Application of the AIST LSTM-based Genre Classification Model to the BRACE dataset

A Special Problem
Presented to the Faculty of the Division of Physical Sciences and Mathematics
College of Arts and Sciences
University of the Philippines Visayas
Miag-ao, Iloilo

By
HUPP, James Ezra E.

Ara Abigail Ambita
Adviser

Contents

Chapter 1

1 Introduction

1.1 Background and Rationale

Dance embodies a unique category of human activity. With its global prevalence and cultural importance, dance holds a pivotal role in both communal and personal aspects, serving as both a unifying element and an outlet for self-expression. It is composed of several different essences, movements, and poses, all of which fit into a certain dance style, also known as a dance genre.

In the preceding years, machine learning has emerged as a tool to identify and analyze human activity. Machine learning, which is a branch of Artificial Intelligence (AI), places emphasis on the use of data and algorithms to replicate human learning (IBM, 2019). Furthermore, deep learning is a subset of machine learning which utilizes artificial neural networks consisting of multiple layers (Lgayhardt 2023). With dance being a topic of recent interest, a few datasets have emerged that feature different dance genres, such as AIST (Tsuchida et al., 2019), Everybody Dance Now (Chan et al., 2019), and BRACE (Moltisanti et al., 2022). Consequently, some studies have been conducted which have used varying machine learning and deep learning techniques to classify video and audio sequences into their respective dance genres.

For example, "AIST DANCE VIDEO DATABASE: MULTI-GENRE, MULTI-DANCER, AND MULTI-CAMERA DATABASE FOR DANCE INFORMATION PROCESSING" by Tsuchida et al. (2019) applies four baseline methods based on a Support Vector Machine (SVM) based model and a Long Short-Term Memory (LSTM) based model to classify videos from the AIST dataset into one of its 10 dance genres. They found that the LSTM model performed better with 91% accuracy. It is pertinent to mention that an SVM algorithm is a type of machine learning tool that is used to predict outcomes or categorize items by sorting them into groups (aswathisasidharan, 2023). Conversely, LSTM is a type of Recurrent Neural Network that detains long-term dependecies in sequential data (Banoula, 2023) (Sherstinsky, 2020). Moreover, "Dance Video Classification into Relevant Street Dancing Styles using Deep Learning Techniques" by Bauskar, D. (2022) uses VGG-16 and VGG-19 to perform the same task, also on the AIST dataset. They found that the VGG-16 based model performed best with 75.86% accuracy. Consisting of 10 dance genres with 13,939 videos Tsuchida et al. (2019) and recorded in a controlled setting, AIST provides an ideal dataset for genre classification tasks. However, this raises the question of how current dance genre classification models perform when tested against more complex, unstructured videos.

The BRACE dataset was created for the purpose of challenging current genre classification models, with more complex videos captured in a 'real world setting' Moltisanti et al. (2022). It contains breakdancing videos from the Red Bull BC One competition in Youtube. Breakdancing, relative to other dance genres, is highly unstructured with more dynamic movements and a broader range of poses and positions. For example, BRACE classifies breakdancing elements (or subgenres) into toprock, footwork, and power moves. Each of these subgenres is distinct from one another and are identified through its own unique set of movements and positions. Contrary to the AIST dataset, the videos in BRACE contain dancers performing in a breakdancing-only competetion with various camera angle changes, motion blur, and audiences in the background. Given the more unstructured nature of breakdancing and the noisy, more complex videos in BRACE, it is therefore hyptothetically more difficult for existing models to accurately classify subgenres within breakdancing. This presents an unprecedented opportunity to assess the accuracy of these models against BRACE.

This special problem aims to use the AIST genre classification model, specifically the L-fixed LSTM-based method, to classify breakdancing footage in BRACE into its subgenres of toprock, footwork, and power moves.

1.2 Statement of the Problem

There is a significant gap in understanding how genre classification models perform when applied to more complex and less controlled dance sequences. Existing studies have relied on datasets featuring controlled settings, such as the AIST dataset, which, while comprehensive, does not fully represent the real-world diversity and complexity of dance performances. Furthermore, dance genres vary in complexity and breakdancing, as an example, contains dynamic movements and more complex poses compared to other genres.

The BRACE dataset, which features the dynamic and highly unstructured genre of breakdancing, provides an opportunity to address this gap. The problem is to assess the adaptability and accuracy performance of the AIST genre recognition model when applied to the BRACE dataset.

This study aims to answer the following questions:

1. To what extent can the AIST genre recognition model accurately classify dance genres in the BRACE dataset?

2. How does the model's accuracy on the BRACE dataset compare to its performance on controlled datasets, such as AIST?

3. What challenges and limitations does the AIST model encounter when dealing with the unique characteristics of breakdancing as represented in the BRACE dataset?

Addressing these questions is vital to the development of more robust genre classification models that can handle the complexities of real-world dance performances.

1.3 Significance of the Study

This special problem is significant in the following areas:

1. Improving Dance Genre Classification: The findings of this study provides an opportunity to enhance the accuracy and effectiveness of dance genre classification models.

Insights gathered from testing the AIST model against the BRACE dataset may result in the creation of new models that are better suited to complex, noisy, video segments.

2. Interdisciplinary Collaboration: This study encourages collaboration between the fields of computer vision and dance. It bridges the gap between technology and the arts, promoting an interdisciplinary approach that improves creativity in both fields.

1.4 Objectives

This study aims to explore how existing dance genre classification models perform when tested on complex, noisy sequences by assessing the accuracy of the AIST genre classification model when applied to the BRACE dataset.

More specifically, this study aims to:

1. Use the L-fixed LSTM-based method of the AIST genre classification model to classify breakdancing sequences into toprock, footwork, and powermoves.
2. Compare the accuracy of the model when used between the AIST database and the BRACE dataset.

1.5 Scope and Limitations

This study encompasses the application of the current dance genre recognition models for the classification of dance genres within the context of unstructured, real-world dance sequences. It focuses on the AIST dataset, which represents controlled and structured dance sequences, and the BRACE dataset, which represents noisy and complex breakdancing dance sequences. Specifically, it uses the L-fixed LSTM-based method of the AIST genre recognition model as it achieved the highest accuracy for the AIST dataset.

The study is limited to classifying subgenres within the genre of breakdancing contained in BRACE. Furthermore, it is important to note that while the study attempts to test the

AIST model against a dataset more reflective of the ”real world”, the BRACE dataset is still a curated one. Thus, real-world dance sequences may exhibit greater variability.

The study will be conducted throughout the first and second semester of the 2023-2024 school year.

Chapter 2

2 Review of Related Literature and Works

2.1 Introduction

A dance genre can be identified by its set of movements and poses. While some recent studies have focused on the task of classifying dance genres based on this characteristic, few have taken into consideration the possible limitations of using controlled datasets to do so. This chapter delves into relevant literature and works, providing insight into the use of different machine learning techniques and the challenges posed by complex dance sequences. The review takes a methodological approach, examining the different methods the studies use to classify genres. By examining the existing methodologies and their results, this chapter aims to elucidate the existing body of knowledge and identify gaps in the field.

2.2 AIST Classifier

This study by Tsuchida et al. (2019) introduces the AIST database, which features 10 major dance genres, solo and group dance sequences by 40 professional dancers, and multiple camera angles. The major dance genres it features are: i.) Popping ii.) Locking iii.) Krump iv.) Waacking v.) Middle Hip-hop vi.) LA Style Hip-hop vii.) Breakdancing viii.) House ix.) Street Jazz x.) Ballet Jazz.

The study aimed to answer the following research questions:

1. “Can we classify the 10 genres by using their video frames only?”
2. “How many video frames should be used to train a model?”
3. “Is the ease of classification different by dance genre?”
4. “Can beat positions help improve classification accuracy?”.

To classify dance sequences into their respective genres, they use the OpenPose library for pose estimation. These movements are then converted into a 42-dimensional feature vector. They then calculate the velocity and acceleration of these movements between frames, resulting in a 126-dimensional vector for each frame. Two methods are then used to aggregate the 126-dimensional vectors into units of time, which are:

- Adaptive method: This method relies on the beat position of the music. Units can vary in length depending on the tempo of the music. One, two, three, or four beats are used as one unit, and the length of each unit corresponds to the tempo.

- L-fixed method: Units have fixed lengths, which can be 20, 40, 60, and so on up to 500 video frames.

Each method concatenates unit-level feature vectors from a certain number of units into window-level feature vectors. This process is then repeated with different window lengths. Finally, four baseline methods are prepared by combining either the Adaptive or L-fixed method with the use of LSTM-based or SVM-based models. Each method is designed to classify every window-level feature vector into one of the 10 dance genres.

- LSTM-Based Model: For this method, a bi-directional recurrent neural network (RNN) with one layer of LSTM cells is employed. This network outputs a 10-dimensional one-hot vector that represents the possible dance genres. An activation function called rectified linear unit (ReLU) is applied to the output of the LSTM. Batch normalization is also applied to the output layer of the dense layers to ensure stable training. The loss function used is cross-entropy, and a batch size of 10 is utilized for training. The model is trained with a learning rate of 5e-4 for 100 epochs, and the trained model is saved at the point of minimum validation loss. This particular model is implemented using the PyTorch framework.

- SVM-Based Model: In the SVM-based model, the process begins with obtaining 200-dimensional vectors through principal component analysis (PCA) to reduce the dimensionality of the training data. Subsequently, the SVM model is trained using these reduced-dimensional vectors. Finally, the dance genre of each window-level feature vector in a video is estimated by employing both the LSTM-based and SVM-based models.

Among the four baseline methods, the researchers found that the L-fixed method with the LSTM-based model provided the best results, with 91.4% accuracy. The L-fixed method with the SVM-based model dropped in accuracy with 84.0%. The study also found krump relatively easier to estimate and house more difficult. Moreover, genres with similiar poses, such as street jazz and ballet jazz, were easily confused by the classifier. Genres with similar movements also dropped in estimation accuracy. This is because house, for example, contains movements also found in other dance genres. Notably, there was no mention of how breakdancing affected the accuracy of their classifier, even though the AIST dataset contains breakdancing sequences.

2.3 Dance Video Classification into Relevant Street Dancing Styles using Deep Learning Techniques

This study by Bauskar (2022) also uses the AIST dataset. He uses three deep learning techniques (VGG-16, VGG-19, CNN) as classifiers. The study involved rigorous hyperparameter optimization to fine-tune the model, employing the ”relu” activation function and ”adam” optimizer. VGG-16 is a convolutional neural network 16 layers deep known for its simplicity. It scored the highest accuracy with 75.86%. VGG-19 is a deeper convolutional neural network with 19 layers. It achieved 68.96% accuracy, suggesting that its higher complexity did not necessarily lead to better performance. VGG-16, being computationally less complex, resulted in faster training and inference times. It should be noted the Bauskar does not specify the specific architectural details of the CNN method, as both VGG-16 and VGG-19 classify as CNN. Regardless, it scored the lowest accuracy at 27%. Compared to Tsuchida's LSTM-based classifier, Bauskar's score significantly lower accuracies. This may be because RNN, which LSTM a type of, is better suited for analyzing sequential data compared to CNN (Gupta, 2023), (Craig, 2023).

2.4 Who's Got the Groove?

In this study by Bendit-Shtull et al. (n.d.), different models were evaluated for action classification in video data, using the Let's Dance dataset by Castro et al. (2018). The baseline methods included a spatial frame-by-frame model, where a CNN was trained on individual video frames, and the final video classification was based on majority voting of its frames. It achieved a test accuracy of 56.%. Additionally, the study compared the proposed models to five temporal models presented by Castro et al. (2018), which incorporated the time dimension using optical flow data, 3D convolution, or stacked convolution, achieving varying levels of accuracy. The temporal 3D CNN using RGB data and the temporal three-stream CNN were among the most successful models, with accuracy scores of 70.11% and 71.%, respectively.

The proposed approach aimed to capture the temporal dimension without using optical flow data. It introduced three models using spatial data from each frame and sequential models to capture motion over time. The two-stream late fusion model utilized RGB images and PoseNet skeleton data. Features were extracted from RGB frames using a pretrained ResNet-18 model, while PoseNet data was processed with a shallow CNN. These features were concatenated to form 614-dimensional representations, and sequences were created through subsampling. Three sequential models were employed: an LSTM, an LSTM with self-attention, and a temporal convolutional neural network (TCN).

The LSTM model included a dropout layer, a LSTM layer with a hidden dimension of 100, and a fully-connected layer for classification. It was trained with hyperparameters such as learning rate, hidden dimension, batch size, dropout rate, weight decay, frame sampling frequency, and optimizer (Adam or SGD). The resulting model was trained with specific hyperparameter settings.

The LSTM with self-attention extended the basic LSTM by incorporating a self-attention mechanism to determine the importance of frames in the sequence. The attention mechanism used scaled dot product self-attention, and hyperparameters were tuned similarly to the LSTM model. The final model had distinct hyperparameter settings.

The temporal convolutional network (TCN) was introduced as an alternative to LSTM architectures. It featured causal convolutions with increasing dilation in each layer, allowing it to process sequences of varying lengths without suffering from vanishing gradients. The TCN was configured with multiple hyperparameters, including batch size, number of layers, convolution channels, dropout probability, learning rate, and optimizer. The final TCN model was trained with specific settings and had a large effective history and receptive field.

In summary, the study evaluated various baseline and proposed models for action classification in video data. The proposed approach aimed to capture the temporal dimension without relying on optical flow data, and it utilized three sequential models: LSTM, LSTM with self-attention, and TCN, each with specific hyperparameter settings. The results demonstrated the effectiveness of these models in capturing temporal information and achieving competitive accuracy levels when compared to the temporal models presented by Castro et al..

2.5 Synthesis

Three distinct studies employed machine learning techniques to classify dance genres, each using different methodologies and achieving varying levels of accuracy. Tsuchida et al. (2019) introduced the AIST database and utilized pose estimation with OpenPose, demonstrating that their L-fixed method coupled with an LSTM-based model yielded the highest accuracy at 91.4%. Bauskar (2022) employed VGG-16, VGG-19, and CNN classifiers on the AIST dataset, with VGG-16 having the most accuracy at 75.86%. Lastly, Bendit-Shtull et al. (n.d.) evaluated various models for action classification using the Let's Dance dataset, showing the effectiveness of proposed sequential models like LSTM with self-attention and TCN in capturing temporal information. Tsuchida et al.'s approach with the L-fixed method and LSTM-based model stood out as the most accurate method for dance genre classification.

Chapter 3

3 Methodology

This chapter discusses the materials and methods used in this special problem.

3.1 Development Tools

This special problem utilizes the following development tools:

1. Visual Studio Code, which is the study's Integrated Development Environment (IDE)

2. Github, which serves as the study's version control system and source code repository

3.2 Software Requirements

The project utilizes the following software and their respective versions:

1. Python version 3.11.4

2. Github

3. Git

The following Python packages and libraries were also used:

1. pandas

2. pathlib

3. Dataloader from torch.utils.data

4. torch.nn

5. tensor

6. Adam optimizer from torch.optim

3.3 Development Process

3.3.1 Development Process Diagram

Figure 1 shows the Development Process Diagram, which summarizes the development process of this study.

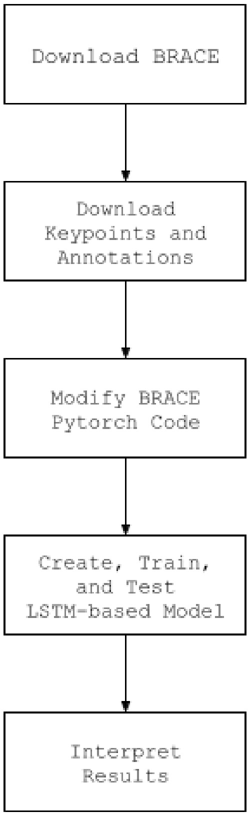
Download BRACE
Download Keypoints and Annotations
Modify BRACE Pytorch Code
Create, Train, and Test LSTM-based Model
Interpret Results

Figure 1: Development Process Diagram

3.4 Dataset

Downloading BRACE has to be done manually in order to avoid copyright infringement. This is because the videos used in the dataset come from the Red Bull BC One channel on YouTube, so Red Bull owns the copyright to all the videos. The authors do however provide a guide in their Github repository, using the command line program *youtube-dl* or *ytdl* to download videos from YouTube from a terminal. The format to download videos is:

youtube-dl -f 'bestvideo[ext=mp4]+bestaudio[ext=m4a]' [video ID]

3.5 Keypoints and Annotations

The BRACE Github repository Moltisanti et al. (2022) contains keypoints, audio features, and annotations of every sequence and segment in the BRACE dataset. Keypoints are sorted by year and by performance. They are further separated by genre where each file is a json file that splits the video into a genre performed by the dancer. Each file is a dictionary that contains an image file as the key which represents a single frame. Every frame has two values which are the bounding box and keypoints.

3.6 Modifying the BRACE Pytorch Code

To integrate the BRACE dataset into the PyTorch framework, Moltisanti's provided Python script from the same repository has been adapted. While originally designed to return sequences and metadata, the script has been modified to return sequence keypoints and their corresponding labels, specifically toprock, footwork, or powermove. The dataset can be instantiated by calling *BraceDataset()*

3.7 Creating, Training, and Testing the LSTM-based Model

Figure 2 shows the Diagram of the LSTM-based model.

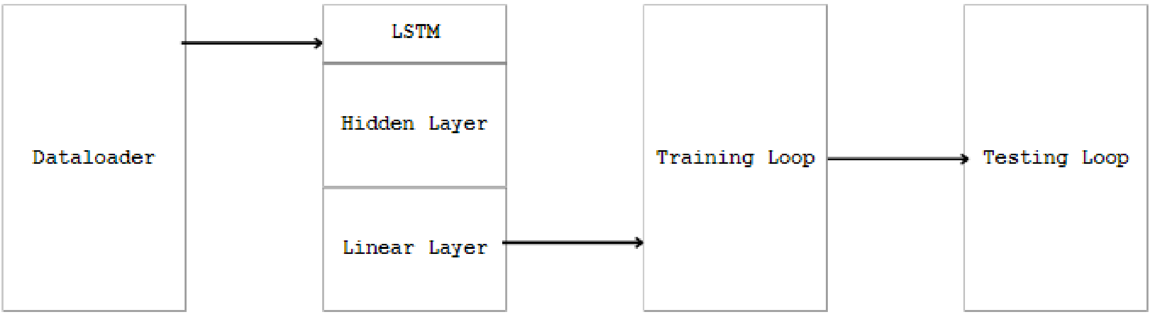

Figure 2: LSTM-based model diagram

The dataloader is divided into the training and testing splits, which were taken from the BraceDataset training and testing splits. PyTorch was the framework of use for this special probem, due to the PyTorch dataset already provided by Moltisanti et al. (2022).

3.7.1 Hyperparameters, Loss Function, and Optimizer

The LSTM-based model employed the following hyperparameters, which have been tested to yield the highest accuracy possible.

1. Number of layers: 3

2. Number of hidden layers: 2

3. Learning rate: 0.01

4. Batch size: 32

5. Number of epochs: 15

The loss function used for the model is Cross-Entropy Loss, which is ideal for classification tasks (Brownlee, 2020). Furthermore, Adam was used as the optimizer of choice.

3.8 Training and Testing

The training process involved iterating over batches of data using a DataLoader, with each batch consisting of input sequences and their corresponding labels. The model's parameters were optimized using the Adam optimizer, with a learning rate of 0.01, and the CrossEntropyLoss function served as the criterion for evaluating the model's performance. During each epoch, the model iteratively updated its weights based on the computed loss, aiming to minimize the difference between predicted and actual labels.

For testing, the evaluation involved running the model on batches of testing data and calculating the test loss, as well as assessing the accuracy of the model's predictions. The model's performance was measured by comparing its predictions against the ground truth labels. It should be noted that the model was set to evaluation mode during testing, and the final accuracy, along with the test loss, was reported after completing the evaluation process. The achieved accuracy of 34.93% suggests the current level of performance, which serves as a baseline for further model refinement and optimization in subsequent iterations.

4 Results and Discussion

4.1 Initial Results

In the initial evaluation, the LSTM-based model yielded an accuracy of 34.93%. This accuracy metric signifies the proportion of correct predictions made by the model among the total predictions. The achieved accuracy suggests that there is room for improvement in the model's performance for genre classification. Moving forward, it is recommended to delve deeper into analyzing if the instances that were misclassified to identify patterns or characteristics that can offer adjustments to the model architecture, hyperparameters, or data preprocessing steps.

References

Tsuchida, S., Fukayama, S., Hamasaki, M., & Goto, M. (2019). Aist dance video database: Multi-genre, multi-dancer, and multi-camera database for dance information processing. *ISMIR*, *1*(5), 6.

Chan, C., Ginosar, S., Zhou, T., & Efros, A. A. (2019). Everybody dance now. *Proceedings of the IEEE/CVF international conference on computer vision*, 5933–5942.

Moltisanti, D., Wu, J., Dai, B., & Loy, C. C. (2022). Brace: The breakdancing competition dataset for dance motion synthesis. *European Conference on Computer Vision*, 329–344.

aswathisasidharan. (2023, June). Support vector machine (svm) algorithm. https://www.geeksforgeeks.org/support-vector-machine-algorithm/

Banoula, M. (2023, April). Introduction to long short-term memory(lstm): Simplilearn. https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/lstm

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, *404*, 132306.

Bauskar, D. (2022). *Dance video classification into relevant street dancing styles using deep learning techniques* [Doctoral dissertation, Dublin, National College of Ireland].

Gupta, S. (2023, September). Rnn vs. cnn: Which neural network is right for your project? https://www.springboard.com/blog/data-science/rnn-vs-cnn/

Craig, L. (2023, August). Cnn vs. rnn: How are they different?: Techtarget. https://www.techtarget.com/searchenterpriseai/feature/CNN-vs-RNN-How-they-differ-and-where-they-overlap

Bendit-Shtull, N., Cheng, T., & Wang, J. (n.d.). Who's got the groove? classifying dance style from video.

Castro, D., Hickson, S., Sangkloy, P., Mittal, B., Dai, S., Hays, J., & Essa, I. (2018). Let's dance: Learning from online dance videos. *arXiv preprint arXiv:1801.07388*.

Brownlee, J. (2020, October). A gentle introduction to cross-entropy for machine learning. https://machinelearningmastery.com/cross-entropy-for-machine-learning/#:~:

text=)%3A%203.288%20bits-,Cross%2DEntropy%20as%20a%20Loss%20Function,be%20used%20for%20classification%20tasks.

Saxena, S. (2023, October). What is lstm? introduction to long short-term memory. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/#:~:text=LSTM%20(Long%20Short%2DTerm%20Memory,ideal%20for%20sequence%20prediction%20tasks.