НЕЙРОІНФОРМАТИКА ТА ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ

NEUROINFORMATICS AND INTELLIGENT SYSTEMS

UDC 004.93

DETERMINATION AND COMPARISON METHODS OF BODY POSITIONS ON STREAM VIDEO

Bilous N. V. – PhD, Associate professor, Professor of the Software Engineering Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Ahekian I. A. – Senior Lecturer of the Software Engineering Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

Kaluhin V. V. – Master of the Software Engineering Department, Kharkiv National University of Radio Electronics, Kharkiv, Ukraine.

ABSTRACT

Context. One of the tasks of computer vision is the task of determining the human body in the image. There are many methods to solve this problem, some are based on specific equipment (motion capture, kinect) and provide the highest accuracy, some give less accuracy but do not require additional equipment and use less computing power. But usually, such equipment has a high cost, so to ensure the low cost of developments designed to determine the body in the image, you should develop algorithms based on computer vision technology. These algorithms can then be applied to various fields to analyze and compare body positions for a variety of purposes.

Objective. The aim of the work is to study the effectiveness of existing libraries to determine the human body position in the image, as well as methods for comparing the obtained poses in terms of speed and accuracy of determination.

Methods. A set of libraries and pose comparison algorithms were analyzed for the purpose of developing a system for determining the correctness of exercise by the user in real time. OpenPose, PoseNet and BlazePose libraries were analyzed for their suitability in recognizing and tracking body parts and movements in real-time video streams. The advantages and disadvantages of each library were evaluated based on their performance, accuracy, and computational efficiency. Additionally, different pose comparison algorithms were analyzed. The effectiveness of each algorithm was evaluated based on their ability to accurately determine and compare body positions.

As a result, the combination of BlazePose and weighted distance method can achieve the best performance in pose recognition, with high accuracy and robustness across a range of challenging scenarios. The weighted distance method can be further enhanced with techniques such as L2 normalization and pose alignment to improve its accuracy and generalization. Overall, the combination of the BlazePose library and weighted distance methods offers a powerful and effective solution for pose recognition, with high F1 index.

Results. Existing models for determining poses have shown similar results in the quality of determination with a run-up of about 2%. When developing a cross-platform software product, the BlazePose library, which has an API for working directly in the browser and on mobile platforms, has a significant advantage in speed and accuracy. Also, as the library uses extended 33 keypoint topology it becomes applicable to a wider list of tasks. In the study of comparison methods, the greatest influence on the results was exerted by the quality of pose determination.

Conclusions. Among the methods of comparison, the method of weighted distances showed the best results. The speed of position determination is inversely proportional to the quality of determination and significantly exceeds the recommended value – 40ms.

KEYWORDS: computer vision, body position, keypoints, pose estimation, pose comparison, blazepose, mediapipe, tensorflow.

ABBREVIATIONS

CNN is a convolutional neural network;
RNN is a recurrent neural network;
OKS is a key points of the object;
PCK is probability of correct keypoint;
API is an application programming interface;
RAM is a random-access memory.

NOMENCLATURE

di is a Euclidean distance between the real key point and the estimated key point;

s is a scale: the area of the boundary field divided by the total area of the image;

DOI 10.15588/1607-3274-2023-2-6

k is a constant that is determined separately for each control point;

d is a Euclidean distance;

n is a dimension of the vectors;

x, y are the corresponding coordinates of the two vectors in the measurement plane *i;*

G and *F* are two vectors of poses compared after L2 normalization;

n is a number of defined control points;

F_{ck} is a value of the probability of finding the correct joint for the element number *k* of the vector F, F_{xy} and G_{xy}

INTRODUCTION

Body position is the alignment of body parts in relationship to one another at any given moment, so the task of determining a person's body position can be defined as the task of finding connection points on the human body, also known as key points – elbows, wrists, knees, and others.

Obtained key points then can be used for the skeletal representation of the human body (see Fig. 1). In this representation, the body is represented as a graph, where each node corresponds to a joint (key point) on the body, and the edges between the nodes represent the bones or limbs [1].

When recognizing pose, usually do not pay much attention to facial recognition, but only to the position of the head. Nevertheless, there are certain scenarios where face recognition can be useful as well. For example, in some applications such as surveillance or security, it may be important to detect the identity of a person based on both posture or body language and their face. There are several classic methods for face recognition that have been used over the years like Eigenface, Fisherface or Viola-Jones algorithms, and wavelet transform [2].

Figure 1 – An example of determining the body pose in the image

Although the purpose of this research is mainly focused on the analysis of the methods for poses determination and comparison, some approaches for face recognition can be used for this aim as well.

Vector-based approach for face analysis involves creating a numerical representation of a person's face in an image, known as a vector. The vector is calculated using various mathematical techniques that consider the shape and features of the face. One popular way of using this approach is face recognition software, where the vector for a given face can be compared to vectors from other faces to determine if they match.

3D model-based method for head position analysis involves creating a 3D model of a person's head and using it to determine the position of the head in a 2D image. The 3D model is created using a machine learning technique called a regression CNN, which is trained using examples of 3D models and their corresponding 2D projections. The approach is more accurate than the vector-based and is less affected by lighting and partial face closure [3]. However, it requires a calibrated camera and knowledge of the location of 3D points on the head, making it more complex to implement.

Modern methods for pose recognition tasks commonly rely only on "Deep Learning" technologies.

CNN can be used to extract features from the input image, and a following fully connected neural network predicts the pose. The CNN typically includes multiple layers that can learn to recognize and extract different features from the image, such as edges, corners, and textures. The fully connected neural network takes these features as input and produces an output that represents the predicted pose [4].

Another way is to use a RNN to predict the pose over time, by considering the temporal dependencies between frames in a video [5].

To evaluate the performance of the pose estimation algorithm several metrics can be used.

An MS COCO data set [6] is used to assess the quality of the pose definition, using the OKS indicator – match the key points of the object. It is calculated from the distance between the predicted points and the marked points, normalized on a human scale. A constant of scale and key point needed to equalize the importance of each key point: the location of the neck is more accurate than the location of the thigh.

$$OKS = \exp\left(-\frac{d_i^2}{2s^2k_i^2}\right). \quad (1)$$

In the above formula *di* is the Euclidean distance between the real key point and the estimated key point, *s* is the scale: the area of the boundary field divided by the total area of the image, *k* is a constant that is determined separately for each control point.

Constants for key points were calculated by a group of researchers with MS COCO.

Another common evaluation metric is PCK and its variant PCKh. PCK measures the percentage of correctly estimated keypoints within a certain distance threshold of the ground truth. PCK is often used in hand pose estimation tasks. The PCK score is computed for each keypoint separately and then averaged over all keypoints to get the final score.PCKh, on the other hand, is a variation of PCK that considers the scale of the person in

DOI 10.15588/1607-3274-2023-2-6

the image. It is defined as the percentage of keypoints whose predicted location is within a certain fraction of the head size distance from the ground truth keypoint [7].

However, these metrics cannot be used to compare different people in different images. If we try to use them for this purpose, the results will not accurately reflect the differences between the poses, as the metrics will not be able to account for variations in body size, shape, and position across different people [8]. The proposed means of comparison should consider all these factors.

After choosing the appropriate pose comparison algorithm the aspect of pose recognition library performance should be taken into account. It will directly impact the processing time required for each frame and thus the overall system's ability to keep up with the video feed in real-time [9].

The minimum frame rate for high-quality video display is 24 frames per second. Therefore, the processing time of each frame should take about 40ms for a complete analysis of the video stream. Analysis is possible even if the process takes more time, but in this case some number of frames will be lost, which will deny the possibility of analyzing fast movements.

1 PROBLEM STATEMENT

The purpose of this research is to identify the most effective combination of pose recognition library and pose comparison algorithm for accurately recognizing and comparing human poses to implement a system that determines the correctness of the exercises on streaming video. The research aims to evaluate the performance of different pose recognition library runtimes and pose comparison algorithms using the F1 index as the performance metric. Based on this the problem statement can be divided into several parts.

Firstly, it is necessary to investigate the literature that provides information on the existing libraries for pose recognition and their characteristics like speed, accuracy of determination, robustness to variations in lighting conditions, background clutter, and occlusions, etc. After that, the most appropriate library that will be used for further research should be selected.

The second step is to analyze the possible algorithms applicable to compare the obtained poses.

Finally, the F1 index as performance metric for selected pose recognition library and different pose comparison algorithms can be calculated. To achieve this goal an experimental application must be implemented and a dataset of poses with corresponding labels indicating which poses are similar must be collected.

Since the library analyzes a two-dimensional image and builds a skeleton in two-dimensional space, the poses will be defined as the same only if they are in the photo in the same angle.

As a result, it is necessary to obtain a set of recommendations for the use of the library for determining poses in the context of creating a system of real-time video analysis, as well as a comparative description of methods for comparing poses.

2 REVIEW OF THE LITERATURE

As a result of research into existing libraries for position determination, it was found that now there is a large list of available solutions. The differences between these libraries primarily lie in the type of model they use, the types of poses they can estimate, their performance on different types of input, etc. OpenPose, PoseNet and BlazePose as the most modern were selected for the review.

OpenPose has an API for python and a plugin for the Unity game engine. Inside, it uses multi-stream optimization, which speeds up image processing speed and accordingly finds more control points in the image in terms of streaming video. According to official documentation, OpenPose can identify 25 key points when assessing the body and legs, 2x21 key points when assessing the arms and 70 points when analyzing the face image [10].

The PoseNet library is based on the TensorFlow Light framework and can distinguish 17 key points in the image. An important detail to note is that the researchers developed both the ResNet and MobileNet PoseNet models. The ResNet model has a higher accuracy, but has a large size and many layers, while the MobileNet model is designed to work on mobile devices [11]. The library can be used in a large number of programming languages, namely Python, C ++, Java, Swift, Objective C and Javascript.

BlazePose is a lightweight convolutional neural network architecture for human pose estimation that is tailored for real-time inference on mobile devices. During inference, the network produces 33 body keypoints for a single person and runs at over 30 frames per second on most modern devices. These additional keypoints provide vital information about face, hands, and feet location with scale and rotation and makes it particularly suited to real-time use cases like fitness tracking and sign language recognition [12].

The main difference of this library is that the neural network uses both heat maps and regression to keypoint coordinates to estimate the body pose. At the same time the new 33 points topology (see Fig. 2) that is a superset of BlazeFace, BlazePalm, and MS COCO[6] allows the library to be consistent with the respective datasets and inference networks.

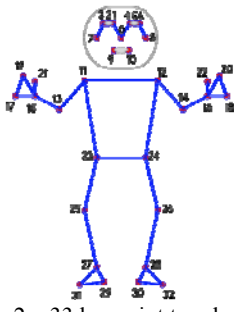

Figure 2 – 33 keypoint topology

DOI 10.15588/1607-3274-2023-2-6

Comparative results of testing the quality and performance of these libraries are shown in Table 1.

The table shows the obtained values of PCK as well as the frames per second values on two different datasets. The first dataset, referred to as AR dataset, contains a wide variety of human poses in the wild. The second is comprised of yoga/fitness poses only. As not all libraries support extended topology the MS COCO topology was used for consistency as the most common one. As an evaluation metric, the Percent of Correct Points with 20% tolerance (PCK@0.2) (where we assume the point to be detected correctly if the 2D Euclidean error is smaller than 20% of the corresponding person's torso size) was used.

Table 1 – The results of comparing performance and accuracy on different datasets

	FPS	AR Dataset	Yoga Dataset
OpenPose	0.4	87.8	83.4
BlazePose Full	10	84.1	84.5
BlazePose Lite	31	79.6	77.6

To solve the problem of variations in starting sizes and different positions of people in the frame when comparing poses, the input images can be preprocessed by resizing them to a fixed size, cropping them to a specific region of interest, and normalizing the pixel values. These techniques can help reduce the impact of image size and position differences and improve the accuracy of pose comparison [9].

Specifically, L2 normalization can be applied to the pose vectors to normalize the joint positions or joint angles. This technique can help to reduce the impact of variations in the magnitude of joint positions or angles, which can occur due to different camera perspectives and subject sizes [13].

When comparing poses, we need to determine the degree of similarity of vectors, because they will never be 100% identical. For this definition, use the concept of distance between vectors. The simplest and most classical way to determine it is the Euclidean distance, calculated by the formula:

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}. \quad (2)$$

In the above formula d is the Euclidean distance, n is the dimension of the vectors, x, y are the corresponding coordinates of the two vectors in the measurement plane i.

But due to the normalization of vectors, this method in its pure form loses its representativeness, because the reduction in image size directly affects the results of its calculation. therefore, we can use the concept of cosine similarity, which is the value of the cosine of the angle between the vectors, calculated by the formula:

$$\cos(a,b) = \frac{a \times b}{|a| \times |b|}. \quad (3)$$

Using the value of cosine similarity, we can calculate the value of the distance between the vectors by the formula:

$$D(F_{xy}, G_{xy}) = \sqrt{2 \times (1 - \cos ineSimilarity(F_{xy}, G_{xy}))}. \quad (4)$$

As a result, we obtain a value through which we can assess the similarity of the positions. The smaller this value, the more similar the poses. The resulting similarity ranges from −1, which means the exact opposite, to 1 means the same, and 0 indicates orthogonality or decorrelation, while the values between them indicate intermediate similarity or dissimilarity.

Another method is the method considering the probability of finding the correct control point. This probability can be provided by the recognition library, and it indicates the level of "confidence" that the joint is at a certain point and not at some other point. Sometimes we know exactly where the joint is, for example, if we can see it clearly; in other cases, we have very low confidence, for example, if the joint is cut or closed. Pre-filtering the image [11] can improve these values, but only in some cases. If we ignore these reliability metrics, we lose valuable information about our data, and can place much more weight and importance on those we are not sure about.

To use this information, researchers from Google George Papandreou and Tyler Zhu have developed a formula that considers the value of the reliability of the definition of a key point [14]:

$$D(F,G) = \frac{1}{\sum_{k=1}^{n} F_{c_k}} \times \sum_{k=1}^{n} F_{c_k} \times \left| F_{xy_k} - G_{xy_k} \right|. \quad (5)$$

In the above formula G and F are two vectors of poses compared after L2 normalization, n is the number of defined control points, F_{ck} is the value of the probability of finding the correct joint for the element number k of the vector F, F_{xy} and G_{xy} – x and y positions of the k-th key point for of each vector.

The third method [15] of comparison does not require prior normalization of coordinates, but the vector is built on a different principle. For every three anatomically connected points, the cosine of the angle between the obtained parts of the body by formula (1.8) is calculated. The difference from the cosine similarity used in the first method is that in this case a value is obtained for a two-

DOI 10.15588/1607-3274-2023-2-6

dimensional vector describing the position of the two limbs relative to each other.

3 MATERIALS AND METHODS

Based on the literature analysis, it was decided to use the BlazePose library for further research.

To perform the study, a Javascript script must be implemented, which defines poses from images using the BlazePose library and stores the obtained data in the MongoDB database. This database was chosen solely for ease of use to store non-relational data.

The next step is to implement the comparison methods described in the previous section and obtain comparison results. Each of the proposed methods compares all poses defined by a single library configuration. So, if we have 50 images and 3 library configurations, we have 150 defined poses, 50 for each configuration. With each of the three comparison methods, we compare 50 poses defined by one configuration. As a result of each comparison, an entry should be made in the database of compared poses, the method of determining poses, the method of comparing poses and the result, which is the value of the distance between two poses and takes a value from 0 to 1, where the smaller the value.

The last step is to calculate the F1 index [16]. The results obtained from the previous stage are grouped by the values of the configuration and the method of comparison. Poses are considered the same if the value obtained by comparison is less than the threshold value. Values from 0.05 to 0.40 with a step of 0.05 are taken as thresholds. An F1 index is calculated for each threshold value and for each group of comparison results.

Therefore, for each configuration of the pose determination method and for each pose comparison method, 8 values will be obtained that reflect the average correctness of determining whether the poses are the same or not for the 8 similarity thresholds.

Based on the results obtained, it will be possible to draw conclusions about which of the following methods of comparing poses provides greater accuracy. You can also evaluate which of the configurations of the BlazePose library provides a better quality of determining the pose for comparison, as well as determine the speed of its operation in different configurations.

The expected result is a recommendation on the configuration of the BlazePose library and the method of comparing poses, which will be the optimal context for the implementation of the system for comparing poses on streaming video in real time.

4 EXPERIMENTS

BlazePose can flexibly configure the model, which affects the speed and accuracy of its operation. It also includes 2 different runtimes: TensorFlow.js and MediaPipe, the first of which provides the flexibility and wider adoption of JavaScript, optimized for several backends including WebGL (GPU), WASM (CPU), and Node. MediaPipe capitalizes on WASM with GPU accelerated processing and provides faster out-of-the-box inference speed. When analyzing the comparison methods, different network configurations were used to obtain a description of the pose from the image.

In particular, the network architecture, the model has the following settings. Input resolution refers to the size of the image fed into the model for pose estimation. The default input resolution is 257x257, but this can be changed to lower or higher resolutions. The higher the resolution, the more accurate the pose estimate will be, but also slower to process. The lower the resolution, the faster the processing time but with lower accuracy. The minimum confidence score sets the threshold for accepting a predicted joint. A higher minimum confidence score will result in fewer, but more accurate joints, while a lower score will result in more, but less accurate joints. The confidence score is a value between 0 and 1, where 1 indicates high confidence and 0 indicates low confidence. Pose smoothing refers to the process of filtering the output of the model over time to produce a smoother, more stable result. The smoothing factor can be adjusted to control the amount of smoothing applied. A higher smoothing factor will produce a smoother result but may introduce a delay in the output. A lower smoothing factor will produce a more responsive output but may result in a less stable result. Also, BlazePose supports several different models -Lite, Full and Heavy, each with different accuracy and speed trade-offs.

It is necessary to investigate the difference in the definition of poses in the comparison and the difference in the speed of the network at different settings.

When comparing poses, methods are used to determine the distance between the vectors. Only the methods of vector construction and methods of calculating the distance differ. 3 methods were used for the study: cosine distance, weighted distance and distance at calculated angles. In the implementation of the first two methods, the vectors are built from the values of the coordinates for each key point on the human body. First, the coordinates are listed so that the starting point of the coordinates is not from the edge of the image, but from the edge of the rectangle surrounding the human body. The next step is L2 vector normalization. The distance between the obtained vectors will characterize the similarity of the poses in the image. The weighted distance method also considers the value of confidence, which indicates the accuracy of the obtained prediction of the position of the key point.

The analysis is performed according to the following algorithm: we obtain descriptions of poses using a neural network in different configurations, we obtain normalized vectors that are compared with each other using both comparison methods, the results are stored and used to calculate the F1-index. This indicator is calculated at different threshold values of the distance between the vectors: from 0.05 to 0.40 in steps of 0.05.

During the experiment, data on the processing speed and loading of these models were also collected. The

DOI 10.15588/1607-3274-2023-2-6

calculations are performed on a 3.2 GHz AMD Ryzen 7 processor, Nvidia GTX 1060 and 32GB of RAM.

5 RESULTS

The results of the study of the recognition speed collected during the study of the methods of comparison of poses are shown in Table 2. The results are given for such indicators as network load time and the average FPS during the process of poses determination. Studies of both speed and comparison methods were performed for each of the architectures with optimizations for speed and quality. As a result, the quality of work of four configurations of the library was analyzed.

The results clearly reflect that the performance of BlazePose can be greatly influenced by the hardware it is run on, such as the GPU or CPU. When running BlazePose on a GPU (MediaPipe Runtime), the computation is accelerated by the GPU's parallel processing capabilities. This allows for real-time processing of the input video frames, making it faster than running the same model on a CPU. At the same time, the load speed is the same, since the size of the model for different runtimes is also almost the same and does not affect this parameter.

Table 2 – The results of the study of the BlazePose model speed for different runtimes and configurations

	MediaPipe Runtime		TensorFlow.js Runtime	
	High quality	High speed	High quality	High speed
Load	4.82 c	1.91 c	4.82 c	1.91 c
FPS	113	135	38	65

It should be noted that the average values are given, but during the study of the results it was found that the difference in the speed of recognition of different poses can reach 100%.

Table 3 shows the results of the comparison of poses grouped by library configurations for determining poses and methods of determination. The symbols CA, AD and WD indicate the results for the methods of comparing the cosine distance, the distance at the calculated angles and the weighted distance, respectively.

The results of comparing the speed of algorithms were not collected for reasons of expediency. The complexity of O-notation algorithms [17] is constant and the same for all three methods.

Among the methods of comparing poses, the best result was demonstrated by the method of weighted distances. This is because when comparing poses, more weight is given to points that have been detected with greater accuracy and thus the overall pose is compared more correctly, while the value of the details is leveled.

The main disadvantages of this approach to comparing poses are that the result directly depends on the shooting angle. Thus, the same poses may look different and vice versa at different camera positions. In order to reduce the impact of the angle, it is necessary to build three-dimensional models of poses and compare them. However, this method takes much longer and the accuracy of construction of three-dimensional models of poses on a two-dimensional image is much lower.

6 DISCUSSION

Studies have shown that when comparing poses, the quality of determining poses is of the greatest importance. However, the higher the recognition quality, the lower the speed, and high-speed video recognition is required.

Due to the use of modern methods of determining the pose, it is possible to implement such a project for poses with low detail, ie for those where significant differences from the original.

The best option in terms of speed and quality of determination in the study was the configuration of the BlazePose model, based on the MediaPipe runtime optimized for faster execution. BlazePose is optimized for speed and can run at over 100 fps on modern GPUs, making it well-suited for real-time applications. This library has been shown to achieve state-of-the-art performance on various benchmark datasets, including COCO and MPII. It can accurately detect keypoints even in challenging scenarios, such as when people are occluded or when they have similar poses. It's worth noting that the accuracy and performance of BlazePose can be influenced by several factors, such as the quality of the input data and the specific use case.

Among the methods of comparing poses, the best result was demonstrated by the method of weighted distances. This is due to the fact that when comparing poses, more weight is given to points that have been detected with greater accuracy and thus the overall pose is compared more correctly, while the value of the details is leveled.

The main disadvantages of this approach to comparing poses is that the result directly depends on the shooting angle. Therefore, the same poses may look different and vice versa at different camera positions. In order to reduce the impact of the angle, it is necessary to build three-dimensional models of poses and compare them.

Cameras with depth sensors can solve this problem. This solution is used in Microsoft Kinect technology [18]. A promising solution used by Apple in new mobile devices is the Lidar sensor [19]. By combining lidar data with BlazePose, it is possible to perform 3D human pose estimation, which can provide more information about the position and orientation of people in space.

DOI 10.15588/1607-3274-2023-2-6

Table 3 – The results of the comparison of poses grouped by library runtimes and configurations for determining poses and methods of determination

	MediaPipe Runtime						TensorFlow.js Runtime					
	High quality			High speed			High quality			High speed		
	CD	AD	WD	CD	AD	WD	CD	AD	WD	CD	AD	WD
$F_1^{0.05}$	52.467	57.845	67.638	53.922	51.504	51.363	0	13.42	20.815	13.516	13.221	14.706
$F_1^{0.10}$	58.492	63.02	65.102	61.788	57.831	65.637	0	26.07	31.74	23.871	23.166	25.08
$F_1^{0.15}$	59.278	63.63	67.918	68.172	66.6	71.136	36.548	37.73	40.94	29.103	31.707	35.112
$F_1^{0.20}$	63.809	64.895	74.546	69.54	67.155	73.242	34.184	39.38	42.32	31.283	38.844	42.636
$F_1^{0.25}$	64.307	65.47	74.916	70.908	69.264	74.997	36.96	37.51	38.41	32.7	38.727	42.294
$F_1^{0.30}$	63.274	64.63	68.36	68.856	67.488	70.083	36.984	35.2	35.88	34.117	38.025	40.584
$F_1^{0.35}$	58.85	64.595	59.622	58.938	59.052	60.255	33.96	33.66	33.235	34.226	33.93	39.672
$F_1^{0.40}$	52.751	58.615	54.15	51.186	54.834	53.118	33.288	33.11	30.935	32.264	32.877	36.822

However, it's worth noting that lidar data can be more challenging to work with than traditional 2D image data. The data can be noisy and is often sparser, which can make it more difficult to accurately estimate 3D poses. Additionally, lidar data is typically collected from a different perspective than the cameras used by BlazePose, which can make it challenging to align the two data sources and estimate poses accurately.

CONCLUSIONS

As a result of the study, the existing methods of determining the poses in the image were analyzed in the context of developing a system for determining the correctness of the exercises in real time and created a comparative description of the methods of comparing poses.

An analysis of existing libraries for determining human body position in open-source images revealed that they have fairly similar values in terms of definition quality, but the BlazePose library has significant advantages in terms of its implementation in the system due to the wide support of programming languages.

Among the methods of comparing poses, the method of weighted distances showed the best results because it takes into account the value of the accuracy of determining the key point and when comparing gives more weight to those points that are found with greater accuracy.

In researching the capabilities of the BlazePose library, it was determined that the best results in terms of speed and quality of determination are provided by the library configuration based on the runtime of the neural network MediaPipe with optimization towards speed. BlazePose is optimized for real-time performance and can run at over 100 frames per second (fps) on modern GPUs, making it well-suited for applications that require fast and accurate human pose estimation.

In addition, BlazePose is designed to handle multiple people in an image or video, making it well-suited for scenarios where people are moving quickly and near each other.

ACKNOWLEDGEMENTS

The work is carried out in the framework of the scientific directions of the Software Engineering department, the research laboratory "Information Technologies in Learning and Computer Vision Systems" of the Kharkiv National University of Radio Electronics with the support of scientists from the Technical University of Applied Sciences Wildau and the Volkswagen Foundation.

REFERENCES

1. Zhipeng Z., Dong X., Shijie H. A Survey of Body Pose Estimation: Recent Advances and Future Prospects, *Journal of Imaging*, 2021, Vol. 7, No. 3, pp. 1–31. DOI: 10.3390/jimaging7030045
2. Shcherbakova G. Y., Krylov V. N., Bilous N. V. Methods of automated classification based on wavelet-transform for automated medical diagnostics, *2015 Information Technologies in Innovation Business Conference (ITIB).* Kharkiv, Ukraine, 7–9 October 2015, [S. l.], 2015. DOI: 10.1109/itib.2015.7355048
3. Rakova A. O., Bilous N. V. Research on Methods for Development of Software System for Face Orientation Vector Determining in the Image, *Radio Electronics, Computer Science, Control*, 2020, No. 3(54), pp. 121–129. DOI: 10.15588/1607-3274-2020-3-11
4. Shih-En W. Convolutional Pose Machines, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* Las Vegas, NV, USA, 27–30 June 2016. [S. l.], 2016. DOI: 10.1109/cvpr.2016.511
5. Kendall A. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization [Electronic resource] / Alex Kendall, Matthew Grimes, Roberto Cipolla // 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015. – [S. l.], 2015. DOI: 10.1109/iccv.2015.336
6. Tsung-Yi L. Microsoft COCO: Common Objects in Context, *Computer Vision – ECCV 2014.* Cham, 2014, pp. 740–755. DOI: 10.1007/978-3-319-10602-1_48
7. Yang Y., Ramanan D. Articulated Human Detection with Flexible Mixtures of Parts, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, Vol. 35, No. 12, pp. 2878–2890. DOI: 10.1109/tpami.2012.261

DOI 10.15588/1607-3274-2023-2-6

8. Cao Z. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019, P. 1. DOI: 10.1109/tpami.2019.2929257
9. Ge L. Real-Time 3D Hand Pose Estimation with 3D Convolutional Neural Networks, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019, Vol. 41, No. 4, pp. 956–970. DOI: 10.1109/tpami.2018.2827052
10. Liu Y. OpenPose-Based Yoga Pose Classification Using Convolutional Neural Network, *Highlights in Science, Engineering and Technology*, 2022, Vol. 23, pp. 72–76. DOI: 10.54097/hset.v23i.3130
11. Bilous N. V., Krasov A. I., Vlasenko V. P. Deletion method of image low-frequency components using fast median filter algorithm, *Journal of Engineering Sciences*, 2016, pp. 7–14. DOI: 10.21272/jes
12. Bazarevsky V., Grishchenko I., Raveendran K. BlazePose: On-device Real-time Body Pose tracking, *Computer Vision and Pattern Recognition*, 2020, P. 4. DOI: 10.48550/arXiv.2006.10204
13. Yu L., Gao Xiao-Shan Improve Robustness and Accuracy of Deep Neural Network with L2 Normalization, *Journal of Systems Science and Complexity*, 2022, pp. 1–26. DOI: 10.1007/s11424-022-1326-y
14. Friedhoff J. Move Mirror: An AI Experiment with Pose Estimation in the Browser using TensorFlow.js [Electronic resource]. Mode of access: https://blog.tensorflow.org/2018/07/move-mirror-ai-experiment-with-pose-estimation-tensorflow-js.html (date of access: 18.04.2023). Title from screen.
15. Borkar P. K., Pulinthitha M. M., Pansare A. Match Pose – A System for Comparing Poses, *International journal of engineering research & technology*, 2019, P. 3. DOI: 10.17577/IJERTV8IS100253
16. Dembczy´nski K., Waegeman W., Cheng W., H¨ullermeier E. An exact algorithm for F-measure maximization, *Neural Information Processing Systems*, 2011, P. 9.
17. Rutanen K. O-notation in algorithm analysis. Data Structures and Algorithms, 2022, P. 216. DOI: 10.48550/arXiv.1309.3210
18. Malmir B. Exploratory studies of Human Gait Changes using Depth Cameras and Sample Entropy [Electronic resource] : thesis. [S. l.], 2018. Mode of access: http://hdl.handle.net/2097/38949 (date of access: 18.04.2023). – Title from screen.
19. Bijelic M., Gruber T., Ritter W. A Benchmark for Lidar Sensors in Fog: Is Detection Breaking Down? *2018 IEEE Intelligent Vehicles Symposium (IV).* Changshu, 26–30 June 2018, [S. l.], 2018. DOI: 10.1109/ivs.2018.8500543

Received 14.04.2023.
Accepted 24.05.2023.

УДК 004.93

МЕТОДИ ВИЗНАЧЕННЯ ТА ПОРІВНЯННЯ ПОЛОЖЕНЬ ТІЛА НА ПОТОКОВОМУ ВІДЕО

Білоус Н. В. – канд. техн. наук, доцент, професор кафедри програмної інженерії, Харківський національний університет радіоелектроніки, Харків, Україна.
Агекян І. А. – старший викладач кафедри програмної інженерії, Харківський національний університет радіоелектроніки, Харків, Україна.
Калугін В. В. – магістр кафедри програмної інженерії, Харківський національний університет радіоелектроніки, Харків, Україна.

АНОТАЦІЯ

Актуальність. Однією з задач комп'ютерного зору є задача визначення тіла людини на зображенні. Існує багато методів вирішення цієї задачі, деякі базуються на специфічному обладнані (motion capture, kinect) та надають найбільшу точність, деякі дають меншу точність, але не потребують додаткового обладнання та використовують меншу обчислювальну потужність. Але зазвичай таке обладнання мав високу вартість, тож щоб забезпечити низьку вартість розробок створених для визначення тіла на зображенні, слід розробляти алгоритми за базі технологій комп'ютерного зору. Ці алгоритми можна застосовувати до різних областей для аналізу та порівняння положень тіла та досягнення різноманітних цілей.

Мета. Метою роботи є дослідження ефективності роботи існуючих бібліотек для визначення пози людини на зображенні а також методів порівняння отриманих поз з точки зору швидкості та точності визначення.

Методи. Дослідження проводяться в контексті розробки системи визначення правильності виконання фізичних вправ користувачем у режимі реального часу. Бібліотеки OpenPose, PoseNet і BlazePose були проаналізовані на предмет їх придатності для розпізнавання та відстеження частин тіла та рухів на відео у реальному часі. Переваги та недоліки кожної бібліотеки були оцінені на основі їх продуктивності, точності та обчислювальної ефективності. Крім того, були проаналізовані різні алгоритми порівняння поз. Ефективність кожного алгоритму оцінювалася на основі їх здатності точно визначати та порівнювати положення тіла.

У результаті поєднання BlazePose і методу зваженої відстані можно досягти найкращої продуктивності в розпізнаванні пози з високою точністю та надійністю в ряді складних сценаріїв. Метод зваженої відстані можна додатково вдосконалити за допомогою таких методів, як нормалізація L2 і вирівнювання пози для підвищення його точності та узагальнення. Загалом поєднання бібліотеки BlazePose та методів зваженої відстані пропонує потужне та ефективне рішення для розпізнавання пози з високим індексом F1.

Результати. Існуючі моделі визначення поз показали схожі результати якості визначення з розбігом близько 2%. При розробці крос-платформного програмного продукту значну перевагу в швидкості має бібліотека BlazePose, що має API для роботи безпосередньо в браузері та на мобільних платформах. Крім того, оскільки бібліотека використовує розширену топологію з 33 ключовими точками, вона може бути застосована для ширшого списку завдань. При дослідженні методів порівняння найбільший вплив на результати справила якість визначення пози.

Висновки. Серед методів порівняння найкращі результати продемонстрував метод зважених дистанцій. Швидкість визначення поз обернено пропорційна якості визначення і значно перевищує рекомендоване значення – 40мс.

DOI 10.15588/1607-3274-2023-2-6

КЛЮЧОВІ СЛОВА: комп'ютерний зір, положення тіла, ключові точки, визначення поз, порівняння поз, blazepose, mediapipe, tensorflow.

ЛІТЕРАТУРА

1. Zhipeng Z. A Survey of Body Pose Estimation: Recent Advances and Future Prospects / Zhang Zhipeng, Xu Dong, Hao Shijie // Journal of Imaging. – 2021. – Vol. 7, No. 3. – P. 1–31. DOI: 10.3390/jimaging7030045
2. Shcherbakova G. Y. Methods of automated classification based on wavelet-transform for automated medical diagnostics / G. Y. Shcherbakova, V. N. Krylov, N. V. Bilous // 2015 Information Technologies in Innovation Business Conference (ITIB), Kharkiv, Ukraine, 7–9 October 2015. – [S. l.], 2015. DOI: 10.1109/itib.2015.7355048
3. Rakova A. O. Research on Methods for Development of Software System for Face Orientation Vector Determining in the Image / A. O. Rakova, N. V. Bilous // Radio Electronics, Computer Science, Control – 2020. – No. 3(54). – P. 121–129. DOI: 10.15588/1607-3274-2020-3-11
4. Shih-En W. Convolutional Pose Machines / Wei Shih-En // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. – [S. l.], 2016. DOI: 10.1109/cvpr.2016.511
5. Kendall A. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization [Electronic resource] / Alex Kendall, Matthew Grimes, Roberto Cipolla // 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015. – [S. l.], 2015. DOI: 10.1109/iccv.2015.336
6. Tsung-Yi L. Microsoft COCO: Common Objects in Context / Lin Tsung-Yi // Computer Vision – ECCV 2014. – Cham, 2014. – P. 740–755. DOI: 10.1007/978-3-319-10602-1_48
7. Yang Y. Articulated Human Detection with Flexible Mixtures of Parts / Y. Yang, D. Ramanan // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2013. – Vol. 35, No. 12. – P. 2878–2890. DOI: 10.1109/tpami.2012.261
8. Cao Z. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields / Z. Cao // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2019. – P. 1. DOI: 10.1109/tpami.2019.2929257
9. Ge L. Real-Time 3D Hand Pose Estimation with 3D Convolutional Neural Networks / L. Ge // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2019. – Vol. 41, No. 4. – P. 956–970. DOI: 10.1109/tpami.2018.2827052
10. Liu Y. OpenPose-Based Yoga Pose Classification Using Convolutional Neural Network / Yuchen Liu // Highlights in Science, Engineering and Technology. – 2022. – Vol. 23. – P. 72–76. DOI: 10.54097/hset.v23i.3130
11. Bilous N. V. Deletion method of image low-frequency components using fast median filter algorithm / N. V. Bilous, A. I. Krasov, V. P. Vlasenko // Journal of Engineering Sciences – 2016. – P. 7–14. DOI: 10.21272/jes
12. Bazarevsky V. BlazePose: On-device Real-time Body Pose tracking / V. Bazarevsky, I. Grishchenko, K. Raveendran // Computer Vision and Pattern Recognition – 2020. – P. 4. DOI: 10.48550/arXiv.2006.10204
13. Yu L. Improve Robustness and Accuracy of Deep Neural Network with L2 Normalization / Lijia Yu, Xiao-Shan Gao // Journal of Systems Science and Complexity. – 2022. – P. 1–26. DOI: 10.1007/s11424-022-1326-y
14. Friedhoff J. Move Mirror: An AI Experiment with Pose Estimation in the Browser using TensorFlow.js [Electronic resource] / J. Friedhoff. – Mode of access: https://blog.tensorflow.org/2018/07/move-mirror-ai-experiment-with-pose-estimation-tensorflow-js.html (date of access: 18.04.2023). – Title from screen.
15. Borkar P. K. Match Pose – A System for Comparing Poses / P. K. Borkar, M. M. Pulinthitha, A. Pansare // International journal of engineering research & technology. – 2019. – P. 3. DOI: 10.17577/IJERTV8IS100253
16. An exact algorithm for F-measure maximization / [K. Dembczy´nski, W. Waegeman, W. Cheng, E. H¨ullermeier] // Neural Information Processing Systems. – 2011. – P. 9.
17. Rutanen K. O-notation in algorithm analysis / K. Rutanen // Data Structures and Algorithms. – 2022. – P. 216. DOI: 10.48550/arXiv.1309.3210
18. Malmir B. Exploratory studies of Human Gait Changes using Depth Cameras and Sample Entropy [Electronic resource] : thesis / Malmir Behnam. – [S. l.], 2018. – Mode of access: http://hdl.handle.net/2097/38949 (date of access: 18.04.2023). – Title from screen.
19. Bijelic M. A Benchmark for Lidar Sensors in Fog: Is Detection Breaking Down? / Mario Bijelic, Tobias Gruber, Werner Ritter // 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, 26–30 June 2018. – [S. l.], 2018. – DOI: 10.1109/ivs.2018.8500543

DOI 10.15588/1607-3274-2023-2-6