RECOGNITION OF HUMAN-COMPUTER INTERACTION GESTURES ACQUIRED BY INERTIAL MOTION SENSORS WITH THE USE OF HIDDEN MARKOV MODELS

Aleksander Sawicki[1], Kristina Daunoravičienė[2], Julius Griskevicius[2]

[1] Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

[2] Faculty of Mechanics,Vilnius Gediminas Technical University, Lithuania

Abstract: The paper presents the algorithm of recognition of selected Human-Computer Interaction (HCI) gestures acquired by inertial motion sensors. The possibilities of using Hidden Markov Models as classifiers have been verified. The experiments investigated the possibility of using a methodology dedicated to the recognition of virtual reality (VR) game gestures to classify HCI gestures. The paper compares the accuracy of classification depending on the method of discretization of the forearm orientation signals. The evaluation of the accuracy of the classification was carried out with the use of 3-fold cross validation. The paper uses author's data corpus containing in total 720 time series acquired from 20 human subjects.

Keywords: HMM, Classification, HCI, IMU

1. Introduction

Gestures can be used to transmit messages through the body's significant, meaningful motions. Consequently, they may constitute a form of non-verbal communication. Gestures, with particular emphasis on hand movements, are considered to be one of the most important in our daily communication. They are considered the most promising in the field of Human-Computer Interaction [1]. Therefore, nowadays a significant number of studies concerning their recognition are carried out.

The scientific work can be divided into three groups in relation to the type of used signals. In the first one, the classification of the gesture is based on the orientation signal which represents information about the rotation of the sensor or the body.

Advances in Computer Science Research, vol. 15, pp. .., 2021. DOI: 10.24427/acsr-2021-vol15-0001

In the second case, the orientation is used simultaneously with other types of signals. In the last group, orientation information is ignored and recognition is performed with raw sensor values.

The first of these groups includes the approach presented in the publications [2,3]. In the first one, the authors recognized the gestures used in VR games based only on the forearm orientation. As a result of IMU data fusion filter operation, a quaternions describing rotations in 3D space were determined. The data in this form were transformed into three Euler angles and further processed. In publication [3] the authors used the hand orientation data recorded as a video. The presented approach uses information on hand tilt in a 2D system that was oriented in accordance with the camera lens. The developed gesture recognition algorithm used only one rotation angle.

The second group of approaches includes the systems described in [4,5]. In the first one, the authors used information about hand orientation as well as its speed and location. Here, the video recordings were also used as the input signals of the system. In another approach [5], the feature vector containing the quaternion was expanded by the pressure signal data. The described system used the signals from the IMU unit and a specialized glove equipped with the pressure sensors.

In the last group of papers, the authors completely abandoned the use of orientation signals in favour of other types of data. In [6] the authors used the signals acquired by an inertial unit such as acceleration and angular velocity. The paper does not use the data fusion algorithm, which would allow to determine the orientation. It should be noted that there are publicly available and open source software packages that allows such a process. The lack of using this procedure is a deliberate and conscious action. A very similar approach was also presented in [7].

It should be emphasized that in the literature there are many articles in which unprocessed signals are used [6,7]. The use of different types of signals [4,5], is not an individual approach either. At the moment, however, there are work no extending the method presented in [2]. There is no major work group on HMM applications to classify gestures on the basis of three-dimensional spatial orientation signals.

After reading the above mentioned paper, the two following questions arise: “Is the methodology presented in [2] adequate for recognizing another group of gestures?” and "If an equally divided division of all Euler’s angles ensures the best accuracy of classification?"could be asked. This paper presents an attempt to answer those questions. The article begins with a brief description of the available database with a specification of the devices used to acquire the motion signal. The following section describes the methodology derived from the literature and the author’s modification. Finally, the results and conclusions are presented.

2. Methodology

The paper contains results of gesture recognition with the use of orientation signals in the form of quaternion time series. The developed pattern recognition algorithm is divided into three main blocks: "Prepocessing", "Vectorization" and "Classification" (Fig. 1.). The forearm orientations in the quaternion time series form are delivered to

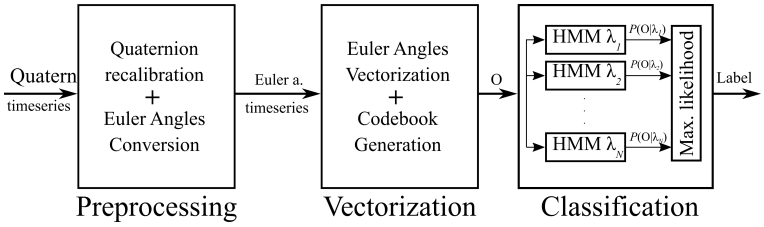

Fig. 1. Simplified structure of the gesture detection algorithm

the system input. In the first "Prepocessing" block, the quaternion series is normalized and converted to Euler angles series.The information in this form is vectorised and then used to generate a sequence of observations (O). In the "Classification" block, with the use of the Viterbi algorithm, a probability parameter (likelihood) is calculated for the four trained hidden Markov models. The result of the block is a single label describing the most probable gesture.

2.1 Data Set

The paper uses an authors collection of gestures that can be used in Human-Computer Interaction (HCI). The gestures described as "Come", "Turn Right", "Turn Over" and "Sit Down" were performed with the right hand. Motion dataset was inspired by the list of motions used in the renowned article [1]. The visualization of gestures is shown in Fig 2. The data corpus was created during the participation in the international "PROM" internship (financed by the Polish National Agency for Academic Exchange) in the Department of Biomechanics at the Gediminas University of Technology in Vilnius. The database consist of motion tracking sessions performed by 20 participants, including 8 men and 12 women. The average age of the participant was 26.15 years with a standard deviation of 6.44. There were no participants with illnesses or injuries that could affect the realization of particular gestures. The study participants performed 9 repetitions of individual gestures, which allowed to obtain

Fig. 2. Visualisation of the used gestures

a total of 720 movement sessions. Following manual segmentation, the single repetition time was of the order of second. Currently, discussions are taking place with the "PROM" coordinators in order to make the data available on the Zendodo or similar platform.

The acquisition of measurement data was carried out with the use of commercially available inertial motion tracking system called "Perception Neuron". This device consists of a set of 17 IMUs (Inertial Measurement Units). Each of the sensors provided the measurement of quantities as acceleration, angular velocity (magnetic field strength is proprietary), and due to the implemented algorithm of data fusion, orientation. The sensors were distributed evenly on the body, which allowed to track the entire skeleton. Fig. 3 A) presents the visualization of the person during the Turn Right gesture. As a result of preprocessing (Section 2.2), each of the registered persons was oriented according to the X axis of the coordinate system. In Fig. 3 B) forearm rotation axes was displayed by additional lines. It should be noted that the inertial system for the right forearm allows for the determination of 3 rotation angles around 3 axes.

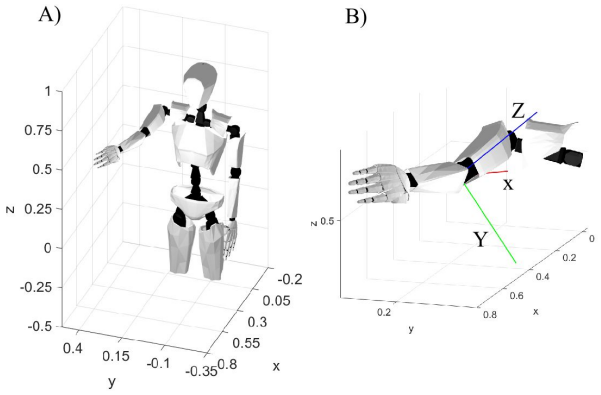

Fig. 3. A) Body visualisation in MATLAB software B) Forearm rotation axis visualisation

In the next part of the study, according to the literature [2], signal concerning the orientation of the forearm were used. The use of measurement data obtained from sensors located on the forearm is a popular solution which is applied in many scientific works [8]. IMU sensors are responsible for determining only their own orientation. The determination of skeletal limb orientation is done by calibration, in which the T-pose calibration gesture is performed by a participant. The perception neuron device enables calibration to be performed when all 17 sensors were detected. Therefore, at the moment of tracing one limb it was necessary to wear the complete suit. Moreover, the torso orientation was used as additional data in signal preprocessing, in elimination of Yaw/Azimuth offset angle approach.

Through signal processing, the orientation of the forearm in the form of a quaternion was converted to Euler's angles in Yaw-Pitch-Roll convention. This means that first the Z axis was rotated by an angle of Ψ, then the Y axis by an angle of θ and finally the X axis was rotated by an angle of Φ. It should be noted that the sensors of the Perception Neuron system are capable of conducting measurements at a frequency of 120 Hz. The precision of the measurements is not specified by the manufacturer. The accuracy of a similar class of devices (e.g. UM7-LT) for dynamic situations is about $\pm 5^{\circ}$ for *Pitch* (θ) and *Roll* (Φ), whereas $\pm 8^{\circ}$ for *Yaw* (Ψ) angle.

2.2 Preprocessing

The data acquisition process was carried out within a few days. Due to the participation of many participants, and thus the long duration of the experiments, the measurements were carried out in more than one room (caused by the organisational issues).

Inertial motion systems allow to determine the orientation of limbs, in relation to the global coordinate system oriented according to the Earth's magnetic pole.

While performing the gestures, some of the participants were directed towards the magnetic South while others were directed towards the magnetic East of the Earth. In further codebook generation work, leaving the data unchanged would affect the *Yaw* angle and interfere with the classification accuracy. Therefore, the preprocessing procedure involved the artificial rotation of the individual persons in order to orient them in the same direction. In general, researchers use various types of normalisation techniques. In [9] the authors multiplied the data by a quaternion conjugated to q0 where "q0 is the heading offset with respect to the magnetic north". Since the perception neuron was equipped with a sensor located on the back, information about its orientation was used. The use of artificial rotation is a necessary process, commonly exploited also in motion data analysis [10].

According to the methodology presented in the literature [2,4], the codebook was generated using orientation information. First, the time series of quaternions were converted into a series of Euler angles. For this purpose, reverse trigonometric functions presented in equation (1) were used.

$$\begin{aligned} \Psi &= atan\tfrac{2q_yq_w-2q_xq_z}{1-2q_y^2-2q_z^2} \\ \theta &= asin(2q_xq_y+2q_zq_w) \\ \Phi &= atan\tfrac{2q_xq_w-2q_yq_z}{1-2q_x^2-2q_z^2} \end{aligned} \tag{1}$$

where:
q_w, q_x, q_y, q_z -quaternion coponents;
Ψ, θ, Φ -rotation angle *Yaw*,*Pitch*,*Roll*.

It should be emphasized that in the proposed instead of the function `atan` the procedure `atan2` was used. As a result, the angles *Yaw* (Ψ) and *Roll* (Φ) are in the range of $\pm 180^\circ$ while *Pitch* (θ) in the range of $\pm 90^\circ$.

2.3 CodeBook generation

In the next stage of the study, according to the methodology presented in [2,4], the angles of rotation were discretized. The three states sequences (resulting from the 3 rotation axes) were used to generate final observations. In order to discretize the Euler angles, the parameter L [2] was defined, which describes the number of states into which the range of 180° was divided. For example, for a parameter L equal to 3, 180° was divided into ranges of 60°. In this case, the *Pitch* (θ) angle generated one of the states in the $0 \div (L-1))$ range, which means one element from $\{0,1,2\}$. As

a result of the `atan2` function, angles *Yaw*(Ψ) and *Roll*(Φ) have a range of 360°. The state determination process is described in Algorithm 1.

Algorithm 1 State (angle, range, L)

Require: $angle \in \langle 0;180\rangle, range \in \{180,360\}, L \in \{3,4,5,6,7,8\}$

```
if range=180 then
    thr=linspace(0,180,L+1)
else
    thr=linspace(0,360,2L+1)
end if
for i = 2 to i < size(thr) do
    if angle<=thr(i) then
        state=i-2
        break
    end if
end for
return state
```

In Fig.4. an exemplary division of Euler angles into states for case A) L=3 and B) L=4 is presented.

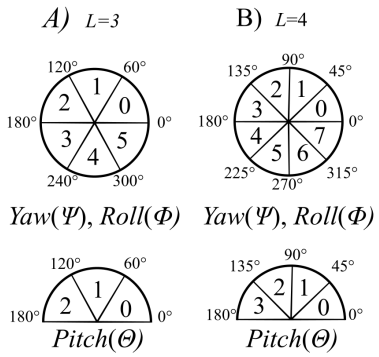

Fig. 4. Generation of angle states based on parameter L parameter A) L=3 B) L=4

In the case described in the literature [2], the final observation was a combination of states of three angles of rotation. Additionally, when the angle *Pitch* (θ) was equal to 0 or $L-1$, the angle state *Roll* (ϕ) was ignored (Algorithm 2). This assumption has reduced the number of observable states.

Algorithm 2 Observation (Yaw, Pitch, Roll, L)

Require: $Yaw, Roll \in \langle 0;360 \rangle, Pitch \in \langle 0;180 \rangle, L \in \{3,4,5,6,7,8\}$
1: **if** $state180(Pitch, L) == 0$ **or** $L-1$ **then**
2: $O = 0 + 2L \cdot state(Pitch, 180, L) + 2L^2 \cdot state(Yaw, 360, L) + 1$
3: **else**
4: $O = state(Roll, 360, L) + 2L \cdot state(Pitch, 180, L) + 2L^2 \cdot state(Yaw, 360, L) + 1$
5: **end if**
6: **return** O

In the original formula (algorithm 2), the value of the *Roll* angle significantly affects the generation of a sequence of observations. It should be noted that this angle has a full range of 360°. Therefore, for each L parameter, it is possible to generate $2L$ states based only on this angle. For example, for parameter $L = 6$, the range of 360 ° will be divided into 12 states with a range of 30°. This assumption significantly increases the requirement for a training base. Therefore, in this paper we propose a modified method of generation of observable states, in which, regardless of the value of the L parameter, the *Roll* was divided into equal states with a width of 120°(Algorithm 4).

Algorithm 3 Proposed observation (Yaw, Pitch, Roll, L)

Require: $Yaw, Roll \in \langle 0;360 \rangle, Pitch \in \langle 0;180 \rangle, L \in \{3,4,5,6,7,8\}$
1: **if** $state(Pitch, 180, L) == 0$ **or** $L-1$ **then**
2: $O = 0 + 2L^2 \cdot state(Pitch, 180, L) + \cdot state(Yaw, 360, L) + 1$
3: **else**
4: $O = (4L^2) \cdot state(Roll, 360, 3) + 2L^2 \cdot state(Pitch, 180, L) + \cdot state(Yaw, 360, L) + 1$
5: **end if**
6: **return** O

2.4 Classification

The aim of the paper was to verify whether the methods described in the article [2] can be used to classify HCI gestures. In their original form, the presented methods were dedicated to the gestures used in VR games. In the conducted studies, the influence of the L parameter, determining the vectorization of data on the accuracy of classification, was examined.

The conducted experiments concerned the recognition of 4 gestures described as "Come", "Turn Right", "Turn Over" and "Sit Down". Each of these gestures was

represented by a sequence of observations related to the forearm orientation. The paper presents the influence of the $L = \{3,4,5,6,7,8\}$ discretization parameter on the results of gesture recognition. In this study, 3-fold cross-validation repeated 5 times was used.

For the classic method taken from the literature, a total of 360 hidden Markov models were trained (4 gestures · 6 variants L parameter · 3 cross validation · 5 repetition). The article presents a modification of Euler angles discretizing parameter. Therefore additional 360 models were trained.

The observations were related to the method of Euler angles discretization. Therefore, the number of observable states M depended on the parameter L. For the classic method (Algorithm 2), the total number of observed states is described by Equation 2.

$$M = 4 \cdot L^3 - 8 \cdot L^2 + 4 \cdot L + 1, \tag{2}$$

The number of observable states for the modified discretization algorithm (Algorithm 3) is described by Equation 3.

$$M = 6 \cdot L^2 - 8 \cdot L + 1, \tag{3}$$

The Literature [2] does not specify the number of hidden states N . Therefore, pilot studies have been carried out in which a fixed number of states of 4 has been selected. For example, for the parameter L=3 (the least complicated models), hidden Markov models were described by parameters with dimensions: 1×4, vector of initial state probabilities; 4×4, matrix of transition probabilities; 4×49, matrix of emission probabilities (classic approach) or 4×31, matrix of emission probabilities (proposed approach).

Training and prediction of hidden Markov models was carried out with the use of *seqHMM* package in *R* programming language. The model parameters were set using the Baum-Welch algorithm. Due to the gradient character of the method, the learning process was restarted 100 times. The classification of individual observations O was carried out using the Viterbi algorithm and the required determination of likelihood indicators for each of the four trained models. The series was classified as the most likely gesture. The observation could therefore be classified correctly, incorrectly or not at all.

3. Resuts

Fig.5. shows 3 classification results described as "Classic Approach", "Proposed Approach" and "Literature Results". The first two use the author's database. The "Proposed Approch" refers to our methodology, whereas the "Classic Approach" to the

methodology used in the publication [2]. the "Literature Results" are results taken directly from the publication [2] and relate to outcomes obtained using the authors' database (which is not public available). In our opinion, a significant difference between "Classic Approach" and "Literature Results" is due to the different types of gestures used in the various databases.

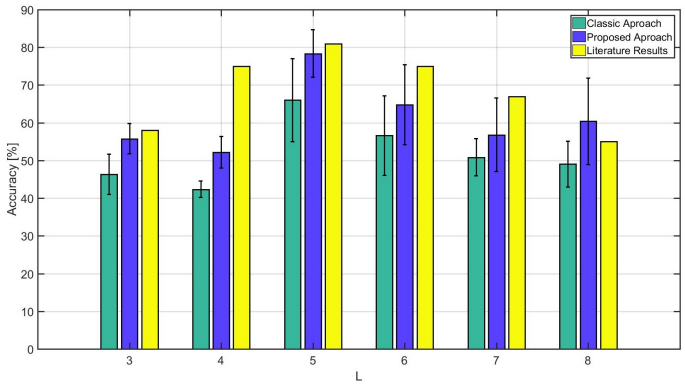

Fig. 5. Classification accuracy as a function of parameter L. Error bars represent standard deviation.

The figure shows the averaged results for the 3-fold cross validation, repeated 5 times. The classification accuracy was presented on the y-axis, whereas L parameter values were presented along the x-axis. Please note that the publication [2] presents only averaged results as a graph, and do not provide any information about standard deviation. Therefore, "Literature Results" bars do not contain error plots.

The highest value for "Proposed Approach" as well "Classic Approach" classification accuracy was achieved for $L = 5$, which is consistent with the "Literature Results". The use of classic codebook generation methodology provides maximum classification accuracy of about 65 %, while the modified method has increased accuracy by about 10 percentage points. The p-score coefficient (Student's t-test) for individual values of L parameter equals: 8.66×10^{-6}; 1.02×10^{-8}; 7.84×10^{-4}, 4.26×10^{-2}; 4.27×10^{-2}; 2.08×10^{-3} respectively. For all cases of comparison of the accuracy of the "Proposed Approach" to the "Classic Approach", a significant statistical difference is observed(p<0.05). A comprehensive summary of the average classification accuracy for particular groups of gestures is presented in Table 1. The presented data are related to the average of a total 15 iterations (3-fold cross-validation repeated 5 times).

Table 1. Classic and Proposed approach gesture classification results. The table presents the mean values and associated standard deviation.

Classic approach accuracy[%]						
Gesture \ L	3	4	5	6	7	8
Turn Right	65.0 ± 8.2	37.8 ± 3.1	84.2 ± 1.8	55.9 ± 17.8	60.0 ± 3.6	44.9 ± 4.8
Sit Down	16.4 ± 3.6	21.4 ± 2.0	48.9 ± 24.9	54.3 ± 23.2	23.7 ± 7.2	31.0 ± 10.4
Turn Over	61.3 ± 10.8	38.6 ± 6.3	44.8 ± 29.6	29.4 ± 5.5	28.8 ± 20.8	34.3 ± 13.3
Come	42.6 ± 6.1	71.7 ± 4.7	86.1 ± 3.1	86.7 ± 5.4	90.9 ± 4.5	85.9 ± 10.3
Proposed approach accuracy[%]						
Gesture \ L	3	4	5	6	7	8
Turn Right	65.0 ± 8.2	52.2 ± 8.6	83.3 ± 11.9	61.7 ± 20.5	60.7 ± 16.3	44.7 ± 28.9
Sit Down	15.6 ± 3.1	25.3 ± 7.8	47.3 ± 32.7	45.0 ± 31.8	22.3 ± 11.5	36.9 ± 25.8
Turn Over	100.0 ± 0.0	60.6 ± 2.1	97.2 ± 2.1	60.6 ± 2.1	72.3 ± 30.1	79.1 ± 17.2
Come	42.4 ± 6.1	70.4 ± 7.8	85.6 ± 3.9	91.9 ± 3.4	71.9 ± 15.9	80.9 ± 16.9

From the presented data, it can be stated that the maximum recognition accuracy is observed for different values of the parameter L for each gesture. For the classic discretization method, the "Turn Right" and "Turn Over" gestures classification accuracy is lower than the "Turn Right" or "Come" motions. In the proposed approach, "Turn over" gestures recognition accuracy has significantly increased. At the same time, recognition of the remaining gestures has not changed significantly. No significant decrease in the recognition of "Turn Right" or "Come" motion patterns was observed. On the other hand, the accuracy of "Sit Down" gesture recognition remained low.

4. Conclusions

As a part of the work, the author developed a comprehensive algorithm for recognizing selected HCI gestures registered with the inertial sensors. The paper uses author's data corpus containing the signals representing a set of four gestures described as "Come", "Turn Right", "Turn Over" and "Sit Down" (20 participants, 720 timeseries). As a consequence of the conducted experiments it was found that the methodology described in the literature [2] and dedicated to the recognition of VR gaming gestures cannot be directly applied to HCI gestures datasets. In the case of using the classic methods on the author's data, the classification accuracy of approximately 65% was obtained.

The paper proposes modification of codebooks generating algorithm, in particular limiting the number of states generated from the *Roll* angle of forearm rotation.

In this study an uneven division of Euler's angles was proposed, in which the state of *Roll* angle assumed only one of three values. As a consequence of the changes, the classification accuracy increased about 10 % points in comparison with the results obtained with the classic algorithm (Fig. 5).

This work provides comprehensive information about the impact of Euler angle discretization on the classification accuracy. It should be emphasized that regardless of the L parameter value, for the new algorithm of codebook generation a higher average accuracy of classification was obtained (in relation to the classic method). Despite the effort of modifying the codebook algorithm, the presented approach is still not universal. The accuracy of the "Sit Down" gesture classification can be considered as insufficient. Therefore, optimization of the algorithm will be the subject of further experiments.

References

[1] Wu, Y., Chen, K., Fu,C.: Natural Gesture Modeling and Recognition Approach Based on Joint Movements and Arm Orientations, IEEE Sensors Journal, Volume: 16, Issue: 21, 2016.

[2] Arsenault, D., Whitehead, A.D.: Gesture recognition using Markov Systems and wearable wireless inertial sensors, IEEE Transactions on Consumer Electronics, Volume: 61, Issue: 4, 2015.

[3] Elmezain, M., Al-Hamadi, A., Michaelis, B.: A hidden markov model-based isolated and meaningful hand gesture recognition, International Journal of Electrical, Computer, and Systems Engineering 3.3: 156-163, 2009.

[4] Elmezain, M., Al-Hamadi A., Michaelis, B.: Hand Gesture Spotting Based on 3D Dynamic Features Using Hidden Markov Models, Communications in Computer and Information Science, vol 61. Springer, Berlin, Heidelberg, 2009.

[5] Di Benedetto A., Palmieri F.A.N., Cavallo A., Falco P.: A Hidden Markov Model-Based Approach to Grasping Hand Gestures Classification, Advances in Neural Networks. WIRN 2015. Smart Innovation, Systems and Technologies, vol 54. Springer, Cham, 2016.

[6] Georgi, M.; Amma, C.; Schultz, T.: Recognizing Hand and Finger Gestures with IMU based Motion and EMG based Muscle Activity Sensing, BIOSTEC 2015 Proceedings of the International Joint Conference on Biomedical Engineering Systems and Technologies - Volume 4, 2015.

[7] Amma, C., Georgi, M.,Schultz, T.: Airwriting: a wearable handwriting recognition system Personal and Ubiquitous Computing, Volume 18, Issue 1, 2014.

[8] Chen, C., Jafari, R., Kehtarnavaz, N.: A Real-Time Human Action Recognition System Using Depth and Inertial Sensor Fusion 6th International Workshop on Advances inSensors and Interfaces (IWASI), 2015.

[9] Comotti, D., Caldara, M., Galizzi, M., Locatelli, P., Re, V.: Inertial based hand position tracking for future applications in rehabilitation environments IEEE SENSORS JOURNAL, VOL. 16, NO. 3, FEBRUARY 1, 2016.

[10] Li, Q., Wang, Y., M., Sharf, A., Cao, Y., Tu, C., Chen, B., Yu, S.: Classication of gait anomalies from kinect The Visual Computer,vol. 34, no. 2, 2018.

ROZPOZNAWANIE GESTÓW INTERAKCJI CZŁOWIEK-KOMPUTER ZAREJESTROWANYCH PRZY UŻYCIU INERCYJNYCH CZUJNIKÓW RUCHU POPRZEZ NIEJAWNE MODELE MARKOVA

Streszczenie Artykuł przedstawia algorytm rozpoznawania wybranych gestów interakcji człowiek-komputer zarejestrowanych przy pomocy inercyjnych czujników ruchu. W niniejszej pracy zweryfikowano możliwości wykorzystania niejawnych Modeli Markova jako klasyfikatora. Zbadano możliwość zastosowania metodyki dedykowanej rozpoznawaniu gestów gry VR do klasyfikacji gestów HCI. W pracy dokonano porównania skuteczności klasyfikacji w zależności od sposobu dyskretyzacji zarejestrowanych sygnałów orientacji przedramienia. Ocena skuteczności klasyfikacji odbyła się z wykorzystaniem trójkrotnej walidacji krzyżowej. W pracy wykorzystano autorski korpus danych zawierający 20 uczestników oraz łącznie 720 szeregów czasowych.

Słowa kluczowe: HMM, Klasyfikacja, HCI, IMU

The work was supported by grant WI/WI/1/2019 from Białystok University of Technology and funded with resources for research by the Ministry of Science and Higher Education in Poland. Funded by the PROM Project: “International scholarship exchange of PhD candidates and academic staff” within the Operational Programme Knowledge Education Development, co-financed from the European Social Fund. The author is grateful to Sławomir Krzysztof Zielinski for substantive contribution.

A SHORT SURVEY ON FULLY-AUTOMATED PEOPLE MOVEMENT AND IDENTITY DETECTION ALGORITHMS

Maciej Szymkowski[1], Karol Przybyszewski[1]

Faculty of Computer Science, Bialystok University of Technology, Białystok, Poland

Abstract: Nowadays, diversified companies use security systems based on cameras to increase safety of their enterprise. However, when the camera observes multiple people, it is hard for humans to directly observe each of them. In the literature, there are multiple computer vision-based approaches that automatically detect person identity and the way he is moving. Moreover, there are approaches that identify people across multiple cameras (re-identification). It is crucial, especially in the crowded places. By these algorithms we can detect people whose behavior is strange. Diversified approaches can be easily found in the literature and online-available repositories. The work, presented in this paper, can be divided into three main parts: literature review, selected algorithms implementation and results comparison. We have to claim that each solution was implemented in Python programming language with sufficient libraries. This technology was selected due to its efficiency and simplicity. Results of the conducted experiments have shown that it is clearly possible to detect people's movement and observe their identities even in crowded places.

Keywords: Computer Vision, Image processing, Artificial Intelligence, People movement, Identity detection, Person Re-Identification, Python programming Language

1. Introduction

Nowadays, one of the most important trends in Computer Science is connected with Computer Vision. This technique allows computers to see, identify and process images in the same way as human vision does. The most common algorithms include video and image analysis. In both of these areas, the main goal is to detect a particular object or recognize its identity (as in biometrics algorithms).

Object tracking has been an active field of research for many years. It can be used in video analysis. Its main goal is to keep track of an object's position along the frames, allowing it to preserve its identity. There are plenty of tracking algorithms,

Advances in Computer Science Research, vol. 15, pp. .., 2021. DOI: 10.24427/acsr-2021-vol15-0002

but the main process is to predict the selected object's position in the next frame. This operation is usually less complex (in the terms of computational complexity) than object detection performed in each frame. Multiple cameras provide additional complexity to this algorithm. It is connected with the fact that the solution has to match the same person on many video streams at the same time.

Another interesting approach is connected with direct frame per frame analysis. This method can be compared with the "brute force" algorithm used in password analysis. The similarity can be easily observed due to a couple of facts. The first of them is the high computational complexity of the approaches based on this idea (the same is observed in "brute force" method for password analysis). High usage of resources will not be observed when we analyze one frame, although when the operation is repeated multiple times on a huge amount of frames then the complexity is easily observable. Moreover, if we want to get additional information about the object (for example its identity) we need to use supplementary methods. Probably we will need to connect information from a couple of frames rather than using only one of them.

If it comes to selection of the tracking method from described previously, we can observe that neither of them can guarantee satisfactory results in exceptionally short time. For this aim, we should use another solution that is motion detection. In the case of this approach, we can observe high speed up (in comparison to frame per frame analysis and simple object tracking). We will only observe moving parts of the video, the scene as a whole will not be analyzed (only some specific areas with moving objects, the rest of it will be classified as background). It also should be claimed that there is no single perfect strategy that can guarantee us satisfactory results in a selected environment. In most of the cases, we need to conduct a huge amount of experiments by which we select the proper algorithm.

The described methods can allow us to point some specific goals of Computer Vision. The most important of them is that this technique provides tools used not only to observe selected objects but also to process and return additional information on the basis of observations. Well-known example is a car onboard system for road signs detection and recognition.

The main goal of our work was to find out which algorithms are recently used for movement and identity detection and which of them can guarantee the results in safisfactory short time. We implemented and compared the selection of them. Each experiment was performed on diversified videos. Most samples were collected from DGait [1] and CMU Panoptic Dataset [2] databases.

This work is organized as follows: in the first section the authors describe diversified approaches connected with detection of people movement and tracking. In

the second one, some information about the comparison results are presented. Finally conclusions and future work are given.

2. Related work

Recent advancements in the area of deep learning resulted in transferring those methods into various topics of computer vision. With the success of CNNs [3,4], deep features learned from the networks has replaced handcrafted features [5] for representing person images. One of the important topics is person re-identification (ReID). Given an image of a person captured on one camera, the task is to identify this person from the gallery set captured by other multiple cameras. Table 1 [6] clearly shows an increasing number of papers where deep learning methods were used to develop ReID methods. Table presents number of papers presented at three top conferences:

- CVPR - Conference on Computer Vision and Pattern Recognition [7],
- ICCV - International Conference on Computer Vision [8],
- ECCV - European Conference on Computer Vision [9].

Table 1. The number of deep learning papers related to person ReID included by the three top conferences in recent years.

	2014	**2015**	**2016**	**2017**	**2018**
CVPR	1	2	5	7	25
ICCV	n.a.	2	n.a.	2	n.a.
ECCV	0	n.a.	2	n.a.	18

Due to the growing efficiency of deep learning methods, we decided to review only papers where those methods were used. Table 2 [6] presents growing accuracy (Rank 1) of the state-of-the-art deep learning models on popular person ReID datasets over the recent years:

- **CUHK03.** The dataset is one of the largest ReID datasets which contains 13,164 images of 1360 identities. All identities are taken from six camera views, and each pedestrian is captured by two cameras. This data set provides two settings. One automatically annotated by a detector and the other manually annotated by humans. Among the two settings, the former is closer to practical scenarios [10].
- **Market-1501.** This dataset consists of 32,643 annotated boxes of 1501 persons. Each pedestrian is collected by at least two cameras and at most six cameras from the front of a supermarket. The boxes of pedestrians are captured by the Deformable Part Model (DPM) detector [11].

– **PRID-2011.** The images of this dataset are captured from two non-overlapping surveillance cameras. One camera captures 749 pedestrians and the other camera captures 385 pedestrians. Among these pedestrians, 200 persons recorded in both cameras. All images are cropped into 128 ×48 pixels. Different from other datasets, PRID 2011 is captured in a relatively clean and simple scene and the dataset has consistent illumination changes [12].

Table 2. Accuracy (Rank 1) of the state-of-the-art deep learning models on popular person ReID datasets over the recent years.

	2016	**2017**	**2018**
CUHK03	85,4	88,5	94,9
PRID 2011	66,8	83,7	93,0
Market-1501	83,7	84,9	93,6

Video-based person Re-ID is an extension of image-based person Re-ID. Zheng et al. [13] introduce a large-scale dataset to enable the learning of deep features for video-based Re-ID. They first train a CNN to extract image features then aggregate them into a sequence features with average/maximum pooling. Other works [14] adopt Recurrent Neural Networks to summarize image-wise features into a single feature by exploiting temporal relation within a sequence.

2.1 Capsule networks

Convolutional neural networks (CNNs) have had great success in solving problems with object recognition and classification. However, they are not perfect. If at the input of the convolutional network we give an object in an orientation that the network does not know, or in which objects appear in places that the network is not used to, the prediction task will likely fail. CNN learns statistical patterns on images, but not the basic concepts of what makes something actually look like a specific real object (e.g. a face).

In 2017, Geoffrey Hinton (and others), borrowed ideas from neurobiology that suggest that the brain is organized into modules called capsules [15] (CapsNets). These capsules are particularly good at recognizing features such as orientation (position, size, orientation), deformation, speed, albedo, hue, texture, etc. In the context of neural networks, capsules are represented by groups of neurons.

The results presented in Hinton's work showed that CapsNets had the highest performance in standard datasets such as MNIST [16] (with a test accuracy of

99.75%) and SmallNORB [17] (with a 45% error reduction over the previous best result). However, the applications and performance of these networks on real and more complex data have not been fully verified. A very important benefit of capsule networks is the transition from black-box neural networks to those that represent more specific characteristics that can help us analyze and understand how the neural network works from the inside.

We should also observe that there are also additional approaches regarding neural networks for object recognition. The most interesting of them was presented in [18]. In this work, the Authors used deep convolutional neural network to classify images from ImageNet LSVRC-2010 competition dataset. What is interesting is that the neural network consisted of 60 million parameters and 650000 neurons. However, the results were not as accurate as expected, the Authors obtained 37,5% and 17% error rates in two testing sets (provided by the organizers of the competition). In the work it was also claimed that the same model was used in the dataset from the another LSVRC competition and the results were much better - error rates were equal from 15,3% to 26,2%. We can conclude that in this case also image quality can have a huge influence on the final results.

What is also interesting is that deep convolutional neural networks and artificial intelligence approaches in general can be used to detect some specific objects. This aim can be mainly observed in biometrics and medical images analysis algorithms. In the first case, neural networks are used to detect specific structures by which human identity can be recognized - for example these can be some parts of fingerprint, eg. not often observed minutiae. In the second idea, machine learning or artificial intelligence is used for detection and segmentation of some pathological changes. The most representative solutions are presented in works [19,20,21]. In these cases, neural networks were used to detect pathological changes in ophthalmic images (especially diabetic retinopathy).

2.2 Selected Computer Vision Surveys

The rapid development of image processing using neural networks has also resulted in a large number of scientific articles describing selected issues in this field. It is worth paying attention to the article "Object Detection in 20 Years: A Survey" [22], where the authors describe extensively over 20 years of history of object detection. The article is based on a review of over 400 papers covering the period from 1990 to 2019. The authors clearly show the division into two main eras of object detection: traditional detection methods (until 2012) and methods based on deep machine learn-

ing (after 2012), among which the most popular concepts are related to convolutional neural networks.

A very good and extensive work describing contemporary deep machine learning architectures is the article "A State-of-the-Art Survey on Deep Learning Theory and Architectures" [23]. The authors comprehensively describe the development of the most important concepts in the field of deep machine learning since 2012. The article also includes a list of the most popular frameworks, SDKs and reference data sets used to implement and evaluate tasks related to deep machine learning.

It is also worth paying attention to the work "A Survey of the Recent Architectures of Deep Convolutional Neural Networks" [24] which focuses on the history of the development of deep convolutional neural networks. The research focuses on showing the internal taxonomy of CNN's latest deep architectures. It also attempts to classify the latest innovations in CNN architectures into seven different categories (English terminology): spatial exploitation, depth, multi-path, width, feature map exploitation, channel boosting, and attention.

3. Conducted experiments

At the beginning of this section, we would like to present the main goal of the conducted experiments as well as the way in which selected algorithms were tested. In the next part of this subsection, comparison between selected methods is also presented.

The main goal of the experiments was to check whether it is possible, with currently used methods, to track people and to gain information about their identity (in biometrics terms). In this case we would like to differentiate people from each other. We do not have data regarding their real identities as well as we do not possess any biometrics databases connected with all analyzed samples.

3.1 Testing procedure

In this subsection we would like to describe how selected algorithms were tested and what was the way to calculate accuracy of the selected solutions. Each algorithm was implemented with Python Programming Language. Testing procedure was realized in the manner described below.

1. In the first step, the selected video was loaded with default Python tools to our algorithm.
2. As the next stage, we marked all people currently visible in the scene. We also differentiate them. Each of them was distincted with different colors.

3. Further we observed whether the number of marked people was correct. Decision correctness was evaluated on the basis of comparison between the returned number and the number of people observed by the human operator.
4. In the next stage, we changed the scene to the next frame from the camera and observed whether each man has his own, correct marker. It means that we were checking whether marker color was preserved. Once again we compared the algorithm decision with the decision of the human operator.
5. Finally, we combined collected data and evaluated the algorithm. Final accuracy of the solution was calculated as in (1).

$$\eta = 1/|X| \cdot \sum_{x \in X} x_{accuracy} \tag{1}$$

where η is a final, generalized accuracy of the selected algorithm, X is a set of all observations (results of the selected algorithm on each data) whilst $x_{accuracy}$ is an accuracy of the observation x.

3.2 Tested methods and obtained results

In this part of the experiments we used three, most popular algorithms available online: YOLO (You Only Look Once) [25], COCO (Common Objects in COntext) [26] and the default tracking algorithm from OpenCV library [27]. Each of them was analyzed and implemented due to its simplicity and low computational complexity. We assumed that all analyzed solutions can be used in real-time monitoring systems. However, we observed broad differences in the obtained results. This implies that selection of the algorithm can have a huge influence on a final decision. The general scheme of the proposed system is presented in Fig. 1.

Our analysis will be started with presentation of the results obtained with implemented solutions. The captured frame with the marked person is presented in Fig. 2.

It is easily observable that the selected solution can mark a person when it is only one visible. Fig. 3, Fig. 4 and Fig. 5. present behavior of implemented algorithms when there are more people in the scene.

On the basis of the presented results, we can claim that it is clearly possible to observe not only the number of people in the scene but also differentiate them. However, sometimes, selected solutions returned results with mistakenly marked people (e.g. two of them were marked as one - in the terms of identity). This problem is clearly observable in Fig. 4 and Fig. 5. In the first of them, two people were marked as one (algorithm decided that their identities were the same). When it comes to Fig.

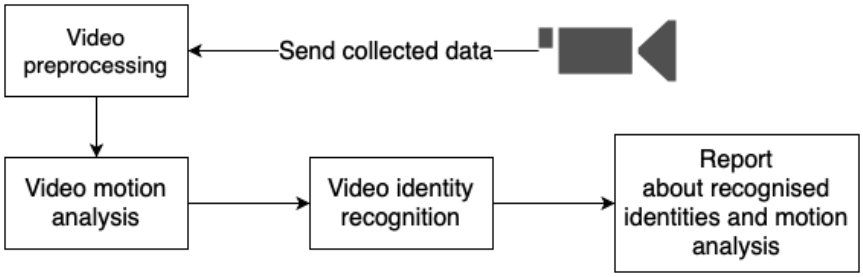

Fig. 1. General scheme of the proposed system.

Fig. 2. Image with person marked by implemented algorithms. Frame was taken from database [28].

5 we can observe that two people were not marked. It means that the analyzed solutions do not recognize them as human. These two mistakes were mostly observed. Other types of them mostly appeared only once.

Right now, we would like to present the results of comparison between all tested methods. We made experiments in regards to accuracy (number of people & identities). Of course, we applied slight modifications to have a possibility to obtain information about identities. It is connected with the fact that Standard OpenCV algorithm, YOLO and COCO are the solutions for object detection. Their main goal is to detect objects and to say what is it (man, book, computer... etc.). We applied

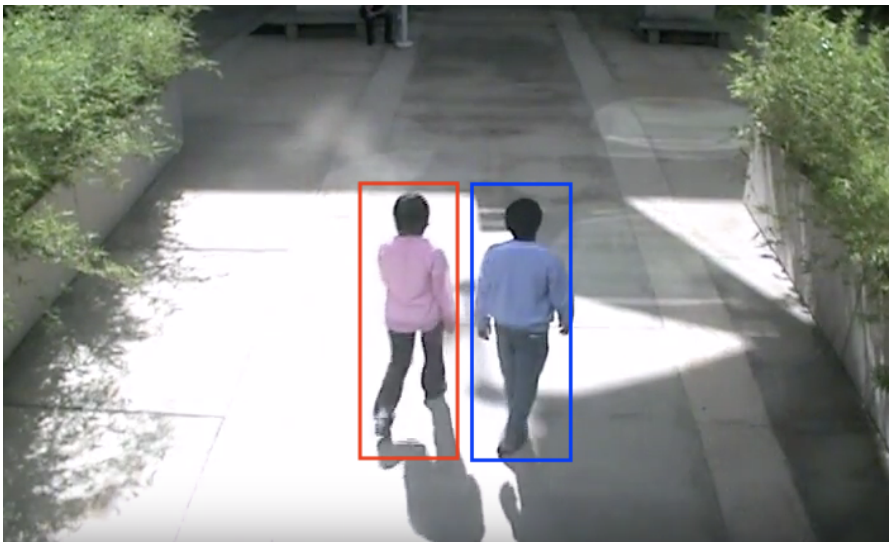

Fig. 3. Multiple people in the scene. Frame was collected from database [28].

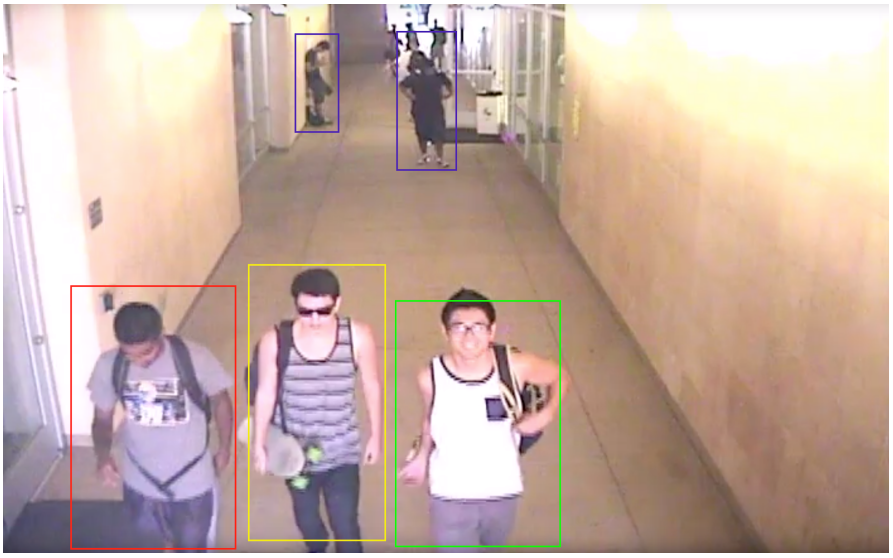

Fig. 4. Multiple people in the scene. Two people were recognized as one (marked with purple rectangle). Frame was taken from database [28].

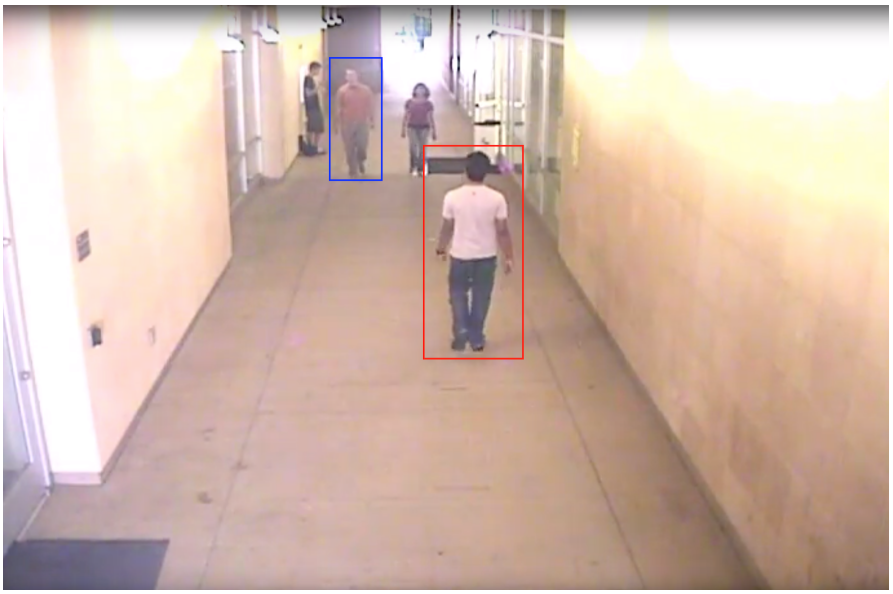

Fig. 5. Recognition of only two people whilst four appeared in the scene. Frame was captured from the database [28].

some additional changes (regarding information found in diversified guides) for people identification. Each method was tested on more than 100 samples. The results of the tests are presented in Table 3.

Table 3. The results of the experiments.

Algorithm	**Mean accuracy (number of people)**	**Mean accuracy (identities)**
Standard algorithm in OpenCV [27]	82,73%	70,2%
YOLO [25]	**89,95%**	**85,36%**
COCO [26]	88,92%	81,24%

On the basis of Table 3. It is easily observable that the standard algorithm in OpenCV cannot guarantee satisfactory results even in the case of counting the number of people. YOLO and COCO returned more precise results however neither of them can be used in real circumstances. It is connected with the fact that, in the real environment, the mistake cannot be higher than 0,5%. If we want to replace human

operators by fully-automated algorithms we have to have real confidence that the decision of the algorithm is always right. We cannot risk that it will often fail and the results will be uncertain.

When it comes to identity detection, we observed that each algorithm once again does not return satisfactory results. Once again, the best was YOLO that reached around 85% of correct recognition rate. We observed that the most serious problems with identity detection were generated when the object was partially visible as well as when the light caused its distortion. We think that these problems can be reduced with some additional preprocessing methods by which distortions can be removed and our video frame will be clearer.

4. Conclusions and Future Work

On the basis of the conducted experiments we have to claim that even if it is possible to track people movement and identify them with recently available solutions and tools, the perfect object detection and tracking algorithm has still to be found. The main problem with all analysed approaches is connected with a combination of high speed and high accuracy.

One possible solution is to combine detection and tracking approaches. Analysis of every n-th frame (n has to be selected experimentally) with computationally expensive detection can allow us to update trackers. With this operation we will probably gain a solution that has less computational complexity and can guarantee results similar to the ones obtained with the analyzed approaches. This idea will be implemented and tested in the nearest future.

The Authors' current work is to optimize selected solutions in the terms of time and computational complexity. Moreover, we are working under our own solution for people movement and identity recognition. The experiments in this case are performed on our own database. In the incoming future, we would like to prepare our own fully-automated system based on embedded hardware and FPGAs for people movement and identity detection. We see a huge potential in embedded systems and IoE (Internet of Everything) solutions (with proper cameras and maybe DSP (Digital Signal Processing) modules to obtain precise results in less time.

The performed survey and conducted experiments have shown us that there is still an unresolved gap of problems in video and image processing as well as in object recognition. Moreover, we would like to work under our own re-identification algorithm that will be based on one of the solutions described in this paper. However the selected base has to be perfectly tuned before creation of the final system.

Acknowledgment

This work was partially supported by works WZ/WI-IIT/4/2020 and W/WI-IIT/2/2019 from Białystok University of Technology and funded with resources for research by the Ministry of Science and Higher Education in Poland.

References

[1] http://www.cvc.uab.es/DGaitDB/Summary.html (Access 15.08.2019)
[2] http://domedb.perception.cs.cmu.edu/index.html (Access 15.08.2019)
[3] Hermans A., Beyer L., and Leibe B.: In defense of the triplet loss for person re-identification, arXiv preprint arXiv:1703.07737, 2017.
[4] Li W., Zhu X., and Gong S.: Harmonious attention network for person re-identification, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2285–2294, 2018.
[5] Cheng D.S., Cristani M., Stoppa M., Bazzani L., and Murino V.: Custom pictorial structures for re-identification, In Bmvc, volume 1, p. 6., 2011.
[6] Wu D., Zheng S.J., Zhang X.P., et al.: Deep learning-based methods for person re-identification: A comprehensive review, Neurocomputing 337, pp. 354–371, https://doi.org/10.1016/j.neucom.2019.01.079, 2019.
[7] http://cvpr2019.thecvf.com/ (Access 10.11.2019)
[8] http://iccv2019.thecvf.com/ (Access 10.11.2019)
[9] https://eccv2020.eu/ (Access 10.11.2019)
[10] Li W., Zhao R., Xiao T., and Wang X.: Deepreid: Deep filter pairing neural network for person re-identification, In Proc. CVPR, 2014.
[11] Zheng L., Shen L., Tian L., Wang S., Wang J., and Tian Q.: Scalable person re-identification: A benchmark, In Proc. ICCV, 2015.
[12] Hirzer M., Beleznai C., Roth P.M., Bischof H.: Person re-identification by descriptive and discriminative classification, in: Proceedings of the Scandinavian Conference on Image Analysis, pp. 91–102, 2011.
[13] Zheng L., et al.: MARS: A Video Benchmark for Large-Scale Person Re-identification, In: European Conference on Computer Vision. Springer, Cham, pp 868-884, 2016.
[14] Zhou Z., Huang Y., Wang W., Wang L., and Tan T.: See the forest for the trees: Joint spatial and temporal recurrent neural networks for video-based person re-identification, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4747–4756, 2017.

[15] Sabour S., Frosst N. , Hinton G.E.: Dynamic routing between capsules, In Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17), Curran Associates Inc., Red Hook, NY, USA, 3859–3869, 2017.

[16] LeCun Y., Bottou L., Bengio Y., Haffner P.: Gradient-based learning applied to document recognition, In Proceedings of the IEEE, vol. 86, 2278–2324, 1998.

[17] LeCun Y., Huang F.J., Bottou L.: Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting, In Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2004

[18] Krizhevsky A., Sutskever I., Hinton G.E.: ImageNet classification with deep convolutional neural networks, In Commun. ACM 60, vol. 6, pp. 84–90. DOI: https://doi.org/10.1145/306538, 2017.

[19] Wang S., Liang Y., Zhang Y.: Deep Convolutional Neural Networks for Diabetic Retinopathy Detection by Image Classification, In Computers & Electrical Engineering, vol. 72, pp. 274-282, 2018.

[20] Pratt H., Coenen F., Broadbent D.M., Harding S.P., Zheng Y.: Convolutional Neural Networks for Diabetic Retinopathy, In Proceedings of International Conference on Medical Imaging, Understanding and Analysis 2016, Loughborough, United Kingdom, pp. 1-6, 2016.

[21] Sarki R., Michalska S., Ahmed K., Wang H., Zhang Y.: Convolutional neural networks for mild diabetic retinopathy detection: an experimental study, DOI: https://doi.org/10.1101/763136, bioRxiv, 2019.

[22] Zou Z., Shi Z., Guo Y., Ye J.: Object detection in 20 years: A survey, arXiv preprint arXiv: 1905.05055, 2019.

[23] Alom M., Zahangir T.M., Taha M., et. al.: A state-of-the-art survey on deep learning theory and architectures, Electronics vol. 8, no. 3: 292, 2019.

[24] Khan A., Anabia S., Umme Z., Aqsa Saeed Q.: A survey of the recent architectures of deep convolutional neural networks, Artificial Intelligence Review, pp. 1-62, 2019.

[25] Redmon J., Farhadi A.: Yolo9000: Better, Faster, Stronger, In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263-7271, 2017.

[26] COCO Algorithm, https://algorithmia.com/algorithms/deeplearning/ObjectDetectionCOCO (Access 21.08.2019)

[27] https://www.pyimagesearch.com/2018/07/30/opencv-object-tracking/ (Access 21.08.2019)

[28] Zhang S., Staudt E., Faltemier T., Roy-Choudhury A.: A Camera Network Tracking (CamNeT) Dataset and Performance Baseline, In IEEE Winter Conference on Applications of Computer Vision, Waikoloa Beach, Hawaii, 2015.

ANALIZA ALGORYTMÓW SKORELOWANYCH Z DETEKCJĄ RUCHU OSÓB I ICH TOŻSAMOŚCI

Streszczenie Współcześnie w wielu miejscach publicznych oraz obszarach zajmowanych przez zróżnicowane firmy możemy zauważyć systemy bezpieczeństwa bazujące na kamerach. Jednakże bardzo ciężko jest pojedynczemu operatorowi obserwować każdą osobę, która pojawi się na obrazie. W tym celu powstały algorytmy bazujące na metodyce Computer Vision, które mają na celu wykrycie nie tylko trasy poruszania się każdej osoby ale również ocenę jej tożsamości. Co więcej tego typu rozwiązania mogą być bardzo przydatne w zatłoczonych miejscach, gdzie niezwykle ważne jest wykrycie niestandardowego zachowania poszczególnych osób. W literaturze oraz bazach dostępnych online możemy znaleźć zróżnicowane podejścia do rzeczonego problemu. W ramach naszej pracy porównujemy kilka z nich. Każde z wybranych rozwiązań zostało zaimplementowane przy użyciu języka Python i bibliotek dostępnych w ramach rzeczonego języka. To środowisko zostało wybrane ze względu na jego wydajność oraz prostotę pisania kodu. Wyniki, które uzyskaliśmy wskazują na to, że aktualnie istniejące solucje mogą być używane do obserwacji trasy poszczególnych osób nawet w zatłoczonych miejscach.

Słowa kluczowe: Przetwarzanie obrazów, Sztuczna inteligencja, Detekcja ruchu, Wykrywanie tożsamości, Język programowania Python

Praca została zrealizowana częściowo na mocy środków pochodzących z pracy WZ/WI-IIT/4/2020 oraz W/WI-IIT/2/2019 przyznanej przez Politechnikę Białostocką z funduszy na badania Ministerstwa Nauki i Szkolnictwa Wyższego w Polsce.