ISSN 2442-6571

Who danced better? Ranked tiktok dance video dataset and pairwise action quality assessment method

Irwandi Hipiny [a,1,*], Hamimah Ujir [a,2], Aidil Azli Alias [a,3], Musdi Shanat [a,4], Mohamad Khairi Ishak [b,5]

[a] Universiti Malaysia Sarawak, Jalan Datuk Mohammad Musa, 94300, Sarawak, Malaysia
[b] School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Nibong Tebal, 14300, Penang, Malaysia
[1] mhihipni@unimas.my; [2] uhamimah@unimas.my; [3] aaazli@unimas.my; [4] smusdi@unimas.my; [5] khairiishak@usm.my
* corresponding author

ARTICLE INFO

Article history
Received September 21, 2022
Revised January 13, 2023
Accepted January 21, 2023
Available online March 31, 2023

Keywords
Action quality assessment
Dance video dataset
Human activity analysis
String matching
Visual codebook

ABSTRACT

Video-based action quality assessment (AQA) is a non-trivial task due to the subtle visual differences between data produced by experts and non-experts. Current methods are extended from the action recognition domain where most are based on temporal pattern matching. AQA has additional requirements where order and tempo matter for rating the quality of an action. We present a novel dataset of ranked TikTok dance videos, and a pairwise AQA method for predicting which video of a same-label pair was sourced from the better dancer. Exhaustive pairings of same-label videos were randomly assigned to 100 human annotators, ultimately producing a ranked list per label category. Our method relies on a successful detection of the subject's 2D pose inside successive query frames where the order and tempo of actions are encoded inside a produced String sequence. The detected 2D pose returns a top-matching Visual word from a Codebook to represent the current frame. Given a same-label pair, we generate a String value of concatenated Visual words for each video. By computing the edit distance score between each String value and the Gold Standard's (i.e., the top-ranked video(s) for that label category), we declare the video with the lower score as the winner. The pairwise AQA method is implemented using two schemes, i.e., with and without text compression. Although the average precision for both schemes over 12 label categories is low, at 0.45 with text compression and 0.48 without, precision values for several label categories are comparable to past methods' (median: 0.47, max: 0.66).

1. Introduction

Present social media platforms and video-sharing sites contain copious amounts of tutorial-type videos of a person (or a group) performing a skilled or semi-skilled task in front of a stationary or moving camera. Audiences seek visual demonstration of a task or a skill to emulate. We can see ample evidence of this trend gaining popularity in recent years. These tutorial videos are popular mainly due to their accessibility and perceived usefulness. From a survey [1] involving 141 university students, all had consumed YouTube tutorial videos before, with 57.4% claiming easy access (to the videos) and saving time spent on figuring out how to perform a task as their main motivation. In 2020, Statista Research published a report on TikTok views categorized by hashtags [2]. The second highest number of hashtag views was dance-related content, at 181b views. Also, 5 out of 10 most popular hashtags were tutorial-type content, covering wide-ranging topics, i.e., fitness/sports (57b views), home renovation/DIY (39b views), beauty/skincare (33b views), and recipes/cooking (18b views). In these videos, individuals of varying skill levels performed tasks such as cooking, assembly, and repairs, as well as performative arts,

for example, martial arts and dancing. The ability to gauge the subject's skill level would be valuable for applications that require indexing and retrieval of a video database according to the subject's task expertise. For example, a video-sharing site could place videos performed by higher-skilled subjects at the top of the list when returning a search result. The current workaround is to crowdsource the ranking data via the upvote and downvote buttons. For this solution to work, it is assumed that the videos performed by higher-skilled subjects were upvoted by the majority, and downvoted if otherwise. Nevertheless, this method is hugely unreliable as the quality of annotations is influenced by the annotator's reliability and the task's difficulty level [3]. Worse yet, some participants may be adversarial by deliberately providing false labels [4].

Automated methods for producing ranking data from videos are classified under Action Quality Assessment (AQA) domain. Existing AQA methods either use regression for estimating the action quality score from a single video, or train models to predict the relative ranking between an input pair of videos. Our proposed method uses the subject's 2D poses throughout a dance performance to codify a String value. We then predict the video sourced from the better dancer by comparing edit distance scores of the following String value pairs. i.e., Subject A vs. Gold Standard and Subject B vs. Gold Standard. We define Gold Standard as the top-ranked video(s) for each label category. We estimate the subject's 2D pose from a single query frame. We will only process query frames containing complete pose data, i.e., all the required keypoints are detected, including false positives. Query frames containing incomplete pose data are rejected outright. The 2D pose is codified based on the position of four joints with respect to two auxiliary axes. A detailed explanation of the pose coding module can be found in Section 3.2. The matching Visual word is then appended into a String value that represents the current video. This process is repeated for the rest of the query frames, ultimately producing a String value of n length.

The two principal contributions of this paper revolve around automatically assessing the quality of a subject's actions, as captured by a stationary camera. The first contribution is the pairwise AQA method for determining the better dancer, given a pair of same-label videos. As for the second contribution, we released a new and open-access dataset for action quality assessment in the hopes of facilitating future research. The rest of this paper is organized as follows. Section 2 describes the related work. In Section 3, we present the novel dataset along with the pairwise AQA method, implemented under two different schemes. In Section 4, we discussed the results obtained from the two schemes. Finally, we conclude this paper in Section 5.

2. Method

Our proposed method has three modules, i.e., Region coding, Pose coding, and String builder module, see Fig. 1.

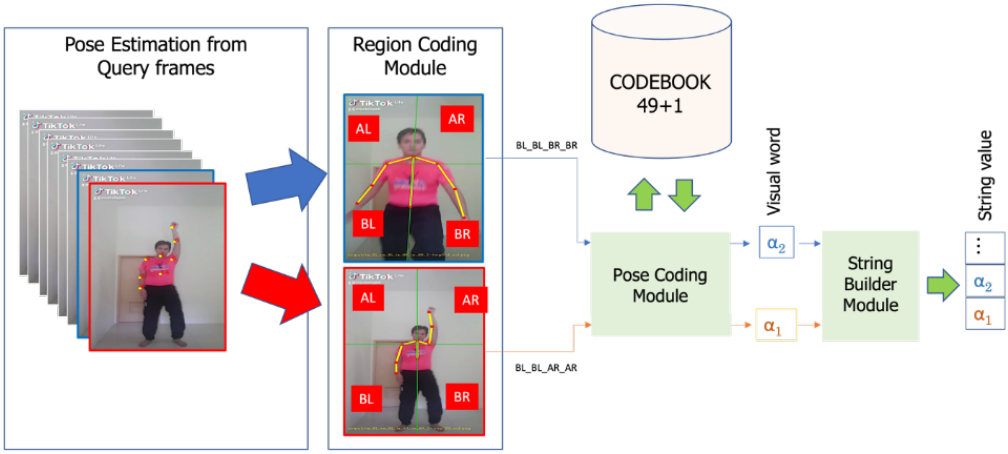

Fig. 1. A visualized explanation of our method.

The Region coding module encodes the detected 2D pose and feeds the encoded value into the Pose coding module. The Pose coding module acts as a look-up function by searching for the closest match inside the Codebook. The returned Visual word is then fed into the String builder module. This process is repeated for every query frame that contains complete pose data. Ultimately, the String builder module produces a String value of n length for a video with n complete query frames. The produced String value is later used during the pairwise skill annotation task between two videos of the same dance label. The rest of the section describes the proposed method by its components.

2.1. Video Classification

Action Quality Assessment (AQA) is a subset of the video classification problem. Therefore, it is beneficial to review existing work on video classification. Earlier methods use appearance and/or motion-based features extracted from consecutive frames. These features are built into a Spatio-temporal volume or a bag of features. One of the earliest appearance-based methods is Dollar et al. [5]. Their method samples local interest points over time, producing a cuboid of intensity, gradient, and motion information. The cuboid is further divided into stacked regions, described as a histogram each. As for motion-based methods, Wang et al. [6] capture local motion(s) by extracting dense trajectories described using motion boundary histograms. Instead of utilizing the entire frame, Hipiny et al. [7] generate a Visual word from a gaze-directed image region. The visual words from a frame sequence are tabulated into a histogram or appended into a String value for matching purposes.

Recent comparison studies, [8], [9] have shown that deep neural networks are superior to handcrafted approaches in the image classification task. As such, it is natural to extend their usage to video classification. Recently, researchers are looking at incorporating temporal data into the deep classification framework to improve accuracy. The success of approaches [10], [11] that utilized two-stream fusion architecture [12] suggested the value of doing so. Karpathy et al. [10] introduced three strategies, i.e., early, late, and slow fusion, to fuse information from a spatial network and a temporal network, with both running in parallel. Feichtenhofer et al. [11] proposed a two-pathway model, i.e., the SlowFast network. The first pathway captures the semantic information from frames captured at a lower rate, whilst the second operates at a higher rate. The difference in temporal resolution ensures that both slow and fast-changing motions are adequately captured. Gong et al. [13] recently developed Auto-TSNet, a two-stream model optimized for a giant multivariate search space. The model utilizes a progressive procedure that performs a search over individual streams, fusion, and attention blocks.

2.2. Video-based Dance Genre Classification

At the early stage, the same methods introduced for video classification were used for video-based dance genre classification. Nevertheless, dance is a highly dynamic and specialized class of human action. Therefore, an extended sequence of frames containing temporal information is often required for accurate classification. Previous work had incorporated temporal information in several ways. Castro et al. [14] used temporal 3D CNNs utilizing raster images, optical flow, and visualized multi-person 2D pose. The three separate stacks run in parallel and are fused at the end to produce the predicted class label. The videos were processed in a 16-frame chunk as a computational cost-saving measure. In Tsuchida et al. [15], the dancer's body motions are learned per frame as a 126-dimensional feature vector containing pose, velocity, and acceleration information. The feature vectors are aggregated within a temporal interval set by beat positions or fixed by a uniform window length. The classifiers were trained using LSTM and SVM models, achieving a 91.4% accuracy on a custom dataset. In Wysoczańska and Trzciński [16], the dance videos were augmented with audio track information. They extended the work [14] by adding a fourth stream, i.e., an audio-specific stream employing Liu et al.'s Bottom-up Broadcast Neural Network [17]. The final class prediction is averaged from the softmax output of each stack. More recently, Hu and Ahuja [18] proposed an LSTM-based hierarchical framework for classifying dance genres from a video. They trained an LSTM model to recognize 3D movements associated with specific body parts. The movements are identified from 3D poses extrapolated from the estimated 2D poses. The 2D to 3D mapping was learned by training temporal convolutional networks on videos with manually segmented movements.

2.3. Video-based AQA Methods

Existing work on automated action quality assessment of videos uses local features or deep neural networks. An example of the former is Pirshiavash et al. [19]. They trained a regression model to map Spatiotemporal features to an action quality score over a training set with manual scoring data. The feature set consists of low-level image features and a high-level feature, i.e., discrete cosine transform (DCT) encoded body pose, extracted from a single video. Similar to Pirshiavash et al.'s, the following deep learning methods also work on a single video and generate scores via regression. Instead of using manual scoring data as labels during supervised training, Parmar and Morris [20] proposed a multitask learning approach. 3D CNNs were used to learn Spatiotemporal motions and appearance-based features inside a video. The 3D CNNs were jointly optimized end-to-end to enable fine-grained action description and AQA scoring. In Xu et al. [21], two complementary networks, i.e., Self-Attentive LSTM (S-LSTM) and Multi-Scale Convolutional Skip LSTM (M-LSTM) were used to predict the AQA score. S-LSTM selectively learns the spatiotemporal features based on the frame's weight. Frames containing complicated and technical movements are weighted heavier than the rest. The second network, i.e., M-LSTM, learns to model the action at multiple scales using varying kernel sizes. This dual-network setup allows the action's local and global information to be adequately captured. Pan et al. [22] predict the AQA score based on the interactive motion pattern of neighboring joints. The patterns are modeled using spatial and temporal graphs. Regression-based methods work on the assumption that manual scoring by human judges is consistent and bias-free. To address this inherent ambiguity, Tang et al. [23] developed an uncertainty-aware score distribution learning where the score is inferred from Gaussian-distributed scores. Yu et al. [24] trained a binary tree classifier on feature vectors learned from query and exemplar pairs to predict the score difference. Each feature vector consists of Spatio-temporal features Markov Model. Based on the relative ranking between input pairs, the algorithm learns a model extracted from both videos and the reference score. This coarse-to-fine approach enables more accurate skill scoring since the final score is averaged from scores of multiple exemplar videos with similar attributes (i.e., stored in neighboring leaves belonging to the same and non-overlapping group). Instead of regressing a score from a video, Doughty et al. [25] trained temporal attention modules to focus only on frames containing skill-relevant parts. Their method uses a novel rank-aware loss function to perform pairwise ranking of egocentric videos. John et al. [26] measure the percentage of overlapping area between aligned foreground images from frames belonging to an unlabeled video and a video with an annotated score. The resulting value indicates how similar the aerobic dance moves are between the pair of videos.

2.4. Dance-related Datasets

Castro et al.'s Let's Dance dataset [14] contains 1,000 videos belonging to 10 different dance categories. These dance categories share overlapping micro-actions, making it impossible to make a class prediction from a single frame. The dances were performed by the same person/duo/group at the same venue; hence the videos share many appearance-based features. A much larger dataset, i.e., AIST Dance Video Database, was introduced by Tsuchida et al. [15]. The dataset focuses on street/urban dancing styles, containing almost 14k videos in 10 different genres. Professional dancers performed the dances, and the recording was done inside a well-lit studio using high-definition cameras. A more recent dataset, i.e., the UID dataset, was introduced by Hu and Ahuja [18]. The dataset contains 1,143 videos belonging to 9 classic dance genres. Most videos were curated from YouTube hence exhibiting greater variations in quality. Dance datasets in other modalities are also available, e.g., motion-capture in Dewan et al. [27] and Li et al. [28]; as well as depth data in [29]–[31].

Like Hu and Ahuja [18], our videos were captured under unregulated background and illumination settings. Unlike the formal dances covered in Castro et al.'s dataset [14], the TikTok dance challenges involve much slower and less intricate dance movements. Nevertheless, the movements still require some skill to execute correctly.

2.5. Dataset and Ground Truth Preparation

We first describe our novel dataset and the ground truth preparation steps. To prepare the dataset, we asked 20 participants (P1–P20) to perform 12 TikTok dance challenges each, resulting in a total of 240 videos, see Fig. 2. The 8–23 seconds videos were captured with various background scenes and lighting conditions. Participants used their own camera devices to capture these videos; hence different levels of image quality are expected. We used an interval of one-fifth of a second for extracting the still frames. We found the interval rate sufficient for capturing our participants' dance motions.

Fig. 2. Random screengrabs from our dataset, each belonging to a participant (P1–P20), from left-to-right, top to bottom.

The 12 TikTok dance challenges are *Diam-diam Menyukaimu* (D-dM), *Laxed (Siren Beat)* (L(SB)), *Lottery – K CAMP* (L-KC), *Blinding Lights* (BL), *Say So* (SS), *Supalonely* (S), *Slide to The Left* (StTL), *The Dance Song* (TDS), *Tokyo* (T), *Big Up's* (BU), *Tak Mau Mau* (TMM), and *JKVE – Upside Down* (JKVE-UD). Each label category has $C_2(20)$ equals 190 possible pairings. Thus, the total pairings across 12 categories equal 2,280 pairs. The pairs were divided randomly and equally between 100 paid human annotators. The annotators were tasked to mark the winning video for each pair. Before the annotator can annotate a winner, he or she must watch the included reference video in full. The short reference videos were sourced from TikTok and selected based on the highest number of likes. Annotators were reminded to select the winner based exclusively on the dance motions' quality whilst ignoring the dancer's appearance and video production quality.

After every pair were annotated, we traversed the entire list and updated each video's accumulated win count, ω,

$$pair(V_m, V_n) = \begin{cases} \omega_{V_m} = \omega_{V_m} + 1, & if\ V_m\ wins \\ \omega_{V_n} = \omega_{V_n} + 1, & otherwise \end{cases} \quad (1)$$

Ultimately, we produced a ranked list per label category as our ground truths. For each label category, the videos are ranked based on their win count. Videos with identical win counts are placed in the same ranking position. The top-ranked video(s) are thus declared as the Gold Standard(s). The complete dataset and the ranking data (for each label category) can be downloaded from kaggle.com/datasets/irwandihipiny/cdrg-unimas-tiktok.

2.6. Building the Codebook

Since a region label has four possible values, four region labels produce 4! possible combinations. However, we only consider the 49 most common poses, as observed in our dataset. We added the 50th class, i.e., OTHERS, to represent the rest of the (low frequency) codes. Fig. 3 shows all 49+1 possible region codes inside our Codebook. The 49+1 codes are each represented by an ASCII character (65 – 114), i.e., a Visual word, inside our Codebook.

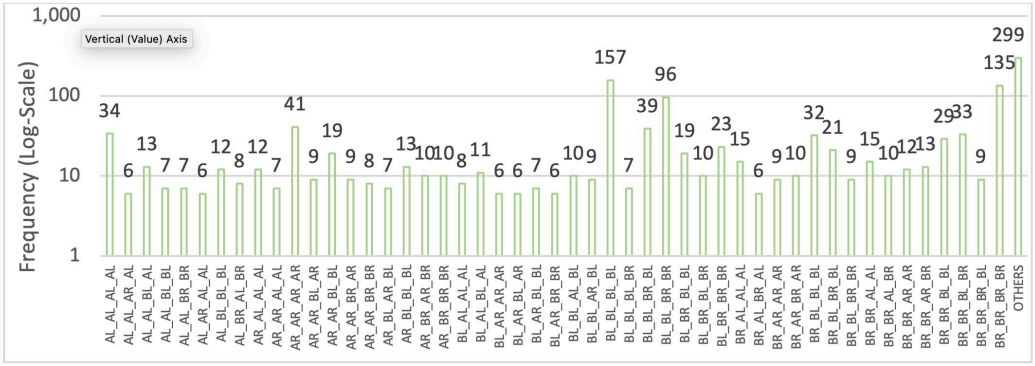

Fig. 3. The frequency (log-scale) of each region code. Each region code is represented as a Visual word (i.e., ASCII character) inside our Codebook.

2.7. Region Coding Module

We implemented Papandreou et al.'s PoseNet [32] in our Region coding module to estimate the subject's 2D pose inside a query frame. PoseNet estimates 2D poses in real-time by returning an array containing fifteen 2D coordinates with a confidence score each. Our method requires only 8 keypoints, regardless of whether the detection is a true or a false positive. See Fig. 4 for query frame samples containing complete pose data.

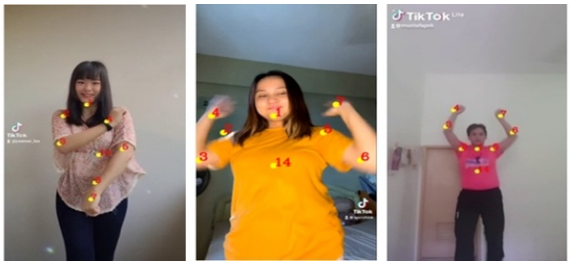

Fig. 4. Sample query frames, each contains complete pose data.

We require the following four key points for pose definition, i.e., *rightElbow*, *rightWrist*, *leftElbow*, and *leftWrist*, and the following four keypoints for region coding, i.e., *rightShoulder*, *leftShoulder*, *neck*, and *chest*. We define a query frame with complete pose data as having all eight keypoints being detected. Query frames without complete pose data are rejected outright. We purposely set a slightly weaker threshold value than the default used in Papandreou et al. [32] to increase the number of query frames per video. We found the default threshold value to be too strict thus producing a low number of query frames. The lower threshold risks a false keypoint detection, causing an erroneous Visual word to be appended to the String value. Nevertheless, we believe this event to be a rare occurrence due to the reliability of Papandreou et al.'s [32] method. Based on our observation, the matched keypoints are often correct, only to be rejected due to a slightly lower confidence score than the set threshold.

The Region coding module encodes a code value given a query frame with complete pose data. The code consists of 4 region labels. Possible values are *Above-Left* (AL), *Above-Right* (AR), *Below-Left* (BL), and *Below-Right* (BR). The value is determined by the position of the current keypoint relative to the two intersecting lines formed by the last four keypoints, i.e., *rightShoulder*, *leftShoulder*, *neck*, and *chest*. The first line/axis connects the *leftShoulder* and *rightShoulder* keypoint, and the second line/axis connects the *neck* and *chest* keypoint.

2.8. Pose Coding Module

The encoded value then becomes an input for the Pose coding module. This module acts as a look-up function, i.e., searching for the top match (Visual word) inside the Codebook for the given encoded value. The same module is used during the Codebook creation to generate query frames (with pose labels) from training videos.

2.9. String Builder Module

The String Builder module generates a String value of n length where n is equal to the number of complete query frames. It appends the matching Visual word, α_j, into the current video's String value, β_{Vi}. This step is repeated for all complete query frames, Φ_{Vi},

$$StringBuilder(\beta_{Vi}, \alpha_j) \; \forall \, j \in \Phi_{Vi} \tag{2}$$

We implemented a secondary scheme with a text compression feature. In this particular scheme, we only append α_j into β_{Vi} if $\alpha_j \neq \alpha_{j-1}$. This scheme compensates for instances of the subject executing a specific dance motion part faster or slower than the ideal duration. Only the appearance order of 2D poses is retained, while the magnitude (i.e., the number of frames per pose) is capped at a fixed value of 1. A combined flowchart explaining the process flow between the three modules is shown in Fig. 5.

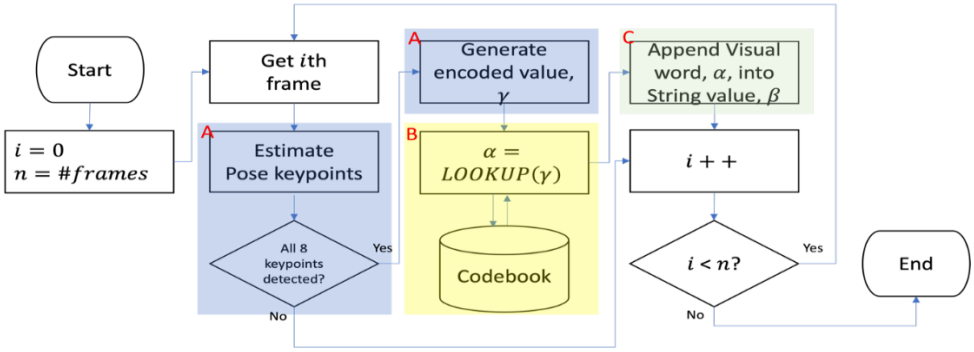

Fig. 5. Process flow between A) Region coding, B) Pose coding, and C) String builder module.

2.10. Pairwise Skill Annotation

Pairwise skill annotation on a same-label pair, $pair(V_m, V_n)$, is performed by first computing the following two edit distance scores. The first score, lev_1, is between V_m's String value and the Gold Standard's. The second score, lev_2, is between V_n's String value and the Gold Standard's. We determine the winning video using,

$$Pairwise(V_m, V_n) = \begin{cases} V_m, & lev_1 < lev_2 \\ V_n, & lev_1 > lev_2 \\ \infty, & lev_1 = lev_2 \end{cases} \tag{3}$$

where ∞ indicates a stalemate. In the event of a label category containing two or more Gold Standards, we compute lev for all cases and select the lowest edit distance score as the representative value for the current video.

3. Results and Discussion

Table 1 shows the precision values obtained using the two schemes for each label category, with the last column reporting the average value. We report results with the default Codebook size of 49+1 Visual words.

Table 1. Precision value by Label Category.

	D-dM	L(SB)	L-KC	BL	SS	S	StTL	TDS	T	BU	TMM	JKVE-UD	Avg.
Primary	0.59	0.31	0.43	0.47	0.53	0.59	0.47	0.53	0.58	0.35	0.31	0.57	0.48
Secondary (w/ Text Comp-ression	0.66	0.37	0.43	0.46	0.33	0.55	0.56	0.38	0.57	0.34	0.32	0.46	0.45

The primary scheme obtains an average precision of 0.48, while the second scheme with text compression manages a slightly lower value of 0.45 over 12 dance labels. Encoding the temporal duration of choreographed dance motions is beneficial since it influences our perception of motion quality (i.e., too slow or too fast). Nevertheless, the secondary scheme outperforms the primary scheme in 4 out of 12 label categories, i.e., D-dm, L(SB), StTL, and TMM. We argue that the best scheme depends on the nature of the choreographed dance motions. Some TikTok dance challenges involve rapidly changing motions; hence temporal duration might not be a deterministic factor in action quality assessment. The best precision value (0.66) is achieved using the secondary scheme on the D-dM set. We note that the difference in precision values achieved using this set's primary and secondary scheme is only 0.07. The most significant difference is in the TDS set, with the primary scheme outperforming the secondary scheme by 0.15.

Our results are not directly comparable to previous work since the dataset(s) and metrics used were different. Admittedly, the average precision over 12 label categories is rather low, at 0.45 with text compression, and 0.48 without. Nevertheless, precision values for several label categories are comparable to past methods' (median: 0.47, max: 0.66). Our method's performance is comparable to the state-of-the-art, for example, Jain et al. [33] reported an average precision of 0.58 for the task of returning video clips containing poorly-executed actions from a same-label set.

Codebook Sizes We repeat the same experiment using the primary scheme with smaller Codebook sizes, see Fig. 6. We reduce the Codebook size to k=38, k =29, k =19, and k =10, by removing Visual words with m lowest frequencies. For example, at k =10, Visual words with a frequency value of 23 or less were removed, i.e., merged into OTHERS. We observe the primary scheme to be resilient against the reduction in Codebook's size. The removal of certain Visual words from the Codebook affects the label categories differently since they may or may not be present inside the videos' original String value (at k =50). At k =10, the frequency values of the remaining Visual words are between 29–157 (excluding OTHERS). The high-frequency values are expected since the 12 dance challenges tend to share visually similar and repetitive dance motions.

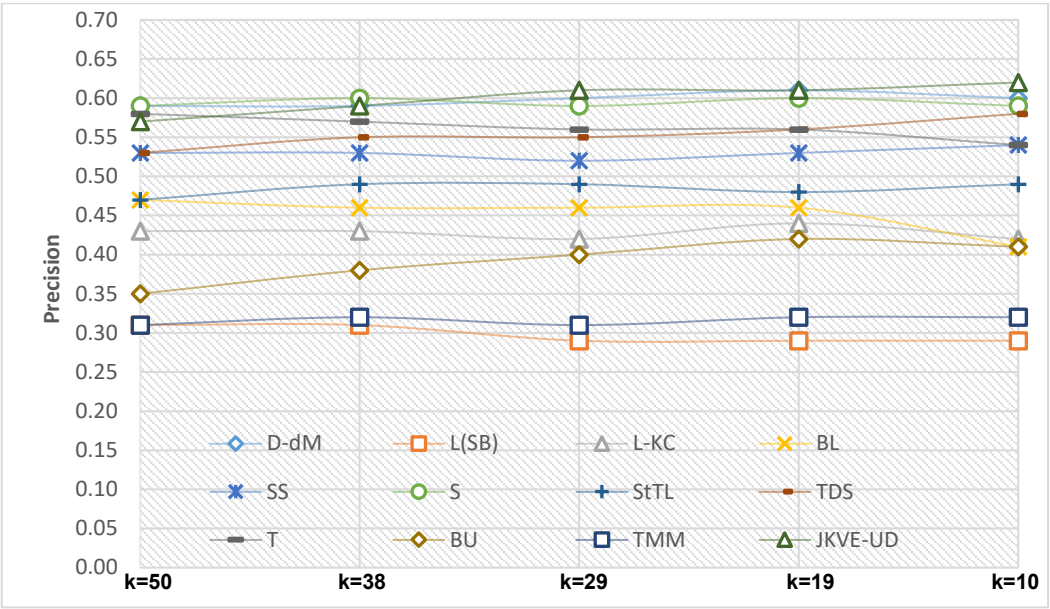

Fig. 6. Precision value for each label category using the primary scheme with different Codebook sizes.

Separation in Ranking We are interested in the method's performance when selecting the winner from pairs with high separation vs. close pairs inside the ranked list. Fig. 7 compares the results obtained using the two schemes. The primary scheme shows an increase in precision value if we limit to high separation pairs. As for the second scheme's precision value is consistent across separation values. The primary scheme works best if the two subjects have a markedly clear difference in skill level. At the max separation value of 8, the primary scheme achieves a high precision value of 0.69.

Fig. 7. Precision value vs. separation. We limit the maximum separation to 8 since some of the ranked lists contain only up to 9 rank positions.

Failure Cases A simple check for 8 mandatory pose key points help eliminates empty and faulty frames (i.e., blurry or containing vague poses). However, false positives can occur during the keypoint detection stage; causing the Region coding module to encode an incorrect value, see Fig. 8. Nevertheless, these failure cases are rare due to the accuracy of Papandreou et al.'s [32] pose estimation method.

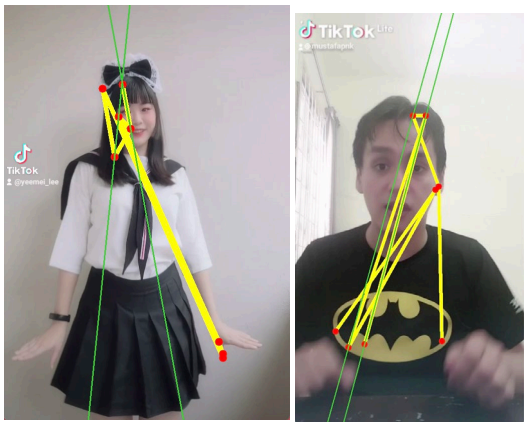

Fig. 8. Failure cases. In the above two examples, even though all 8 mandatory keypoints were positively detected, at least one turned out to be a false positive. This causes an incorrect formation of the two intersecting (green) lines and/or erroneous position labeling of the joint(s).

3.1. Evaluation Metrics

To evaluate our method, we check the predicted winner, $Pairwise(V_m, V_n)$, of each same-category pair against the category-specific ranked list. We define the following two scenarios as a $True\ Positive$ event, i.e., i) the predicted winner is indeed ranked higher, and ii) a stalemate is predicted for a pair of videos with identical win counts. We report the precision values in the following subsection.

4. Conclusion

We presented a pairwise AQA method with two schemes to determine a winner from a pair of same-label TikTok dance videos. In both schemes, the subject's 2D poses are codified into a String value based on the position of selected joints relative to two auxiliary axes. Using both schemes, we compute the edit distance score between the resulting String value and the Gold Standard's. The secondary scheme implements an additional step with text compression before calculating the edit distance score. We have tested these two schemes on a newly created dataset and achieved an average precision of 0.48 and 0.45, respectively. We also experimented with different Codebook sizes and compared precision values for pairs with high and low separation regarding ranking position. We also show the current limitation of our method, i.e., dependency on the 2D pose estimation result. An incorrect pose estimation produces an erroneous String value that affects the precision of the pairwise action quality assessment task. We see our work as a step toward an automated ranking of general and task-independent videos based on the subject's skill level. Measuring the similarity of task-related motions to an expert's is a valid approach for estimating task mastery. The novel TikTok dance video dataset is made publicly available on the authors' website to motivate more work in skill determination from video. Further work involves exploring higher-dimensional ways to represent the subject's 2D/3D poses and testing on additional AQA datasets.

Acknowledgment

We wish to acknowledge Universiti Malaysia Sarawak for funding the publication of this research work.

Declarations

Author contribution. Hipiny, I., was responsible for the design and overall investigation. Hipiny, I. and Ujir, H. co-designed the method. Shanat, M., Alias, A.A. and Hipiny, I. designed and conducted the data collection activities. Ishak, M.K. was responsible for statistical data analysis. All authors had read and approved the final manuscript.

Funding statement. Universiti Malaysia Sarawak's internal research grant, i.e., UNIMAS CDRG (F08/CDRG/1820/2019).

Conflict of interest. The authors declare no conflict of interest.

Additional information. No additional information is available for this paper.

Data and Software Availability Statements

The dataset is available at the following URL: https://www.kaggle.com/datasets/irwandihipiny/cdrg-unimas-tiktok under the following license: CC BY-NC 4.0.

References

[1] E. Roque Rodríguez, "Youtube tutorials as a non-formal learning strategy for university students," *RIDE. Rev. Iberoam. para la Investig. y el Desarro. Educ.*, vol. 11, no. 21, p. 153, Dec. 2020, doi: 10.23913/RIDE.V11I21.797.

[2] L. Ceci, "Top categories on TikTok by hashtag views 2020 | Statista," *Statista*. Available at : Statista.

[3] J. Lin, T. Yu, and Z. J. Wang, "Rethinking Crowdsourcing Annotation: Partial Annotation with Salient Labels for Multilabel Aerial Image Classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, doi: 10.1109/TGRS.2022.3191735.

[4] G. Dawson and R. Polikar, "OpinionRank: Extracting Ground Truth Labels from Unreliable Expert Opinions with Graph-Based Spectral Ranking," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2021-July, Jul. 2021, doi: 10.1109/IJCNN52387.2021.9533320.

[5] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," *Proc. - 2nd Jt. IEEE Int. Work. Vis. Surveill. Perform. Eval. Track. Surveillance, VS-PETS*, vol. 2005, pp. 65–72, 2005, doi: 10.1109/VSPETS.2005.1570899.

[6] H. Wang, A. Kläser, C. Schmid, and C. L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, May 2013, doi: 10.1007/S11263-012-0594-8.

[7] I. Hipiny and H. Ujir, "Measuring task performance using gaze regions," *2015 9th Int. Conf. IT Asia Transform. Big Data into Knowledge, CITA 2015 - Proc.*, Dec. 2015, doi: 10.1109/CITA.2015.7349836.

[8] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev. 2020 538*, vol. 53, no. 8, pp. 5455–5516, Apr. 2020, doi: 10.1007/S10462-020-09825-6.

[9] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.

[10] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and F. F. Li, "Large-scale video classification with convolutional neural networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1725–1732, Sep. 2014, doi: 10.1109/CVPR.2014.223.

[11] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-October, pp. 6201–6210, Oct. 2019, doi: 10.1109/ICCV.2019.00630.

[12] K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," *Adv. Neural Inf. Process. Syst.*, vol. 1, no. January, pp. 568–576, Jun. 2014, doi:10.48550/arXiv.1406.2199.

[13] X. Gong, H. Wang, Z. Shou, M. Feiszli, Z. Wang, and Z. Yan, "Searching for Two-Stream Models in Multivariate Space for Video Recognition," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 8013–8022, 2021, doi: 10.1109/ICCV48922.2021.00793.

[14] D. Castro *et al.*, "Let's Dance: Learning From Online Dance Videos," Jan. 2018, doi: 10.48550/arXiv.1406.2199.

[15] S. Tsuchida, S. Fukayama, M. Hamasaki, and M. Goto, "AIST Dance Video Database: Multi-Genre, Multi-Dancer, and Multi-Camera Database for Dance Information Processing," Nov. 2019, doi: 10.5281/zenodo.3527853.

[16] M. Wysoczańska and T. Trzciński, "Multimodal dance recognition," *VISIGRAPP 2020 - Proc. 15th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl.*, vol. 5, pp. 558–565, 2020, doi: 10.5220/0009326005580565.

[17] C. Liu, L. Feng, G. Liu, H. Wang, and S. Liu, "Bottom-up broadcast neural network for music genre classification," *Multimed. Tools Appl.*, vol. 80, no. 5, pp. 7313–7331, Feb. 2021, doi: 10.1007/S11042-020-09643-6.

[18] X. Hu and N. Ahuja, "Unsupervised 3D Pose Estimation for Hierarchical Dance Video Recognition," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 10995–11004, 2021, doi: 10.1109/ICCV48922.2021.01083.

[19] H. Pirsiavash, C. Vondrick, and A. Torralba, "Assessing the quality of actions," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8694 LNCS, no. PART 6, pp. 556–571, 2014, doi: 10.1007/978-3-319-10599-4_36.

[20] P. Parmar and B. T. Morris, "What and how well you performed? a multitask learning approach to action quality assessment," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 304–313, Jun. 2019, doi: 10.1109/CVPR.2019.00039.

[21] C. Xu, Y. Fu, B. Zhang, Z. Chen, Y. G. Jiang, and X. Xue, "Learning to Score Figure Skating Sport Videos," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4578–4590, Dec. 2020, doi: 10.1109/TCSVT.2019.2927118.

[22] J. H. Pan, J. Gao, and W. S. Zheng, "Action assessment by joint relation graphs," *Proc. IEEE Int. Conf.*

Comput. Vis., vol. 2019-October, pp. 6330–6339, Oct. 2019, doi: 10.1109/ICCV.2019.00643.

[23] Y. Tang *et al.*, "Uncertainty-Aware Score Distribution Learning for Action Quality Assessment," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 9836–9845, 2020, doi: 10.1109/CVPR42600.2020.00986.

[24] X. Yu, Y. Rao, W. Zhao, J. Lu, and J. Zhou, "Group-aware Contrastive Regression for Action Quality Assessment," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 7899–7908, 2021, doi: 10.1109/ICCV48922.2021.00782.

[25] H. Doughty, W. Mayol-Cuevas, and Di. Damen, "The pros and cons: Rank-aware temporal attention for skill determination in long videos," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2019-June, pp. 7854–7863, Jun. 2019, doi: 10.1109/CVPR.2019.00805.

[26] F. John, I. Hipiny, and H. Ujir, "Assessing performance of aerobic routines using background subtraction and intersected image region," *Proc. - 2019 Int. Conf. Comput. Drone Appl. IConDA 2019*, pp. 38–41, Dec. 2019, doi: 10.1109/ICONDA47345.2019.9034912.

[27] S. Dewan, S. Agarwal, and N. Singh, "A deep learning pipeline for Indian dance style classification,", vol. 10696, pp. 265–273, Apr. 2018, doi: 10.1117/12.2309445.

[28] B. Li, Y. Zhao, Z. Shi, and L. Sheng, "DanceFormer: Music Conditioned 3D Dance Generation with Parametric Motion Transformer," *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 2, pp. 1272–1279, Jun. 2022, doi: 10.1609/AAAI.V36I2.20014.

[29] H. Matsuyama, K. Hiroi, K. Kaji, T. Yonezawa, and N. Kawaguchi, "Hybrid activity recognition for ballroom dance exercise using video and wearable sensor," *2019 Jt. 8th Int. Conf. Informatics, Electron. Vision, ICIEV 2019 3rd Int. Conf. Imaging, Vis. Pattern Recognition, icIVPR 2019 with Int. Conf. Act. Behav. Comput. ABC 2019*, pp. 112–117, May 2019, doi: 10.1109/ICIEV.2019.8858524.

[30] H. Bhuyan, J. Killi, J. K. Dash, P. P. Das, and S. Paul, "Motion Recognition in Bharatanatyam Dance," *IEEE Access*, vol. 10, pp. 67128–67139, 2022, doi: 10.1109/ACCESS.2022.3184735.

[31] M. Ma, S. Sun, and Y. Gao, "Data-Driven Computer Choreography Based on Kinect and 3D Technology," *Sci. Program.*, vol. 2022, 2022, doi: 10.1155/2022/2352024.

[32] G. Papandreou, T. Zhu, L. C. Chen, S. Gidaris, J. Tompson, and K. Murphy, "Personlab: Person pose estimation and instance segmentation with a bottom-up, part-based, geometric embedding model," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11218 LNCS, pp. 282–299, 2018, doi: 10.1007/978-3-030-01264-9_17.

[33] H. Jain, G. Harit, and A. Sharma, "Action Quality Assessment Using Siamese Network-Based Deep Metric Learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 6, pp. 2260–2273, Jun. 2021, doi:10.1109/TCSVT.2020.3017727.