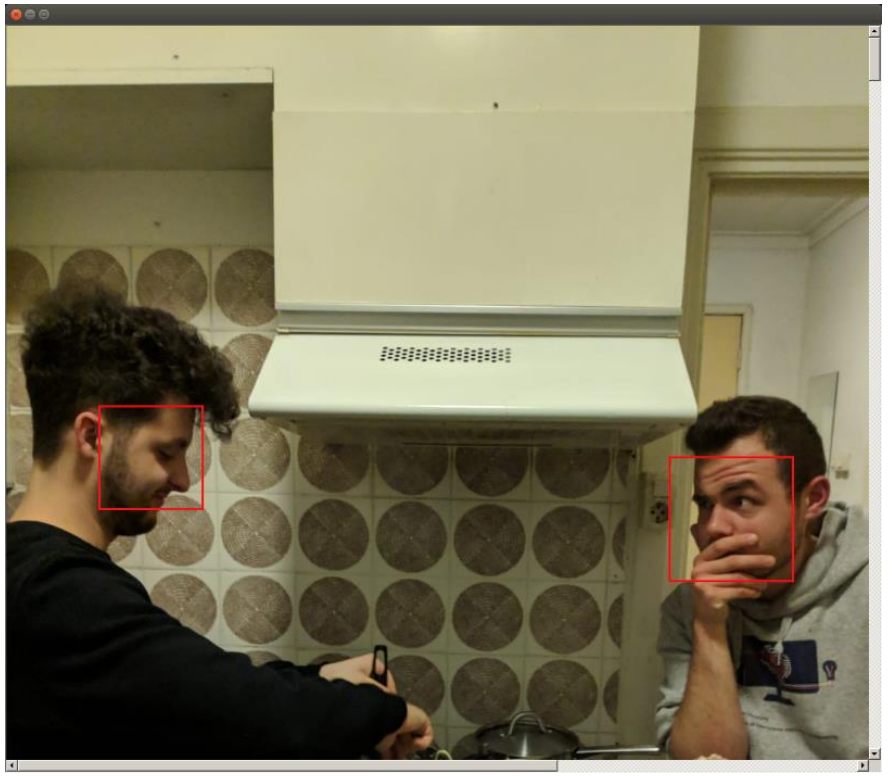




State-of-the-art

Face detection

- Dlib library, Machine Learning Tooling [49]
- Convolutional Neural Network (CNN) based face detector [48]
- Pre-trained model with network weights available [48]
- Accurately localizing faces from frontal, left and right side



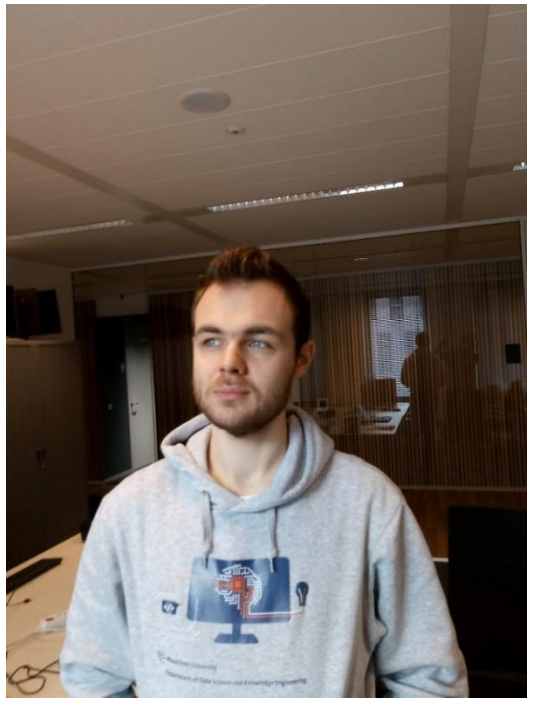
State-of-the-art

Face recognition

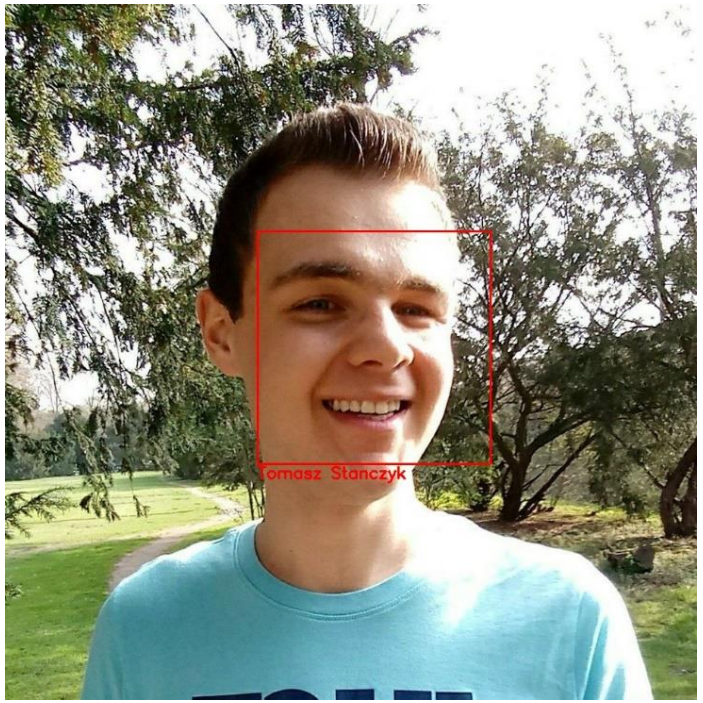
- Dlib face recognition tool [51]
- Pre-trained model with network weights publicly available [51]
- Single photograph of a person sufficient for creating a facial descriptor (encoding)
- Accurate prediction on another, not related photograph with the same person
  - based on the created descriptor

State-of-the-art

Face recognition – examples



Descriptor photo

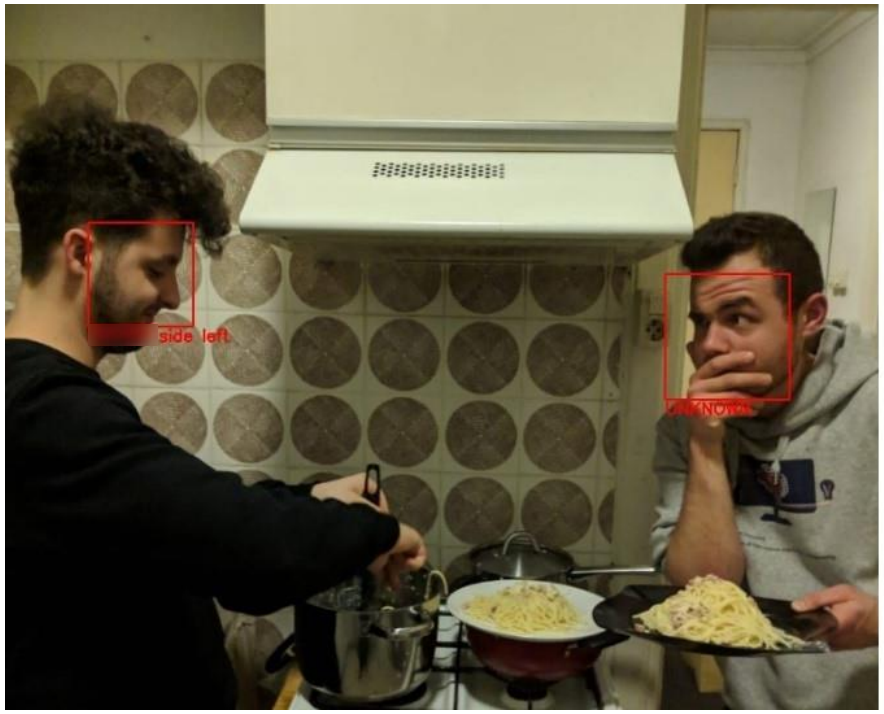


Recognition photo

State-of-the-art

Face recognition

Descriptor photo



Recognition photo

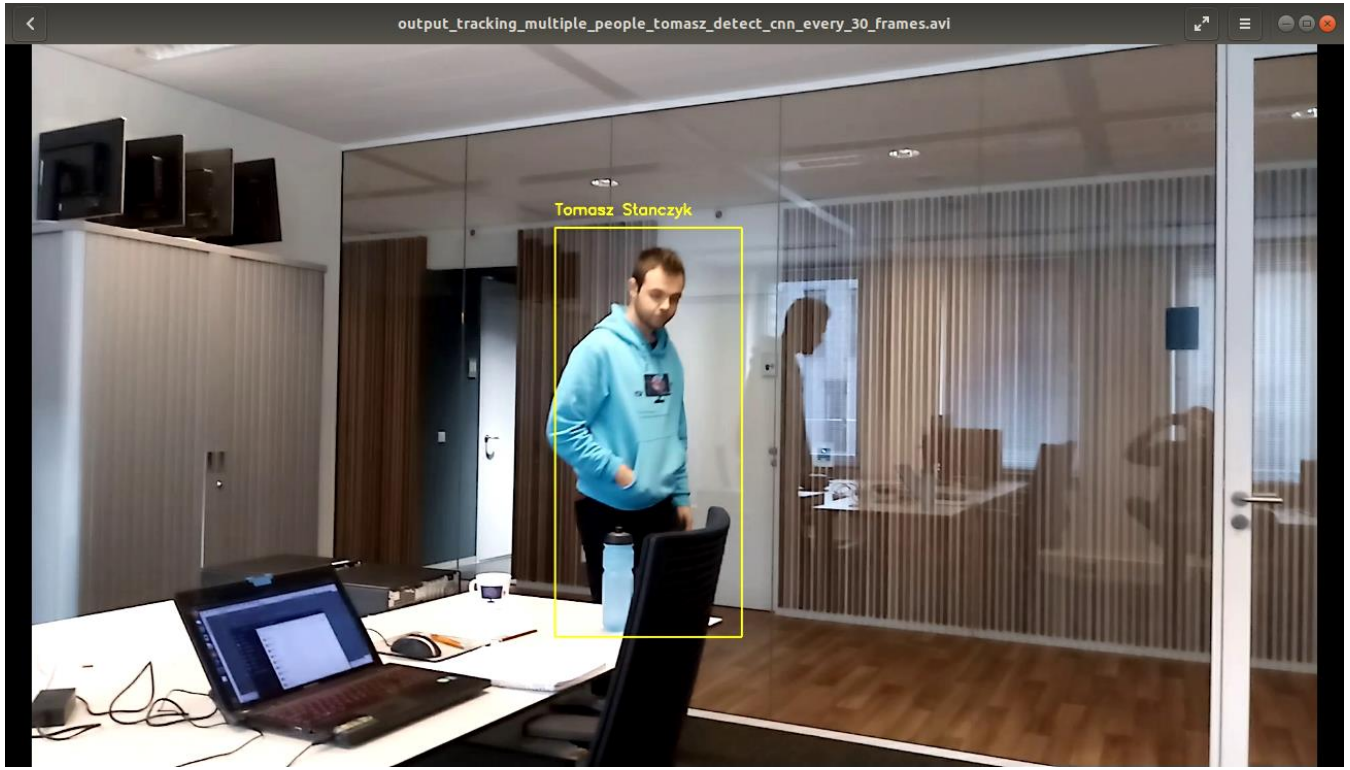
State-of-the-art

Object tracking

- Dlib's implementation of correlation tracker [55]
- Requiring an initial position of the object intended to track in one frame
- Computing analogous location in subsequent frames.
- Simple to use and integrate with other solutions

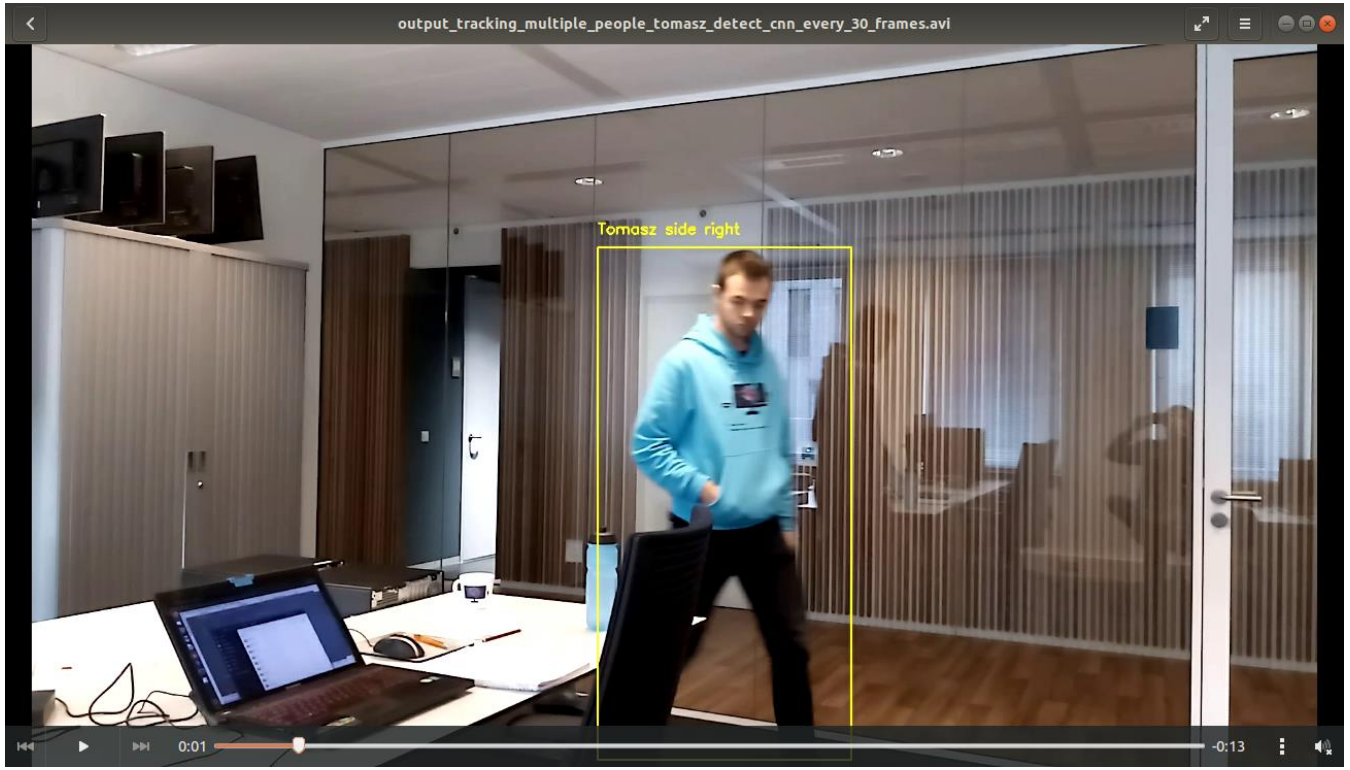
State-of-the-art

Object tracking – example



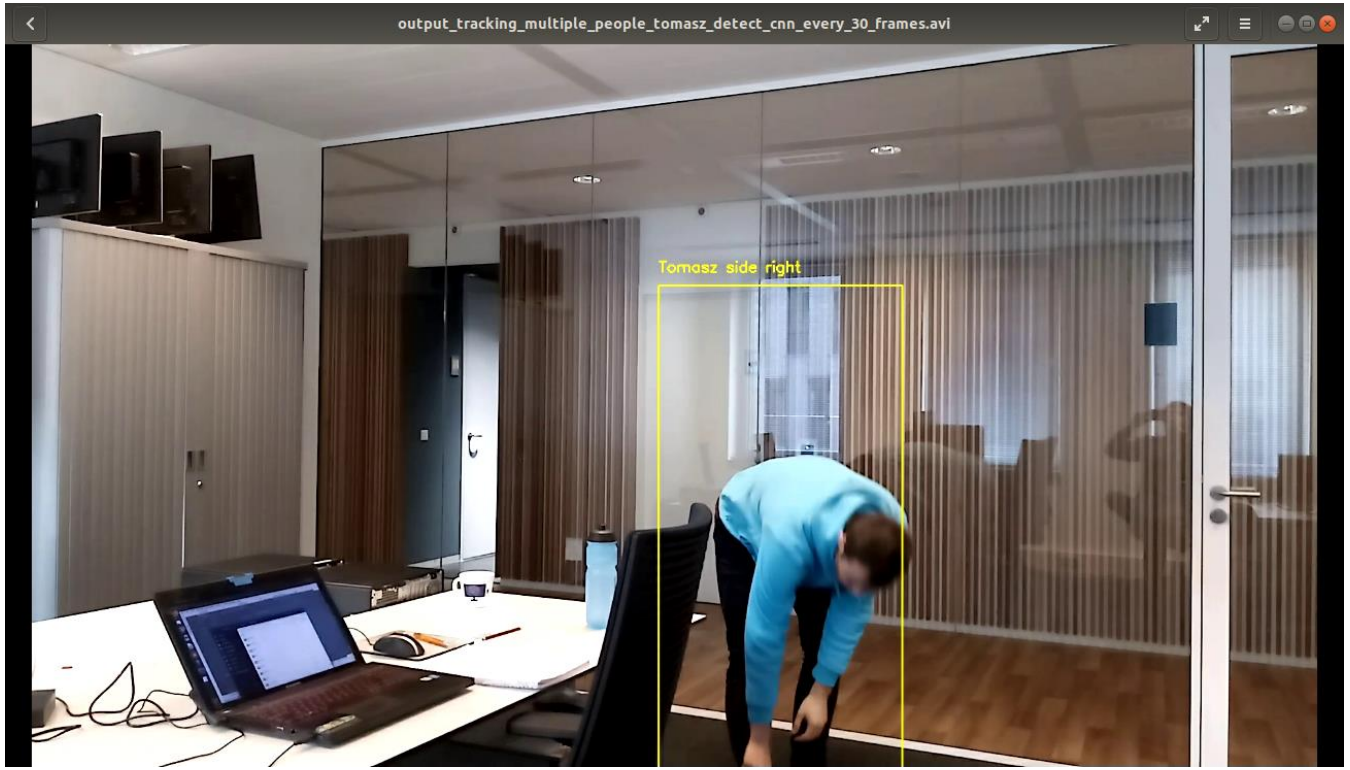
State-of-the-art

Object tracking – example



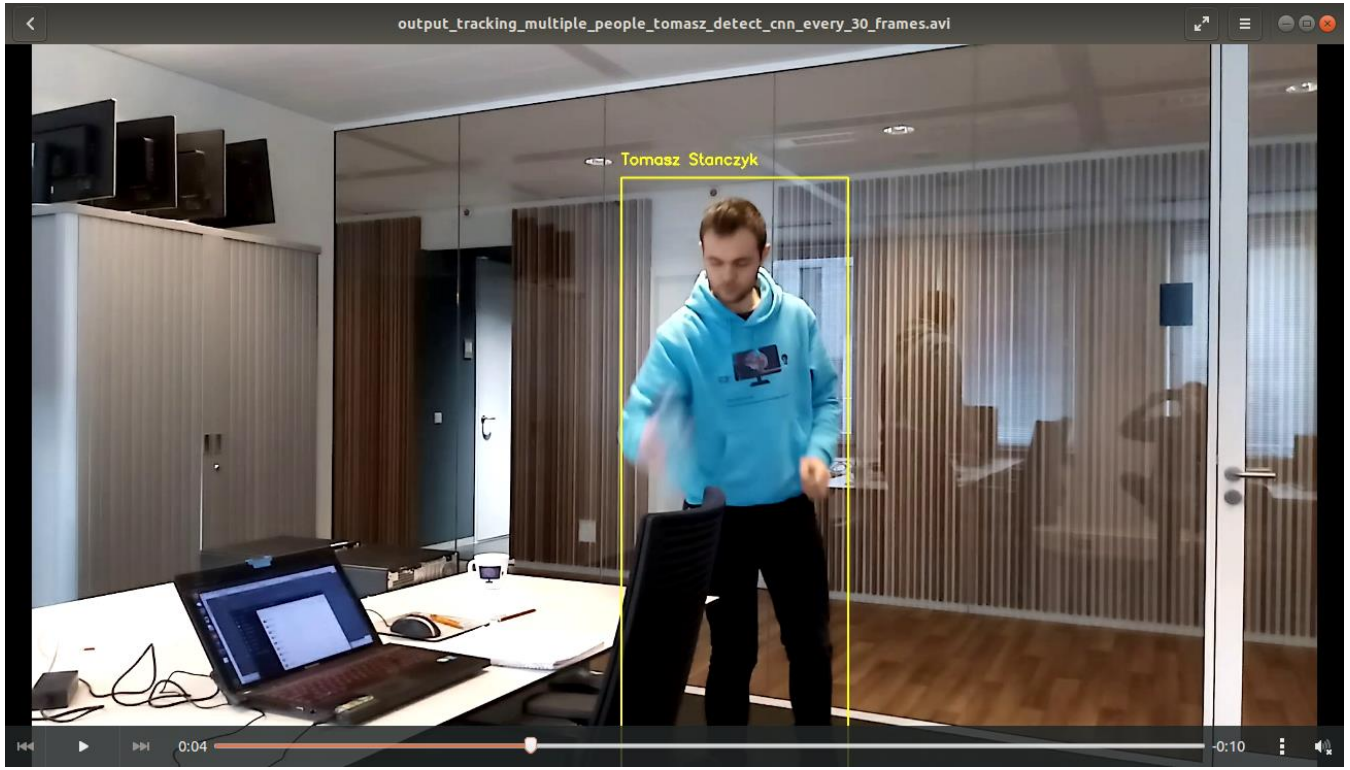
State-of-the-art

Object tracking – example



State-of-the-art

Object tracking – example



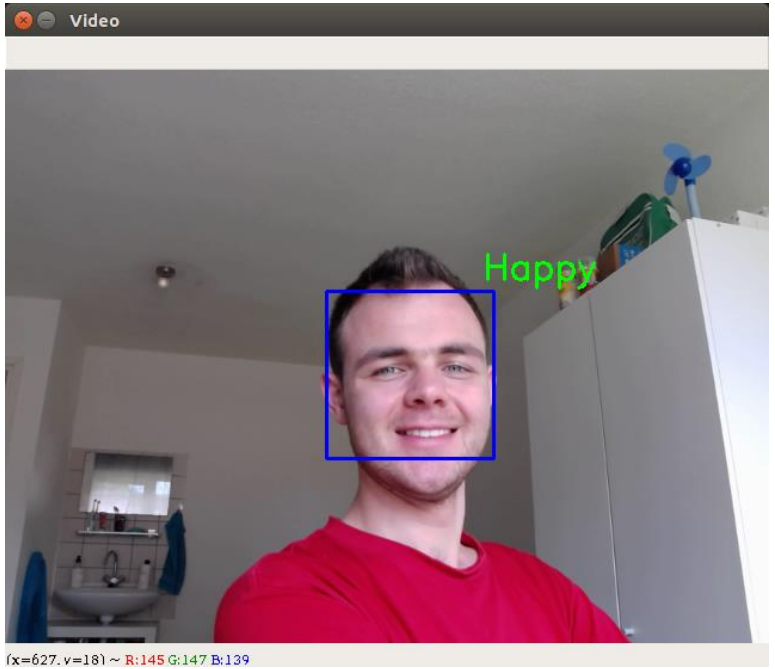
State-of-the-art

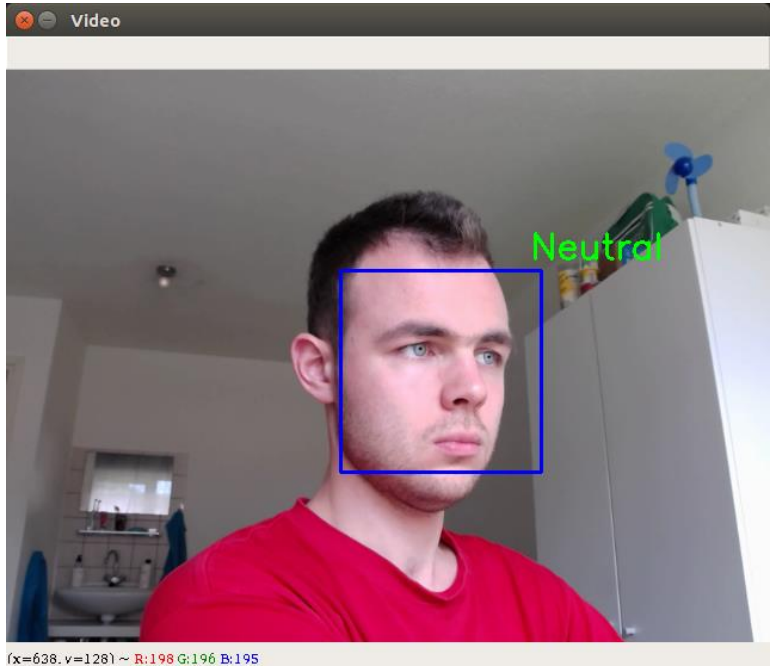
Emotion recognition

- Facial-Expression-Recognition.Pytorch implementation [57]
- Model architecture and pre-trained network weights publicly available
- Model robustness to the position of the face on the input image
  - e.g. only right side of the face available
- Easy and straightforward to use and integrate

State-of-the-art

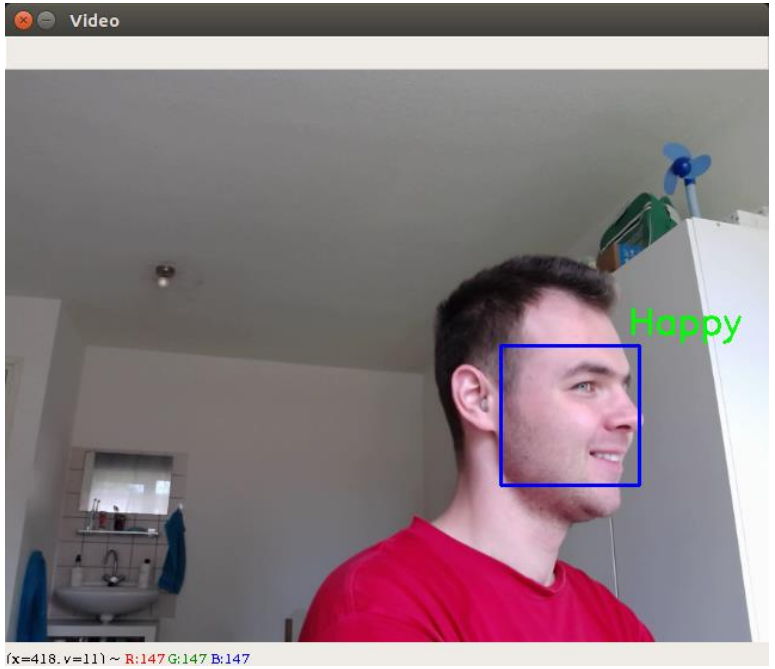
Emotion recognition – examples

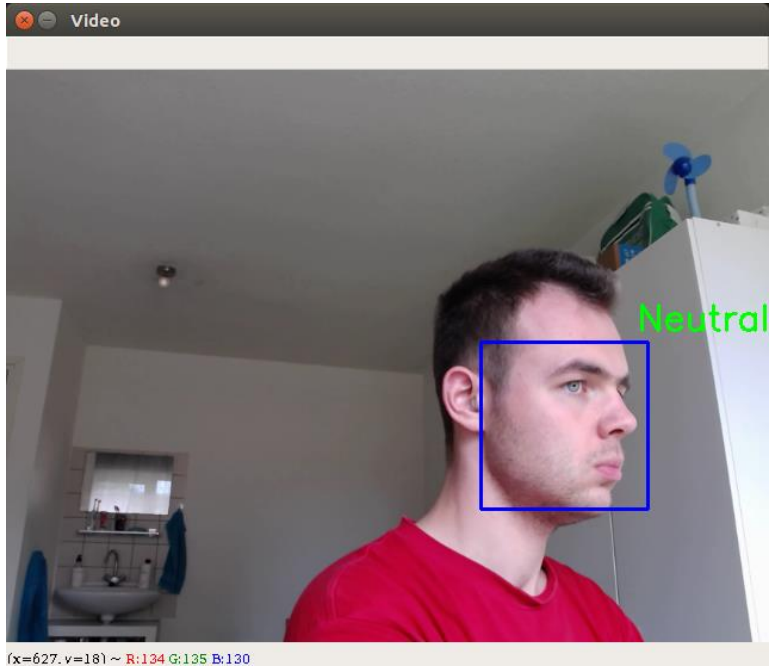




State-of-the-art

Emotion recognition – examples





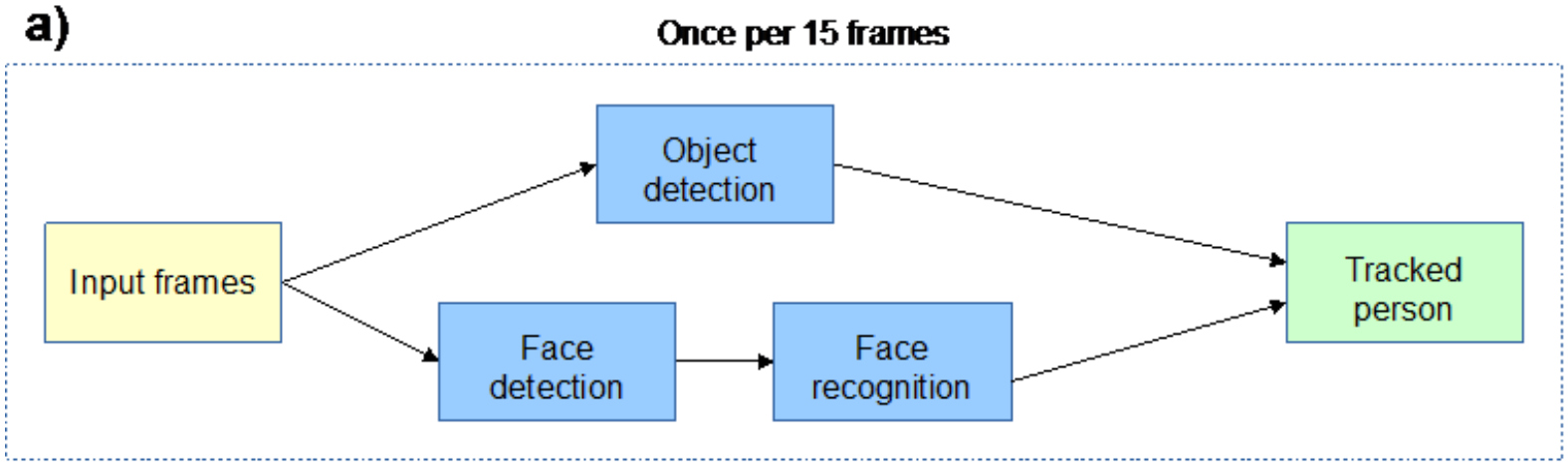
Methods

System external overview

- Connecting all components (including action recognition) into one system
- Video stream from an RGB camera as system input
  - Frame rate 30 FPS
  - Frame sequences of the length 120 frames (4 seconds) recorded
- Activity log as system output
  - date-time stamp indicating the start of the sequence registering,
  - single participant of the action,
  - action recognition prediction in form of top 5 predictions,
  - facial expression of the participant.

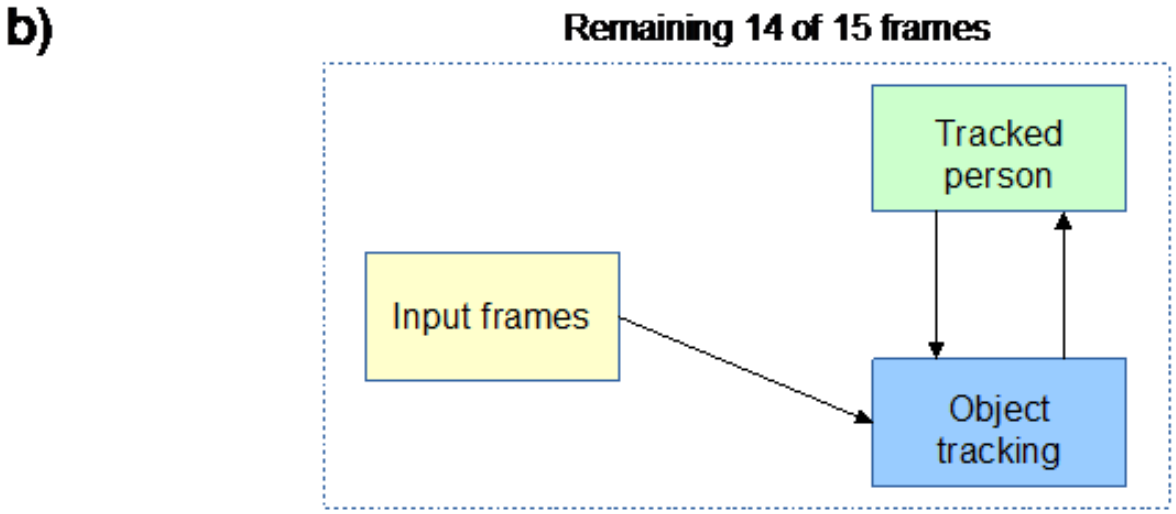
Methods

System architecture – person identification



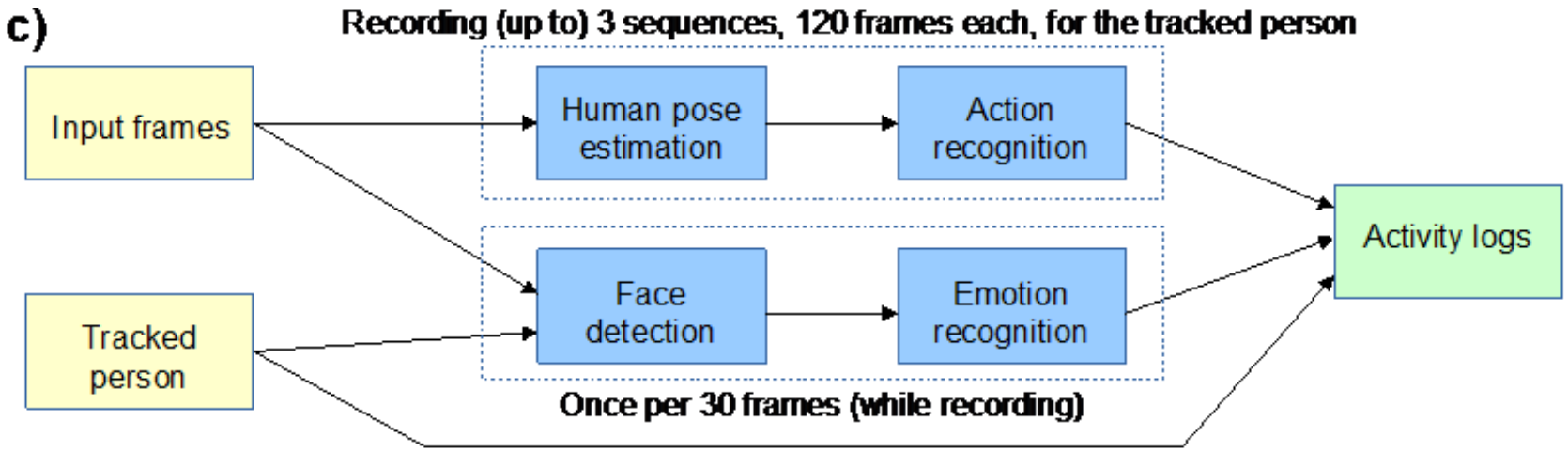
Methods

System architecture – person tracking



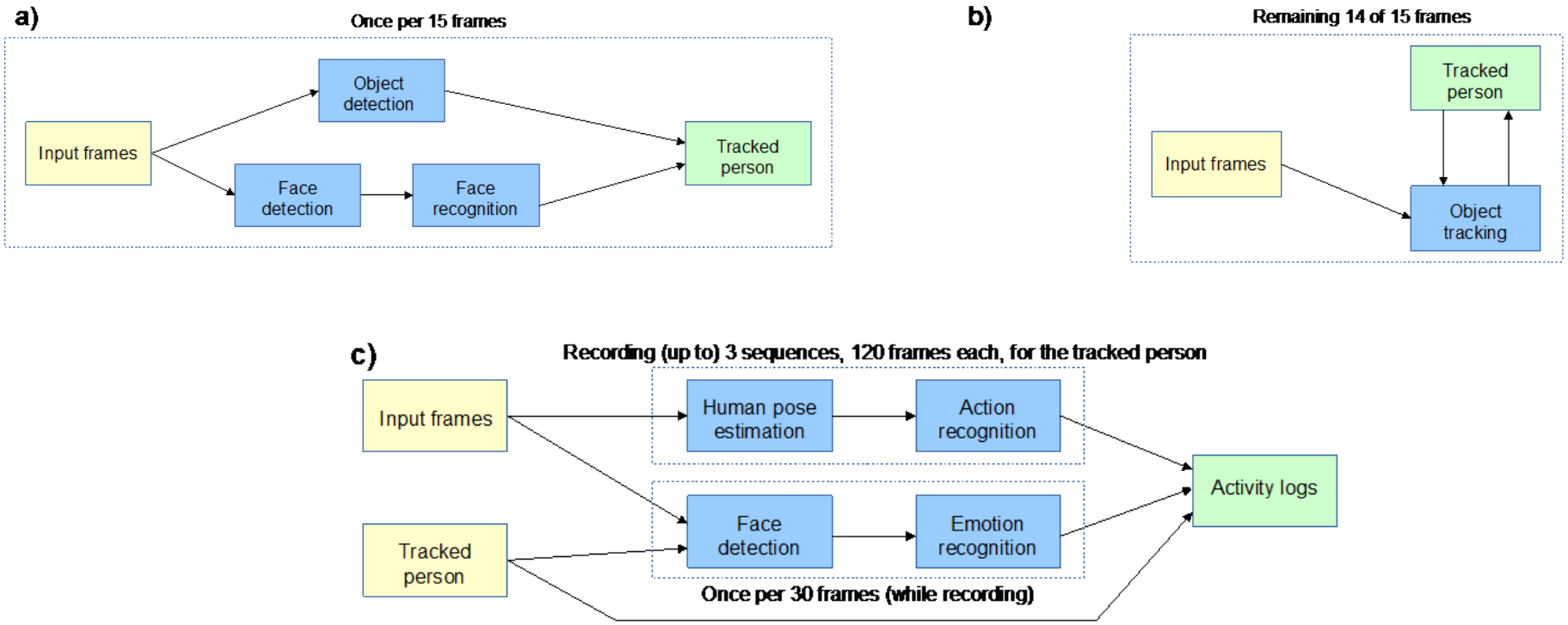
Methods

System architecture – action recognition and emotion recognition



Methods

System whole architecture



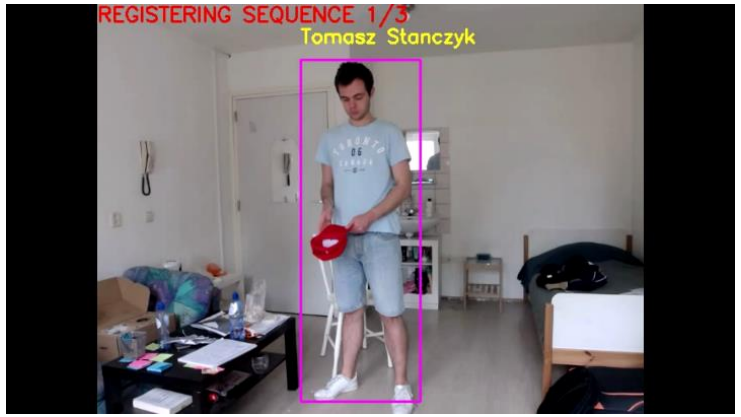
Experiments

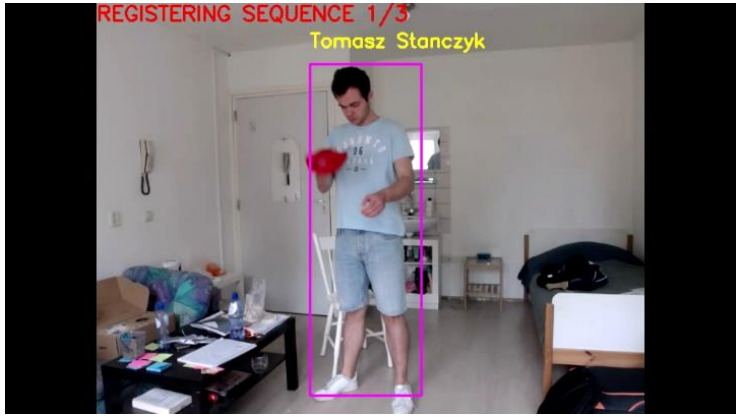
Practical test of the system

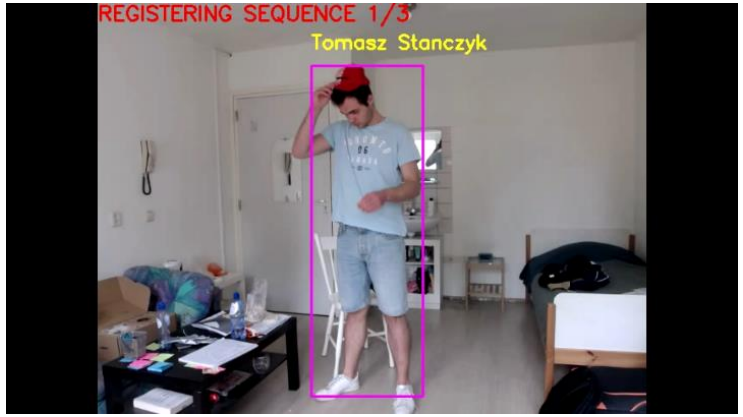
- Video coming from a file or RGB camera stream
- Several frames taken
- Corresponding activity log file content presented

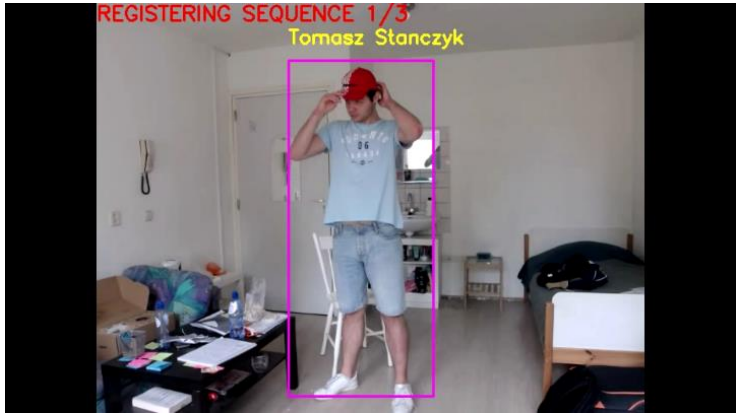
Experiments

Practical test of the system – prerecorded video clip









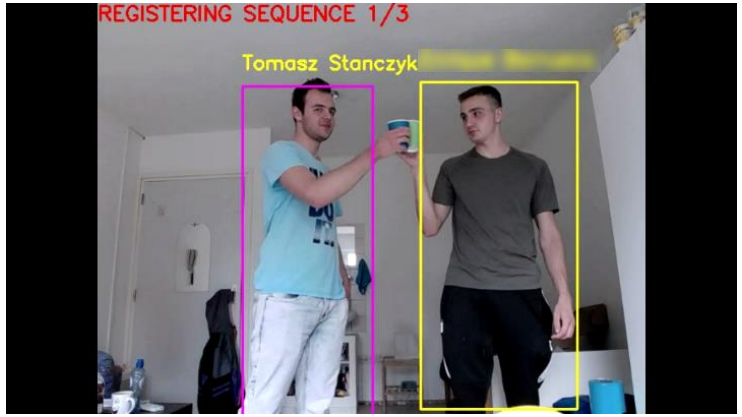
Experiments

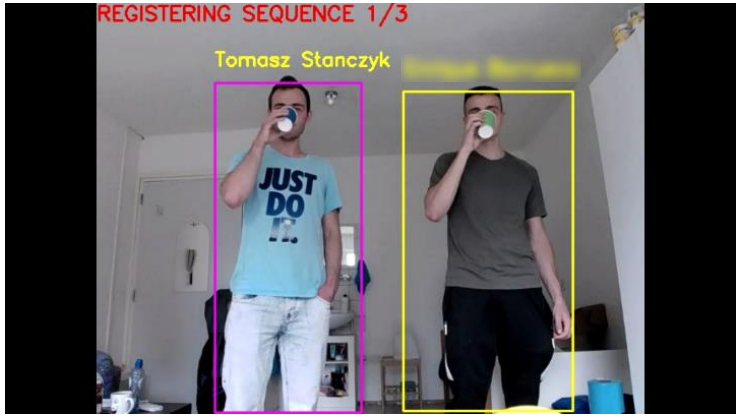
Practical test of the system – prerecorded video clip

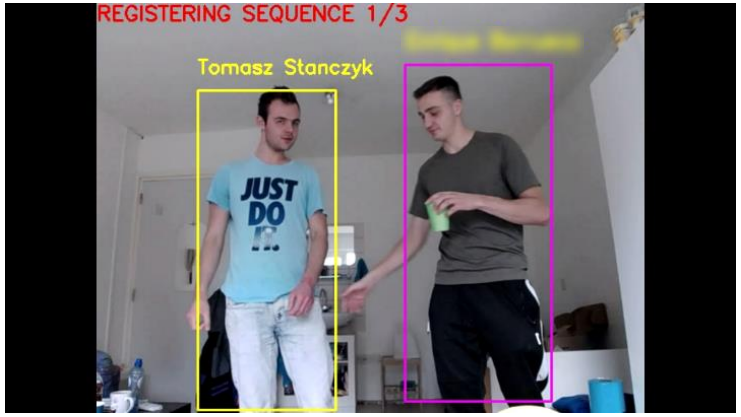
```
####################
2020-07-15 09:01:44.141652
Action participant: Tomasz Stanczyk
Sequence 1 outcome:
1 [ 8.8459] put on a hat/cap (A020)
2 [ 3.7683] make a phone call/answer phone (A028)
3 [ 3.6260] take off a hat/cap (A021)
4 [ 3.2108] brushing hair (A004)
5 [ 2.6035] wear jacket (A014)
Facial expression: Neutral


The average of 1 outcome(s):
1 [ 8.8459] put on a hat/cap (A020)
2 [ 3.7683] make a phone call/answer phone (A028)
3 [ 3.6260] take off a hat/cap (A021)
4 [ 3.2108] brushing hair (A004)
5 [ 2.6035] wear jacket (A014)
Facial expression: Neutral
####################
```

Experiments

Practical test of the system – RGB camera video sequence (1)







Experiments

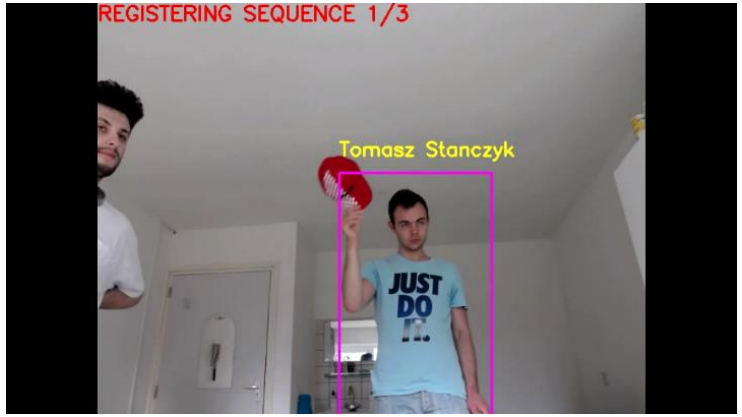
Practical test of the system – RGB camera video sequence (1)

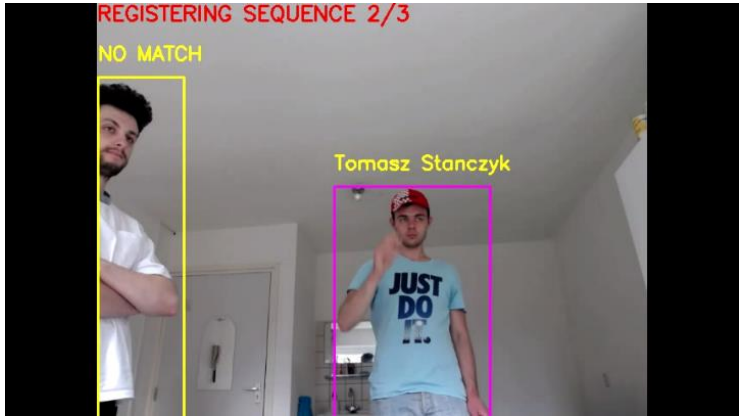
```
####################
2020-07-08 20:09:57.914668
Action participant: Tomasz Stanczyk
Sequence 1 outcome:
1 [ 7.3516] drink water (A001)
2 [ 6.9163] touch other person's pocket (A057)
3 [ 6.2947] giving something to other person
(A056)
4 [ 6.2301] walking towards each other (A059)
5 [ 5.9356] punching/slapping other person (A050)
Facial expression: Neutral


The average of 1 outcome(s):
1 [ 7.3516] drink water (A001)
2 [ 6.9163] touch other person's pocket (A057)
3 [ 6.2947] giving something to other person
(A056)
4 [ 6.2301] walking towards each other (A059)
5 [ 5.9356] punching/slapping other person (A050)
Average facial expression: Neutral
####################
...
```

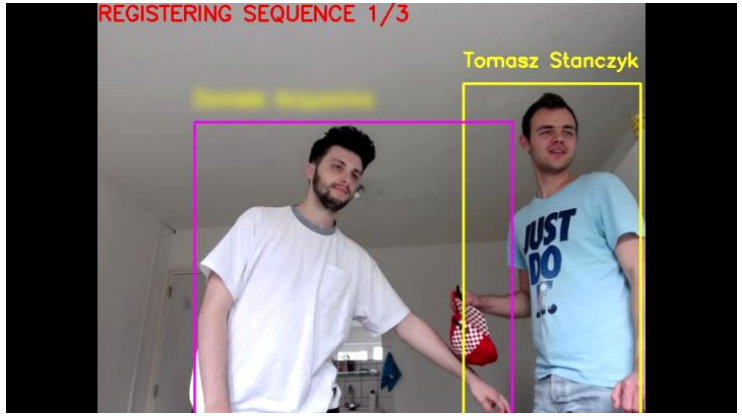
```
...
####################
2020-07-08 20:10:09.569003
Action participant: [redacted]
Sequence 1 outcome:
1 [ 6.6111] punching/slapping other person (A050)
2 [ 6.5966] touch other person's pocket (A057)
3 [ 5.5184] walking towards each other (A059)
4 [ 4.9165] giving something to other person
(A056)
5 [ 4.6787] walking apart from each other (A060)
Facial expression: Happy


The average of 1 outcome(s):
1 [ 6.6111] punching/slapping other person (A050)
2 [ 6.5966] touch other person's pocket (A057)
3 [ 5.5184] walking towards each other (A059)
4 [ 4.9165] giving something to other person
(A056)
5 [ 4.6787] walking apart from each other (A060)
Average facial expression: Happy
####################
```

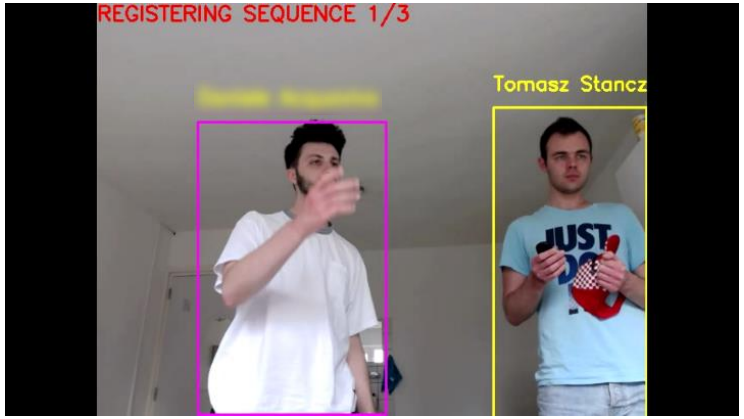
Experiments

Practical test of the system – RGB camera video sequence (2)









Experiments

Practical test of the system – RGB camera video sequence (2)

```
####################
2020-07-07 18:14:47.174790
Action participant: Tomasz Stanczyk
Sequence 1 outcome:
1 [ 2.9433] take off a hat/cap (A021)
2 [ 2.3708] falling (A043)
3 [ 2.0793] walking towards each other (A059)
4 [ 1.8369] walking apart from each other (A060)
5 [ 1.6966] brushing hair (A004)
Facial expression: Sad

2020-07-07 18:14:55.314172
Action participant: Tomasz Stanczyk
Sequence 2 outcome:
1 [ 3.1054] put on a hat/cap (A020)
2 [ 2.5951] walking towards each other (A059)
3 [ 2.2877] standing up (from sitting position) (A009)
4 [ 2.2489] take off a hat/cap (A021)
5 [ 2.0504] punching/slapping other person (A050)
Facial expression: Sad

The average of 2 outcome(s):
1 [ 2.5961] take off a hat/cap (A021)
2 [ 2.3522] put on a hat/cap (A020)
3 [ 2.3372] walking towards each other (A059)
4 [ 1.8328] falling (A043)
5 [ 1.7006] standing up (from sitting position) (A009)
Average facial expression: Sad
####################
```

```
####################
2020-07-07 18:15:11.790064
Action participant: [redacted]
Sequence 1 outcome:
1 [ 6.6559] touch other person's pocket (A057)
2 [ 5.7932] giving something to other person (A056)
3 [ 5.3238] punching/slapping other person (A050)
4 [ 4.2379] walking towards each other (A059)
5 [ 3.6650] pushing other person (A052)
Facial expression: Neutral


The average of 1 outcome(s):
1 [ 6.6559] touch other person's pocket (A057)
2 [ 5.7932] giving something to other person (A056)
3 [ 5.3238] punching/slapping other person (A050)
4 [ 4.2379] walking towards each other (A059)
5 [ 3.6650] pushing other person (A052)
Average facial expression: Neutral
####################
```

Experiments

Practical test of the system – limitations

- Possibly not accurate prediction results for more than once action in the sequence
  - Network "looking for" action matching to all performed action
  - Potentially not performed action predicted
- Possibly confusing similar actions
  - E.g. reading and writing
- Top 1 prediction on custom samples 70.55% and top 5: 87.12%
  - In 12.88% of the cases action not present in top 5 predictions
- Videos coming directly from the camera – more behavior not related to the main action
  - E.g. walking towards a hook to pick a jacket and wearing it afterwards

Experiments

Performance speed – testing PC hardware

- Memory: 16 GB
- Processor: Intel® Xeon(R) CPU E5-1650 v4 @ 3.60GHz × 12
- Graphics : GeForce GTX 1070 Ti/PCIe/SSE2
- OS type: 64-bit

Experiments

Performance speed – action recognition component

- Custom samples with 120 or more frames selected
  - 57 testing samples used
- Only first 120 frames used
  - The same number as frames registered for action recognition
- Extracting human pose skeletons for each sample
- Inputting the stacked skeletons into network models

Experiments

Performance speed – action recognition component

- Performing test on the whole component and separately on pose extraction and network inference
- Frame rate of the recording samples: 30 FPS

	Whole action recognition component	Skeleton extraction	Network inference
Average processing time per sample	4.3197 s	3.8592 s	0.4233 s
Average frame rate	27.7798 FPS	31.0944 FPS	283.4978 FPS

Experiments

Performance speed – whole integrated system

- Using all components of the system
  - Full architecture
- Processing the same samples as for action recognition test
- Separate runs with enabling and disabling particular parts
  - Displaying processed frames on the screen
  - Saving processed frames to a separate video output

Experiments

Performance speed – whole integrated system (custom samples)

- Each sample: 120 frames (4 seconds)

	Saving video: YES **Displaying frames: YES**	**Saving video: NO** **Displaying frames: YES**	**Saving video: YES** **Displaying frames: NO**	**Saving video: NO** **Displaying frames: NO**
Average processing time per sample	8.8086 s	8.4293 s	7.4979 s	7.0672 s
Average frame rate	13.7610 FPS	14.3639 FPS	16.1760 FPS	17.1638 FPS

Experiments

Performance speed – whole integrated system

- Additional video clip tested (30 FPS)
  - 1024 frames and 2 people
- People tracked for 720 frames in total
- 304 frames with no:
  - Tracking (yet object detection and face detection being active)
  - Action recognition
  - Emotion recognition

Experiments

Performance speed – whole integrated system (additional video file)

- 1024 frames (around 34.1333 seconds)

	Saving video: YES Displaying frames: YES	Saving video: NO Displaying frames: YES	Saving video: YES Displaying frames: NO	Saving video: NO Displaying frames: NO
Average processing time per sample	43.1666 s	40.2168 s	33.6417 s	29.9671 s
Average frame rate	23.7221 FPS	25.4620 FPS	30.4384 FPS	34.1707 FPS

Discussion

- Necessity of fast action recognition
  - Reporting the inference results immediately
- Action recognition: 27.7798 FPS
  - Nearly real-time for 30 FPS input
- Whole system: up to 17.1638 FPS
- Numbers specific for the testing PC
  - Higher frame rates potentially possible for better hardware (e.g. stronger GPU)
- Performance of speed – not completely satisfactory
- Promising results (34.1707 FPS) when not all components are used at the time

Discussion

- Further enhancements
  - Multi-processing
  - Multi-threading
  - Re-selection of the ready-to-use-components
- Functional enhancements
  - Multi-person tracking
  - Mechanical tracking with a PTZ camera
  - Event detection
- Top priority: system working satisfactorily fast

Conclusion

Answering the following research question:

(1) How such a system [solving the stated problem] can be created and can it be properly integrated?

- Existing solutions found
  - Object detection
  - Face detection
  - Face recognition
  - Object tracking
  - Emotion recognition
- Suitable action recognition component developed (Part I)
- All components connected together and integrated into one system with presented architecture

Conclusion

(Partially) Answering the following research question:

(6) Can [the developed] action recognition component run in real time?

- Testing on a modern, yet affordable PC
- Input frame rate set to 30 FPS
- Running nearly in real-time
  - Average frame rate of 27.7798 FPS

Conclusion

(Partially) Answering the following research question:

(2) Can the whole system run in real-time?

- Average frame rate reached 17.1638 FPS
  - Not acknowledged as real-time
- With further optimization, higher testing frame rate could be reached

Conclusion

Answering the following research question:

(3) Is high-performance PC hardware needed for this system?

And complementing the following research questions:

(6) Can [the developed] action recognition component run in real time?
(2) Can the whole system run in real-time?

- Results on testing PC: not reaching 30 FPS (yet with 30 FPS input)
- 30 FPS could potentially be reached (or surpassed) by PC with stronger hardware
  - Both action recognition and the whole system
  - Real-time possibility

Acknowledgement

- RWTH Aachen's cluster infrastructure was used for model training and evaluation described in Section I.3.A and I.3.C.

References

[1] A. K. Soutter. What can we learn about wellbeing in school? 2011.

[2] L. J. Harrison, E. Murray. Stress, Coping and Wellbeing in Kindergarten: Children's Perspectives on Personal, Interpersonal and Institutional Challenges of School. 2014.

[3] G. G. Fillenbaum. The wellbeing of the elderly: approaches to multidimensional assessment. 1984.

[4] A. Shahroudy, J. Liu, T. Ng, G. Wang. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. 2016.

[5] K. Soomro, A. R. Zamir, M. Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild, 2012.

[6] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, T. Serre. HMDB: A Large Video Database for Human Motion Recognition. 2011

[7] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei. Large-scale Video Classification with Convolutional Neural Networks. 2014.

[8] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, A. Zisserman. The Kinetics Human Action Video Dataset. 2017.

[9] J. Carreira, E. Noland, A. Banki-Horvath, C. Hillier, Andrew Zisserman. A Short Note about Kinetics-600. 2018.

[10] J. Carreira, E. Noland, C. Hillier, Andrew Zisserman. A Short Note on the Kinetics-700 Human Action Dataset, 2019.

[11] https://github.com/yysijie/st-gcn/blob/master/OLD_README.md#kinetics-skeleton

[12] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, Yaser Sheikh. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. 2018.

[13] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, Kate Saenko, T. Darrell. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. 2015.

[14] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. 2016.

[15] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, B. Russell. ActionVLAD: Learning spatio-temporal aggregation for action classification. 2017.

References

[16] G. Varol, I. Laptev, C. Schmid. Long-term Temporal Convolutions for Action Recognition. 2016.

[17] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. 2014.

[18] D. Tran, J. Ray, Z. Shou, S.-F. Chang, M. Paluri. ConvNet Architecture Search for Spatiotemporal Feature Learning. 2017.

[19] C. Feichtenhofer, A. Pinz, A. Zisserman. Convolutional Two-Stream Network Fusion for Video Action Recognition. 2016.

[20] J. Carreira, A. Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. 2017.

[21] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, L. Van Gool. Temporal 3D ConvNets: New Architecture and Transfer Learning for Video Classification. 2017.

[22] S. Zhang, X. Liu, J. Xiao. On geometric features for skeleton-based action recognition using multilayer LSTM networks. 2017.

[23] Z. Ding, P. Wang, P. O. Ogunbona, W. Li. Investigation of Different Skeleton Features for CNN-based 3D Action Recognition. 2017.

[24] L. Shi, Y. Zhang, J. Cheng, H. Lu. Skeleton-Based Action Recognition with Multi-Stream Adaptive Graph Convolutional Networks. 2019.

[25] T. N. Kipf, M. Welling. Semi-Supervised Classification with Graph Convolutional Networks. 2016.

[26] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. 2018.

[27] L. Shi, Y. Zhang, J. Cheng, H. Lu. Two-Stream Adaptive Graph Convolutional Networks for Skeleton-Based Action Recognition. 2018.

[28] Y. Wen, L. Gao, H. Fu, F.-L.Zhang, S. Xia. Graph CNNs with Motif and Variable Temporal Block for Skeleton-Based Action Recognition. 2019.

[29] C. Li, Q. Zhong, D. Xie, S. Pu. Co-occurrence Feature Learning from Skeleton Data for Action Recognition and Detection with Hierarchical Aggregation. 2018.

[30] C. Xie, C. Li, B. Zhang, C. Chen, J. Han, C. Zou, J. Liu. Memory Attention Networks for Skeleton-based Action Recognition. 2018.

References

[31] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, Q. Tian. Actional-Structural Graph Convolutional Networks for Skeleton-based Action Recognition. 2019.

[32] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, N. Zheng. View Adaptive Neural Networks for High Performance Skeleton-based Human Action Recognition. 2018.

[33] W. Peng, X. Hong, H. Chen, G. Zhao. Learning Graph Convolutional Network for Skeleton-based Human Action Recognition by Neural Searching. 2019.

[34] L. Shi, Y. Zhang, J. Cheng, H. Lu. Skeleton-Based Action Recognition with Directed Graph Neural Networks. 2019.

[35] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, H. Lu. Skeleton-Based Action Recognition with Shift Graph Convolutional Network. 2020.

[36] Z. Liu, H. Zhang, Z. Chen, Z. Wang, W. Ouyang. Disentangling and Unifying Graph Convolutions for Skeleton-Based Action Recognition. 2020.

[37] https://github.com/lshiwjx/2s-AGCN

[38] S. Aubry, S. Laraba, J. Tilmanne, T. Dutoit. Action recognition based on 2D skeletons extracted from RGB videos. 2019.

[39] F. Marshall, S. Zhang, B. Scotney. Comparison of Activity Recognition using 2D and 3D Skeletal Joint Data. 2019.

[40] B. Li, M. He, X. Cheng, Y. Chen, Y. Dai. Skeleton Based Action Recognition Using Translation-Scale Invariant Image Mapping And Multi-Scale Deep CNN. 2017.

[41] D. Osokin. Real-time 2D Multi-Person Pose Estimation on CPU: Lightweight OpenPose. 2018.

[42] https://github.com/CMU-Perceptual-Computing-Lab/openpose

[43] https://github.com/Daniil-Osokin/lightweight-human-pose-estimation.pytorch

[44] J. Redmon, A. Farhadi. YOLOv3: An Incremental Improvement. 2018.

[45] https://pjreddie.com/darknet/yolo/

References

[46] https://github.com/pjreddie/darknet

[47] https://github.com/eriklindernoren/PyTorch-YOLOv3

[48] http://dlib.net/python/index.html#dlib.cnn_face_detection_model_v1

[49] D. E. King. Dlib-ml: A Machine Learning Toolkit. 2009.

[50] http://dlib.net/cnn_face_detector.py.html

[51] http://dlib.net/python/index.html#dlib.face_recognition_model_v1

[52] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, Lior Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. 2014.

[53] F. Schroff, D. Kalenichenko, J. Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. 2015.

[54] http://dlib.net/face_recognition.py.html

[55] http://dlib.net/python/index.html#dlib.correlation_tracker

[56] M. Danelljan, G. Häger, F. S. Khan, M. Felsberg. Accurate Scale Estimation for Robust Visual Tracking. 2014.

[57] https://github.com/WuJie1010/Facial-Expression-Recognition.Pytorch

Thank you for your attention!

Questions and answers