공학박사학위논문

Performance Improvement of Deep Autoencoders for Computer Vision Models Using Human Body Embeddings

신체 임베딩을 활용한 오토인코더 기반 컴퓨터 비전 모형의 성능 개선

2021 년 8 월

서울대학교 대학원

산업공학과

박 종 혁

Performance Improvement of Deep Autoencoders for Computer Vision Models Using Human Body Embeddings

신체 임베딩을 활용한 오토인코더 기반 컴퓨터 비전 모형의 성능 개선

지도교수 박 종 헌

이 논문을 공학박사 학위논문으로 제출함

2021 년 7월

서울대학교 대학원

산업공학과

박 종 혁

박종혁의 공학박사 학위논문을 인준함

2021 년 7월

위 원 장	조 성 준
부위원장	박 종 헌
위 원	이 재 욱
위 원	박 우 진
위 원	허 재 석

Abstract

Performance Improvement of Deep Autoencoders for Computer Vision Models Using Human Body Embeddings

Jonghyuk Park
Department of Industrial Engineering
The Graduate School
Seoul National University

Deep learning models have dominated the field of computer vision, achieving state-of-the-art performance in various tasks. In particular, with recent increases in images and videos of people being posted on social media, research on computer vision tasks for analyzing human visual information is being used in various ways.

This thesis addresses classifying fashion styles and measuring motion similarity as two computer vision tasks related to humans. In real-world fashion style classification problems, the number of samples collected for each style class varies according to the fashion trend at the time of data collection, resulting in class imbalance. In this thesis, to cope with this class imbalance problem, generalized few-shot learning, in which both minority classes and majority classes are used for learning and evaluation, is employed. Additionally, the modalities of the foreground images, cropped to show only the body and fashion item parts, and the fashion attribute information are reflected in the fashion image embedding through a variational autoencoder. The

K-fashion dataset collected from a Korean fashion shopping mall is used for the model training and evaluation.

Motion similarity measurement is used as a sub-module in various tasks such as action recognition, anomaly detection, and person re-identification; however, it has attracted less attention than the other tasks because the same motion can be represented differently depending on the performer's body structure and camera angle. The lack of public datasets for model training and evaluation also makes research challenging. Therefore, we propose an artificial dataset for model training, with motion embeddings separated from the body structure and camera angle attributes for training using an autoencoder architecture. The autoencoder is designed to generate motion embeddings for each body part to measure motion similarity by body part. Furthermore, motion speed is synchronized by matching patches performing similar motions using dynamic time warping. The similarity score dataset for evaluation was collected through a crowdsourcing platform utilizing videos of NTU RGB+D 120, a dataset for action recognition.

When the proposed models were verified with each evaluation dataset, both outperformed the baselines. In the fashion style classification problem, the proposed model showed the most balanced performance, without bias toward either the minority classes or the majority classes, among all the models. In addition, In the motion similarity measurement experiments, the correlation coefficient of the proposed model to the human-measured similarity score was higher than that of the baselines.

Keywords: Human body, Fashion, Motion analysis, Computer vision application, Autoencoder, Industrial engineering

Student Number: 2016-21106

Contents

Abstract **i**

Contents **vi**

List of Tables **viii**

List of Figures **xiii**

Chapter 1 Introduction **1**

1.1 Background and motivation 1

1.2 Research contribution 5

1.2.1 Fashion style classification 5

1.2.2 Human motion similarity 9

1.2.3 Summary of the contributions 11

1.3 Thesis outline 13

Chapter 2 Literature Review **14**

2.1 Fashion style classification 14

2.1.1 Machine learning and deep learning-based approaches 14

2.1.2 Class imbalance 15

2.1.3 Variational autoencoder 17

2.2 Human motion similarity 19
2.2.1 Measuring the similarity between two people 19
2.2.2 Human body embedding 20
2.2.3 Datasets for measuring the similarity 20
2.2.4 Triplet and quadruplet losses 21
2.2.5 Dynamic time warping 22

Chapter 3 Fashion Style Classification **24**

3.1 Dataset for fashion style classification: K-fashion 24
3.2 Multimodal variational inference for fashion style classification . . . 28
3.2.1 CADA-VAE 31
3.2.2 Generating multimodal features 33
3.2.3 Classifier training with cyclic oversampling 36
3.3 Experimental results for fashion style classification 38
3.3.1 Implementation details 38
3.3.2 Settings for experiments 42
3.3.3 Experimental results on K-fashion 44
3.3.4 Qualitative analysis 48
3.3.5 Effectiveness of the cyclic oversampling 50

Chapter 4 Motion Similarity Measurement **53**

4.1 Datasets for motion similarity 53
4.1.1 Synthetic motion dataset: SARA dataset 53
4.1.2 NTU RGB+D 120 similarity annotations 55
4.2 Framework for measuring motion similarity 58

4.2.1 Body part embedding model 58
4.2.2 Measuring motion similarity 67
4.3 Experimental results for measuring motion similarity 68
4.3.1 Implementation details 68
4.3.2 Experimental results on NTU RGB+D 120 similarity annotations 72
4.3.3 Visualization of motion latent clusters 78
4.4 Application 81
4.4.1 Real-world application with dancing videos 81
4.4.2 Tuning similarity scores to match human perception 87

Chapter 5 Conclusions **89**
5.1 Summary and contributions 89
5.2 Limitations and future research 91

Appendices **93**

Chapter A NTU RGB+D 120 Similarity Annotations **94**
A.1 Data collection 94
A.2 AMT score analysis 95

Chapter B Data Cleansing of NTU RGB+D 120 Skeletal Data **100**

Chapter C Motion Sequence Generation Using Mixamo **102**

Bibliography **104**

국문초록 **123**

List of Tables

Table 3.1 Distribution of the sampled K-fashion dataset [1] used for the experiments. 27

Table 3.2 Comparing MVStyle to the baselines on K-fashion. All numbers are percent accuracy, and the maximum value of each column is marked in bold. 45

Table 3.3 Performances of MVStyle using various combinations of modalities. All numbers are percent accuracy, and the maximum value of each column is marked in bold. T3 avg and T3_H avg represent the average of T3 and T3_H for all sampled datasets, respectively. 47

Table 3.4 Comparing the cyclic oversampling to the baselines on K-fashion $\rho = 110.84$. All numbers are T3_H represented in percent. The best performance is marked in bold. 52

Table 4.1 SARA dataset overview. 54

Table 4.2 Rank correlations with AMT scores. 73

Table 4.3 Ablation study on the proposed loss function. 74

Table 4.4 Motion similarity by body part for the sample pairs in Figure 4.7. Body parts with relatively lower similarity scores are marked bold. 78

List of Figures

Figure 1.1 The number of images retrieved by searching fashion style class labels such as "street fashion" on Instagram. The search took place on December 7, 2020. 3

Figure 1.2 Difference between few-shot learning and generalized few-shot learning. 6

Figure 1.3 Examples of the foreground image and the attributes for a fashion image. 8

Figure 1.4 A high-level overview of the proposed method. The model takes a sequence of human joint coordinates and produces embeddings of body parts which are used to analyze the similarity between different motions. 10

Figure 2.1 Visualization of DTW algorithm. (a) is a figure describing the process of finding the optimal alignment between two time series P and Q. (b) shows that the elements of the time series P and Q were matched according to the optimal alignment obtained by DTW. 23

Figure 3.1 Examples of images from K-fashion [1] dataset. 25

Figure 3.2 Visualization of CADA-VAE [2] and classifier for fashion style classification. (a) describes the structure of CADA-VAE using three modalities. (b) shows the style classifier using only the image encoder trained in (a). When fitting the classifier, the number of sampled latent variables (s) is adjusted by the proposed cyclic oversampling. 30
Figure 3.3 Examples of foreground images. 34
Figure 3.4 Generating one-hot vector representing the fashion attributes. 35
Figure 3.5 Illustration of ResNet-50 architecture. 39
Figure 3.6 Examples in which the generated foreground image captures human body and fashion items. 49
Figure 3.7 Examples that is judged as the correct answer when the attributes modality is included. 50
Figure 3.8 Sampling schedule of the cyclic oversampling and the comparisons. Each schedule method is drawn in a different color. 51

Figure 4.1 The examples of the AMT annotations pair from NTU RGB+D 120 [3] dataset. (a), (b), (c), and (d) are an example of scores 4, 3, 2, and 1 respectively; (e) is the histogram of the total collected scores. 57
Figure 4.2 Body parts decomposition. Middle hip is the origin of the coordinate system. 60
Figure 4.3 Visualization of the proposed model. Each body part is drawn in a different color. 61

Figure 4.4 Visualization of the motion variation loss. (a) shows a situation where loss brings positive and semi-positive samples closer when they are mapped far; (b) indicates that the loss drives them far when they are mapped close. 63

Figure 4.5 Visualization of the motion variation for positive and semi-positive samples of the SARA dataset. The motion variation is computed from two samples that belong to the same motion class but have different characteristics (e.g., **Energy**). . . . 64

Figure 4.6 Network structure of encoders and decoders. Except for the camera view encoder, encoders and decoders are equal in number to body parts. 70

Figure 4.7 Pairs (from NTU RGB+D 120 [3]) for body part similarity in Table 4.4. 77

Figure 4.8 Visualization of motion latent vectors. The motion classes of the SARA validation set are clustered by colors in the left part of (a). The dark green (*Adventure152*) and light yellow (*Dance132*), circled in black, correspond to the similar motions that were performed while standing with the elbows bent and leaning back (shown in the right part of (a)). The visualization of 21 sampled actions of NTU RGB+D 120 [3] is made in the left part of (b). The blue (*cheer up*) and red (*stretch oneself*) positioned on the upper right, represent similar motions (shown in the right part of (b)). 80

Figure 4.9 Illustration example comparing two similar dance sequences by body part. A threshold of 0.4 was chosen to separate similar (green) motions. 82
Figure 4.10 Illustration example comparing two similar dance sequences by body part. A threshold of 0.4 was chosen to separate similar (green) motions. 83
Figure 4.11 Illustration example comparing two dissimilar dance sequences by body part. A threshold of 0.4 was chosen to separate similar (green) and different (red) motions. 84
Figure 4.12 Illustration example comparing two dissimilar dance sequences by body part. A threshold of 0.4 was chosen to separate similar (green) and different (red) motions. 85
Figure 4.13 Relationship between the similarity scores of NTU RGB+D 120 similarity annotations and the similarity measured from the proposed framework. The x-axis represents the similarity measured from the proposed framework, and the y-axis represents the values obtained by normalizing the similarity scores of 1-4 points collected from humans to a 0-1 scale. The relationship can be approximated by the logistic function. . 88

Figure A.1 Web page for AMT instructions. All video clips are from NTU RGB+D 120 [4, 3]. 96
Figure A.2 Web page for AMT scoring. All video clips are from NTU RGB+D 120 [4, 3]. 97

Figure A.3 Average AMT scores per action category. The blue and red bars mean the average scores when the similarity is measured with samples belonging to the same action category and different action categories, respectively. 99

Figure B.1 Example of cleansing NTU RGB+D 120 [4, 3] skeletal data: In each example, the left side image is the original skeleton data from NTU RGB+D and the right side image is estimated joint annotations by our estimation model. 101

Figure C.1 Motion generation using the Mixamo [5] tool. 103

Chapter 1

Introduction

1.1 Background and motivation

Computer vision, a field that has benefited significantly from the development of deep learning, is applied in various tasks such as object classification, detection, and segmentation [6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. Computer vision technology plays an important role in real-world industries in various ways, such as aiding the analysis of user trends using vast amounts of user-generated images and videos uploaded on social media [16, 17, 18]. Most posts on TikTok, which became the most downloaded application globally with 8.5 million downloads from Google Play and the iOS App Store in 2020, have a short-form video format in which humans appear. The categories of posts featuring people (Entertainment, Dance, Pranks, Fitness/Sports, Home Renovation, Beauty, Fashion) ranked in the top seven most-viewed categories, based on survey results in July 2020 [19]. This example implies that computer vision applications to process images containing people could be more widely utilized.

This thesis addresses two human-related computer vision tasks: image-based fashion style classification and video-based human motion similarity measurement. Fashion is one of the fastest-growing areas and one that could most benefit from the

recent advances in deep learning-based machine vision. Most tasks in this domain, including fashion category classification [20, 21], segmentation [22, 23], retrieval [24, 25], and recommendation [26, 27], have been actively researched and successfully addressed using models based on convolutional neural networks (CNNs) [28]. However, only a few attempts have been made to solve fashion style classification [29, 30, 31], which aims to classify the style of a fashion image. These studies related to fashion style presented a fashion style dataset, providing benchmark performance for it.

Although previous studies have recorded superior performances, they are limited because they do not take into account the class imbalance problem that exists in real-world scenarios. Fashion style classes are highly imbalanced as there are styles that dominate the majority, such as everyday or trendy styles. For instance, fashion images of "Street style" are much more frequent than those of "Hippy style," as shown in Figure 1.1, where the bar graph represents the number of images retrieved by searching the fashion style class labels on Instagram. Therefore, training the model without considering that fashion style is divided into majorities and minorities in real-world problems will produce reasoning biased toward the majority.

Human motion, essentially a combination of translational and rotational motions of each body joint, contains a significant amount of information inherent to a human. In particular, motion similarity, which can be obtained by analyzing human motion, has a wide range of applications. For instance, motion similarity can be used for action recognition [32, 33, 34, 35, 36, 37]. It is also possible to measure motion similarity to determine whether a task is performed well [38, 39, 40, 41, 42] or identify abnormal behavior [43, 44, 45]. A motion comparison system is helpful for matching a target person from different cameras for re-identification [43, 45, 46, 47,

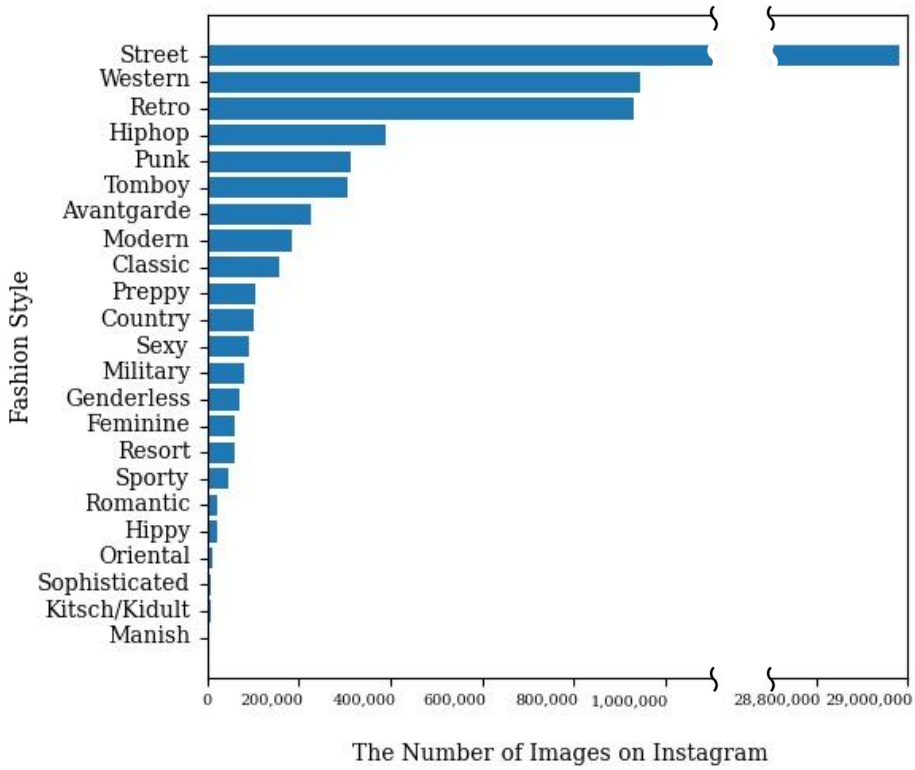


Figure 1.1: The number of images retrieved by searching fashion style class labels such as "street fashion" on Instagram. The search took place on December 7, 2020.

48, 49].

While analyzing human motion plays an essential role in the tasks mentioned above, motion similarity research has attracted less attention so far than the other research areas due to the following reasons. First, measuring motion similarity is a challenging problem. Different camera views or human body structures cause a variety of 2D joint coordinates, even for similar motions in videos. This makes it impossible to measure the similarity directly using the joint coordinates. Second, the availability of large-scale datasets for learning motion similarity is limited. Finally,

to the best of our knowledge, few human motion datasets are available to assess the performance of different motion similarity computation methods.

In this thesis, we propose models that address both points mentioned above and evaluate the models to verify their effectiveness.

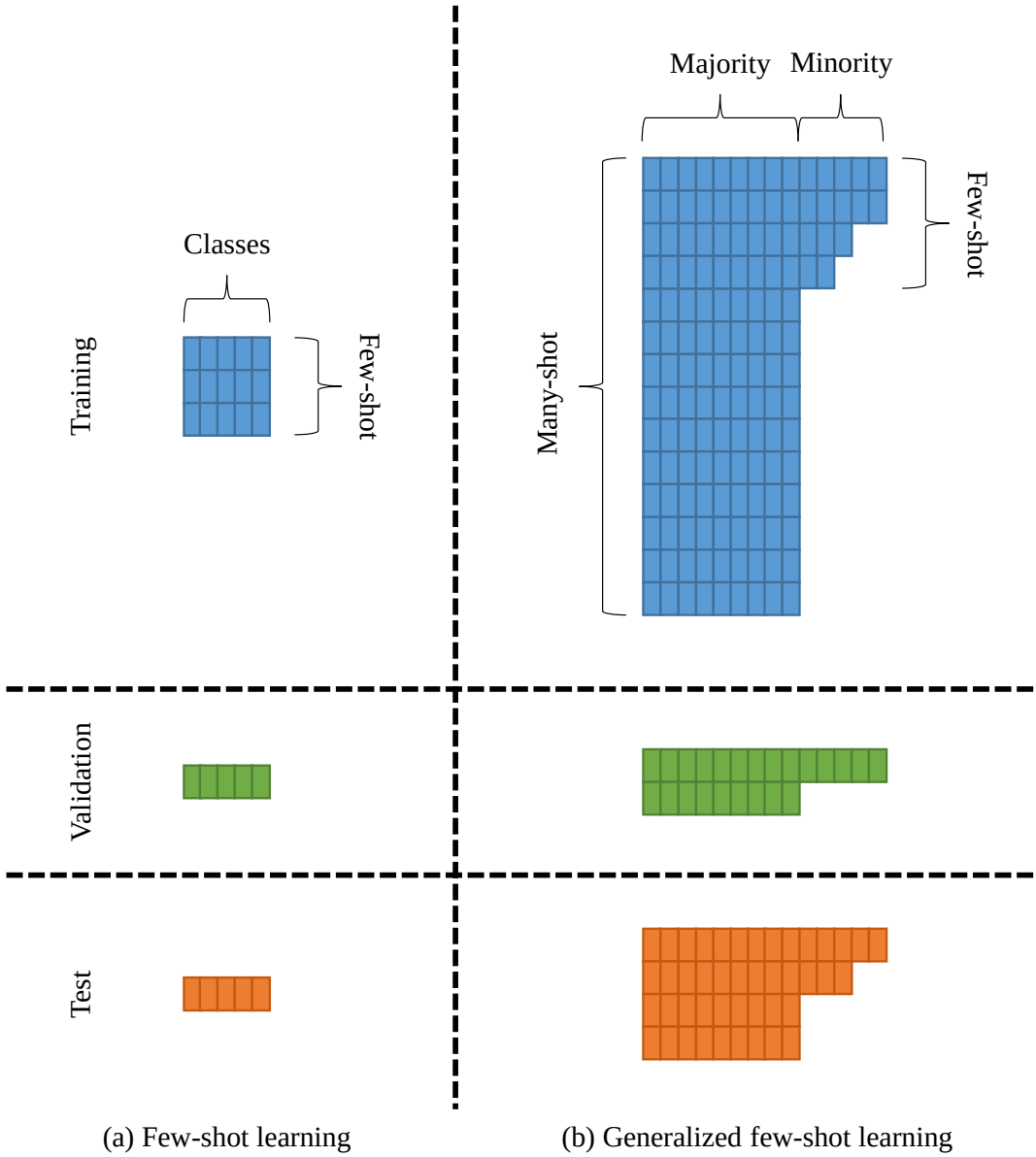
1.2 Research contribution

As mentioned above, this thesis deals with two subjects: fashion style classification and human motion similarity. In this section, we address contributions to improve the performance of each subject using embeddings generated from body information in images.

1.2.1 Fashion style classification

As fashion images exposed through social media increases, the fashion style classification model has more opportunities to be used, such as recommending fashion items and providing customized advertisements to users, by recognizing the fashion style of the images. In collecting data for these real-world fashion style classification models, class imbalance occurs due to fashion trends, as mentioned in Section 1.1. This imbalance makes it possible to formulate the fashion style classification as generalized few-shot learning (GFSL) [2, 50]. This setting, also known as the step imbalance [51] or two-level imbalance [52], divides classes into the majority and minority and uses only a minimal number of samples for the minority classes. Because GFSL uses both data-rich majority classes and data-poor minority classes for model training and evaluation, it is a more generalized problem setting than few-shot learning (FSL), in which the model only trains and evaluates minority classes. Figure 1.2 shows the difference between FSL and GFSL.

When solving a GFSL problem, multimodal information is generally used together to improve the performance of the minority classes. The cross- and distribution-aligned variational autoencoder (CADA-VAE) [2] reflects the information of other modalities by making the latent distribution of the modalities close in the latent



(a) Few-shot learning (b) Generalized few-shot learning

Figure 1.2: Difference between few-shot learning and generalized few-shot learning.

space. DRAGON, which stands for "a moDulaR Approach for lonG-tail classificatiON" [52] rearranges the probabilities of two modalities through late fusion. The above studies improve the performance of minority classes by supplementing information from other modalities.

We address the fashion style classification problem by extending the CADA-VAE framework, which is more suitable for multiple modalities than DRAGON. Performance for the minority classes is maximized by using embeddings generated from two additional modalities: foreground images and attributes specified to the human body information in the image. The foreground images were cropped to only the part of the body wearing a fashion item using the model of [22]. With the foreground images, we force the model to focus on the body and fashion item parts. Attributes describing the characteristics of the clothing item according to the body structure, such as a top or bottom, are generated as one-hot vectors and fed to CADA-VAE. Examples for the foreground image and the attributes for one fashion image are shown in the Figure 1.3.

After the training of CADA-VAE, the proposed cyclic oversampling algorithm adjusts the oversampling ratio for the latent variable of the minority classes every epoch. This adjustment prevents learning biased toward the majority or minority during the training of the classifier. The authors of CADA-VAE reported that the model performed best in situations where the numbers of minority and majority sampled variables are in a specific ratio. However, because this ratio should vary when the dataset changes, we propose a cyclic oversampling algorithm that does not require manual tuning.

The proposed architecture utilizing multimodal variational inference for fash-

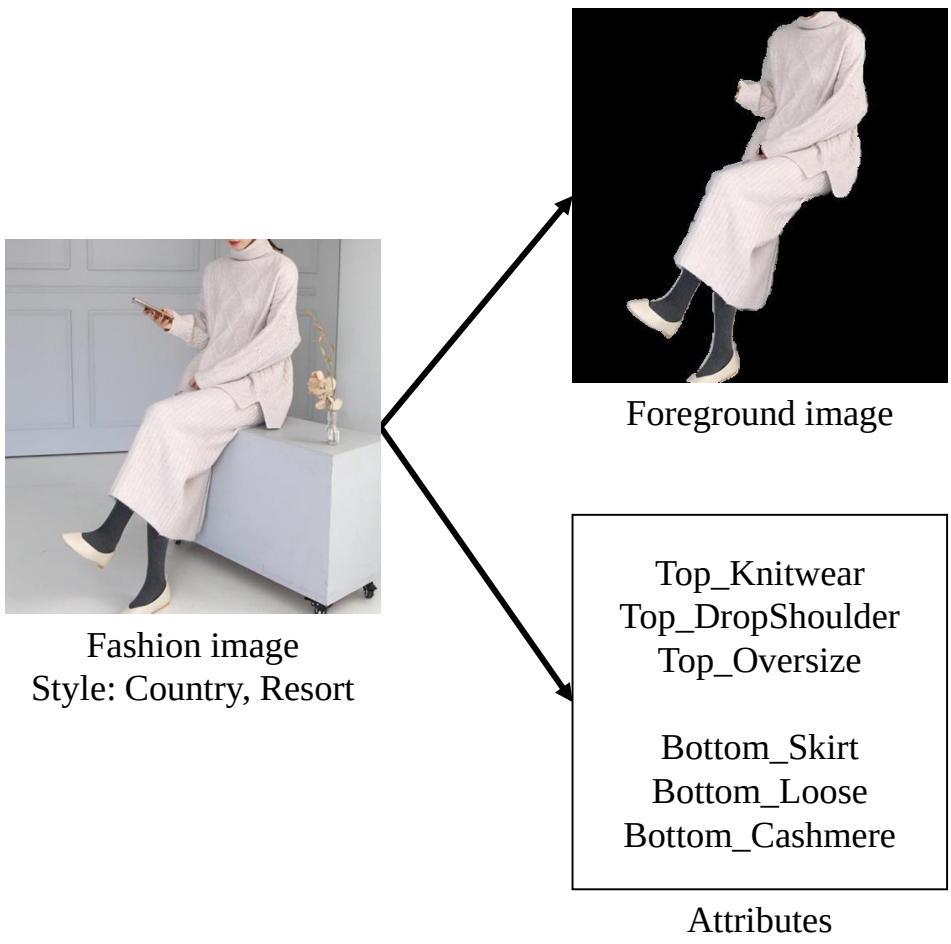


Figure 1.3: Examples of the foreground image and the attributes for a fashion image.

ion style classification (MVStyle) was trained and evaluated with the K-fashion dataset [1], which is composed of multi-label fashion style images collected from shopping malls in Korea. The performance measurement showed that MVStyle outperformed the baselines. Specifically, utilizing the combination of all modalities produced the best performance.

1.2.2 Human motion similarity

Measuring human motion similarity has been utilized for various human-related computer vision tasks, such as action recognition, performance evaluation, anomaly detection, and person re-identification, as mentioned above. This work attempts to compare short video clips of basic human motions. Our target motions are short enough (1−2 s) to be described by a few words or a sentence and can be easily imitated after a demonstration. The comparison is made solely by body movements, excluding differences in body size and appearance. In this work, we represent a motion as a sequence of joint positions and do not consider an interaction between a human and an object in the environment. To build a comparison system, we extend the autoencoder-based model of [53] to split human motion into five body parts and map them to a latent space. The similarity is measured by comparing the encoded motion vectors. The overview of the method for measuring motion similarity is depicted in Figure 1.4.

The proposed model is trained on our synthetic motion dataset, an extended version of the dataset [53] from Adobe Mixamo [5]. We collected human motion animations with variations in characteristic elements (e.g., movement and angle) of each motion. These variations are important for learning the motion similarity, as

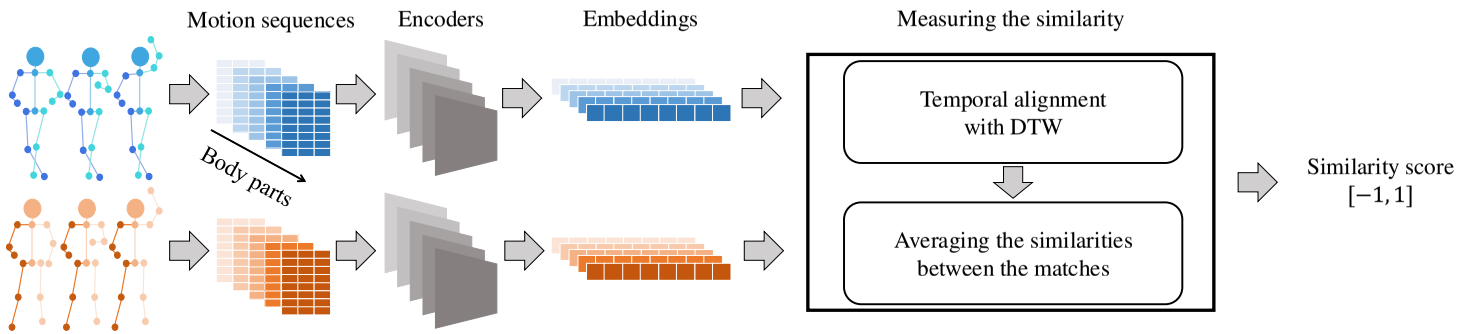


Figure 1.4: A high-level overview of the proposed method. The model takes a sequence of human joint coordinates and produces embeddings of body parts which are used to analyze the similarity between different motions.

they allow us to distinguish between very similar movements.

To incorporate this property into the model effectively, we propose a motion variation loss. This loss forces the distance between two motion embeddings to be proportional to the motion variation. Overall, our goal is to learn disentangled motion embeddings from the skeletons and camera views, in contrast to the existing motion similarity-learning methods utilized in other tasks such as action recognition. The motion embeddings, which are divided into five body parts and learned through motion variation loss in this study, allow robust analysis of the motion similarity.

To assess the performance of the proposed model on real-world data, we utilized the NTU RGB+D 120 [4, 3] dataset, which has been widely used for action recognition [54, 55, 56, 57, 58, 59, 60, 61, 62]. NTU RGB+D 120 is composed of videos where people perform various actions with different camera angles. Because there are no labels in the dataset to measure the similarity, we collected labels via Amazon Mechanical Turk (AMT) [63]. The proposed method achieved a higher correlation between the evaluated similarity scores and human perception than the baseline models. Both datasets and codes have been made publicly available[1].

1.2.3 Summary of the contributions

In summary, our main contributions in this thesis are as follows:

(a) We address two computer vision tasks to process images containing people through autoencoder-based models.

(b) We improve the performance of each problem by learning the embedding of the model to include the human body information in the image.

[1] https://chico2121.github.io/bpe/

(c) The proposed architectures outperform the baselines in each problem.

1.3 Thesis outline

The rest of the thesis is organized as follows. In Chapter 2, we review literature related to the problems. In Chapter 3 and 4, we introduce the datasets used to train the models, explain the proposed architectures for classifying fashion style and measuring motion similarity, and present the experimental results of each task. Finally, in Chapter 5, we give concluding remarks and possible future research directions of this thesis.

Chapter 2

Literature Review

2.1 Fashion style classification

2.1.1 Machine learning and deep learning-based approaches

While most fashion-related problems such as category classification [20, 21], item segmentation [22, 23], item retrieval [24, 25], and item recommendation [26, 27] have been actively studied, only a few studies have been conducted on the fashion style classification problem. Most research projects addressing fashion style classification have been focused on the introduction of fashion datasets, such as FashionStyle14 [29], Hipster Wars [31], and WEARSTYLE [30], with the benchmark model performance.

Specifically, Takagi *et al.* [29] presented style classification using a CNN as the backbone network, and they measured its performance using a newly presented fashion style dataset called FashionStyle14, which included 14 style classes. Kiapour *et al.* [31] released a dataset, Hipster Wars, composed of five style classes and evaluated the performance of each class using various machine learning techniques. Miyamoto *et al.* [30] proposed a dataset called WEARSTYLE based on the images and classes collected from a website. The authors found that style classification performance was improved when using the foreground images, which are images with the back-

ground removed. Based on this insight, we select foreground images as one of the modalities to use. In addition, Ma *et al.* [64] constructed a dataset from images of fashion shows. The dataset was annotated with 527 styles and visual features such as collar shape and pants length. Results produced from the proposed model achieved the best performance among the comparison models.

Nonetheless, the previous studies have a major drawback: the class imbalance problem when collecting large-scale fashion style data is not considered. Furthermore, because the model was evaluated only with a single-label dataset, there are limitations in real-world scenarios.

2.1.2 Class imbalance

General algorithms

Several algorithms have been proposed to train imbalanced datasets effectively. Lin *et al.* [65] developed cross-entropy loss and applied down-weight to well-classified examples to lower the contribution, allowing the model to focus on hard examples. Li *et al.* [66] found that easy examples with many samples and challenging examples close to being outliers significantly influence learning the gradient norm distribution according to the number of samples in the class. Therefore, the authors improved the performance by down-weighting the easy examples and challenging examples. Cui *et al.* [67] calculated the effective number, which means the number of influential samples, and applied it to the loss term to give weights to improve the performance. In addition, Cao *et al.* [68] improved the minority's accuracy by providing a margin dependent on the class's number of samples. Meanwhile, there are studies to solve the class imbalance problem through resampling methods.

The authors improved performance by oversampling the minority classes [69, 70]

and under-sampling the majority classes[71]. In this paper, we oversample minority features from the learned latent distribution. Furthermore, by using the proposed cyclic oversampling, the model does not need to be overfitted for the minority classes.

Generalized few-shot learning

In FSL, classes are not divided into the majority and minority, and only a small number of samples, k, are provided to train the model. In contrast, in GFSL, both the majority and the minority are put into training [2]. To measure the model's performance in the GFSL, Schonfeld *et al.* [2] used the harmonic mean between the accuracy of the majority classes and the accuracy of the minority classes as an evaluation metric. It was shown that the proposed model improved the performance in GFSL situations by utilizing both visual and textual features as input values. Huang *et al.* [72] introduced a model that reduces the variance of latent variables in a class. The authors constructed a loss by adding a term using the k-means clustering algorithm to the loss of [2]. Ye *et al.* [50] tried to solve the GFSL problem by synthesizing a classifier trained with majority classes and a classifier trained with minority classes. Xian *et al.* [73] extended the GFSL problem in image classification to the video classification domain by adopting spatiotemporal CNN and 3D CNN structures.

Learning with multimodal information

[2, 72, 74] aimed at learning joint representation by aligning the distribution of features obtained from different modalities. In [2, 72], a distribution-alignment term and cross-alignment term were added to the VAE loss to reflect multimodal information in the latent distribution. Hubert Tsai *et al.* [74] used maximum mean

discrepancy [75] as the loss to make the distribution between the image and semantic embedding close. Samuel *et al.* [52] rescored the classification probability for each class obtained from an image and attribute information by feeding the number of samples for each class as an input to the model.

By combining two images (images, foreground images) and attributes, we maximize the performance for classifying fashion styles.

2.1.3 Variational autoencoder

A variational autoencoder (VAE) [76] estimates latent distribution parameters from input values through the encoder and puts the sampled latent variable from the distribution into the decoder to reconstruct the input values. A normal distribution is generally used as the prior probability distribution to approximate the latent distribution. The approximation is achieved through Kullback-Leibler (KL) divergence included in the loss function, resulting in the total loss function, including KL divergence, as follows:

$$\mathcal{L} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)\|p_\theta(z)), \tag{2.1}$$

where x and z are the input and latent variables, respectively. The first term on the right-hand side of (2.1) means the reconstruction error of the input. The latent distribution $q_\phi(z|x)$ of z is created from the encoder with parameter ϕ, and the decoder with parameter θ reconstructs x using z sampled from the latent distribution. Then, z is sampled using a reparameterization trick so that the VAE is trained through backpropagation. The second term on the right-hand side of (2.1) is the KL divergence, which approximates the latent distribution to prior distribution, $p_\theta(z)$.

After the VAE was introduced, many VAE variants [2, 72, 77, 78, 79, 80, 81, 82,

83] were published. We adopted the VAE of [2] to learn the latent distribution from multimodal information suitable for the fashion domain.

2.2 Human motion similarity

2.2.1 Measuring the similarity between two people

Defining similarity between human poses is a fundamental task for building a video retrieval system, and many studies have approached it in various ways. Ferrari *et al.* [84] defined a feature vector representing a human pose in an image based on the pictorial structure, computed vector space distances between poses in a query image, and individual frames of a video in a database, and they aggregated the distances from all frames to obtain the relevant score for the video track. The system in [85] measured the distance between poses as a function of joint angles and retrieved similar videos containing the frames near a query pose image. Kim and Kim [40] measured the similarity between two dance poses using the joint angles of the person in a frame. In [86], fixed-length short motion sequences were clustered into groups and used as a motion representation. In contrast to [40, 84, 85], in which the methods for either image-level or video-level retrieval rely on pose similarity between two image frames, our method defines motion similarity between a pair of motion sequences directly.

Apart from analyzing independent motions, Shen *et al.* [87] proposed an approach for measuring motion similarity in interaction-based activities. While this approach is promising for tasks of interacting with objects, it cannot be utilized to compare independent movements without interactions or when the objects are located far apart. Moreover, the algorithm is unsuitable for real-time applications. In [35], long short term memory with a layer normalization architecture was utilized to generate motion embeddings. In the training phase, the authors replaced the hard-negative mining required for similarity learning with maximum mean discrepancy (MMD) [75]

and reduced the computational cost. We employ this architecture as one of the main baselines when assessing the performance of our model.

2.2.2 Human body embedding

Decomposing a body into several parts based on the human skeleton structure and constructing representations for individual parts are common approaches to understand human action. For instance, Choutas *et al.* [88] suggested a fixed-size representation of a video clip containing a motion as a collection of trajectory maps of individual joints. Guo and Choi [89] argued that learning local representations separately on four limbs and the torso was helpful for short-term human motion prediction.

Liu *et al.* [90] suggested the hierarchical partwise bag-of-words representation focused on the visual salience of different body areas with seven bag-of-words features (limb, head, leg, foot, upper, lower, and full) in three levels (low level, middle level, and high level). Hake [91] extracted interaction triples—a body part, an action verb, and an object—from images based on the features of part regions. Jammalamadaka *et al.* [92] proposed a method to classify a body part image to a corresponding class and constructed an image embedding vector based on the classification scores. In [92], the learned body part embeddings were not merely intermediate representations for subsequent classifiers but also contained general information for 2D pose reconstruction and could be appropriate for measuring motion similarity.

2.2.3 Datasets for measuring the similarity

Few datasets contain a pair of motions with similarity annotation. Mori *et al.* [93] suggested annotating pose similarity automatically by determining whether the

mean joint distance satisfied a given threshold constraint. Despite the ease of constructing a large-scale dataset, this method could not generate similar pairs of poses in terms of human perception. Other studies [86, 87] evaluated their models against binary classification or retrieval test sets constructed from action recognition datasets by regarding motions of the same action label as similar. In [87], the authors defined an evaluation task in which a comparison system was required to assign higher similarities to motions that shared more specific class labels for a query motion. However, this class-based strategy could not correctly capture the intraclass variation of motion, as actions from the same class might be less similar than a pair of actions from different classes.

Motivated by the lack of motion similarity datasets, we propose a new dataset containing motion similarity annotations obtained from crowd workers for approximately $20,000$ video pairs.

2.2.4 Triplet and quadruplet losses

Schroff *et al.* [94] proposed a triplet loss that took three images as input. Specifically, the input was composed of a reference image of a person (the anchor), another image of the same person (the positive sample), and an image of a different person (the negative sample). The loss minimizes the distance of anchor-positive features while maximizing the distance of anchor-negative features. The triplet loss has been actively applied in many studies. Wohlhart and Lepetit [95] utilized it to predict classes of objects and 3D poses. Hermans *et al.* [96] proposed a triplet loss that included a sampling method, showing state-of-the-art performance in person re-identification. Kim *et al.* [97] proposed a new triplet loss using continuous labels that preserve the distance ratio of numeric labels in the learned latent space, allowing

the model to learn the degree of similarity, not just the order. Meanwhile, [98, 99] learned the distance between features using four samples. Using the triplet loss as a basis, the authors constrained the minimum interclass distance to be larger than the maximum intraclass distance. While such approaches focused on the interclass separation through a manually defined constraint, we aim to map the distance in a latent space between the intraclass samples using ground-truth motion variation labels.

2.2.5 Dynamic time warping

Dynamic time warping (DTW) [100] is an algorithm that determines the optimal alignment of two time series with different lengths. It can be utilized to measure motion similarity, as it allows the comparison of two motions with varying speeds [101]. Let $P = (\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_{T_P}) \in \mathbb{R}^{h_\mathbf{p} \times T_P}$ denote the time series of $h_\mathbf{p}$-dimensional vectors with time-length T_P. Similar to P, $Q = (\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{T_Q}) \in \mathbb{R}^{h_\mathbf{q} \times T_Q}$ represents the time series of $h_\mathbf{q}$-dimensional vectors with time length T_Q. To align the two time series, DTW constructs cost matrix $\mathbf{D} \in \mathbb{R}^{T_P \times T_Q}$ using dynamic programming. Each matrix element $\mathbf{D}_{ij} = d(\mathbf{p}_i, \mathbf{q}_j)$ is the cost between $\mathbf{p}_i$ and $\mathbf{q}_j$, where $i \in [1 : T_P]$, $j \in [1 : T_Q]$ and $d(\cdot)$ is a distance metric. The optimal alignment is the path with the smallest sum of costs from $\mathbf{D}_{11}$ to $\mathbf{D}_{T_P T_Q}$, like the gray path in Figure 2.1 (a). The path obtained in this way is matched not with the same time points, but with the points of a similar pattern, as shown in Figure 2.1 (b). We utilize DTW to align two motions and calculate their similarity.

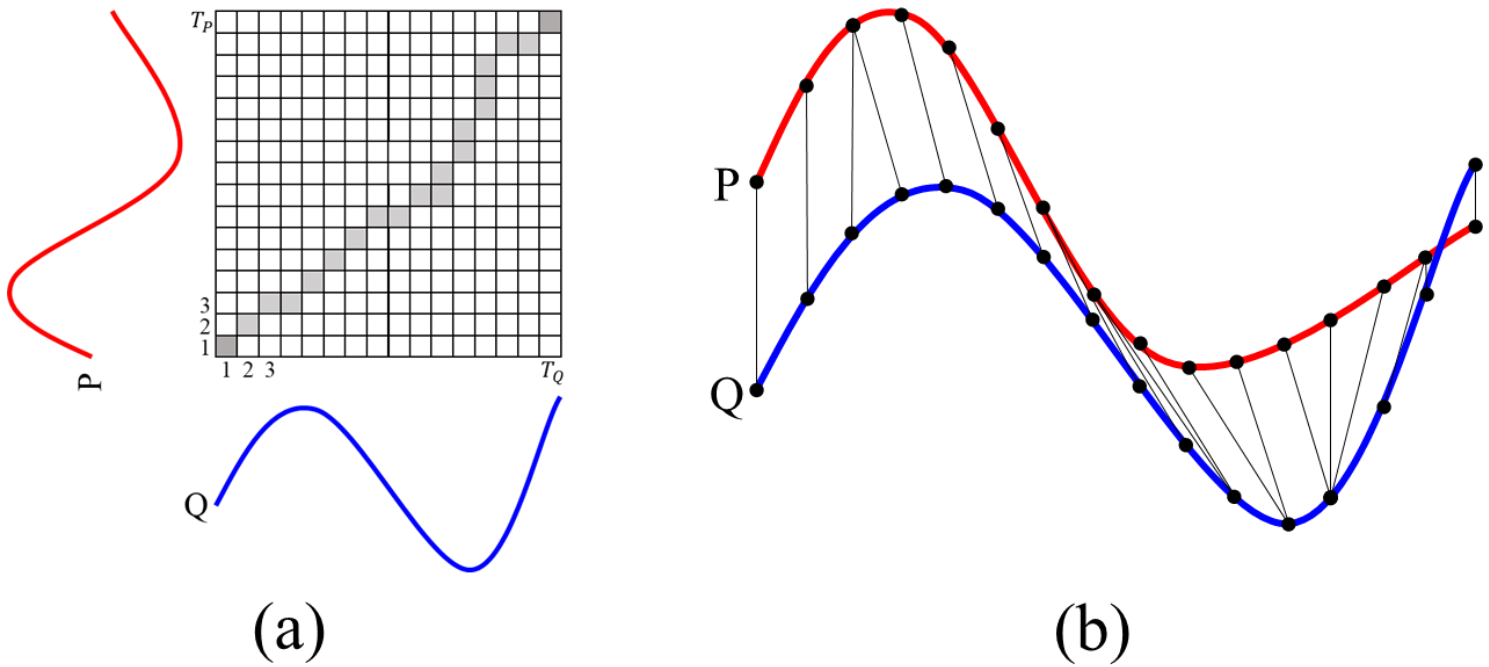


Figure 2.1: Visualization of DTW algorithm. (a) is a figure describing the process of finding the optimal alignment between two time series P and Q. (b) shows that the elements of the time series P and Q were matched according to the optimal alignment obtained by DTW.

Chapter 3

Fashion Style Classification

3.1 Dataset for fashion style classification: K-fashion

K-fashion [1] is a dataset collected by the National Information Society Agency in Korea and consists of 1.2 million women's fashion images from Korean shopping malls. Fashion experts selected 23 style classes, and labels for each image include up to two style classes. Fashion-related attributes (e.g., collar shape, pattern, and fabric) were also collected so that users can use them as side information. The attributes of each image were chosen from a total of 186 attributes determined by fashion experts. Example images for each style class are shown in Figure 3.1.

Since K-fashion is a vast dataset containing 1.2 million fashion images, we sampled the dataset to make the experiments more efficient. The training and validation sets were randomly sampled and reconstructed considering various class imbalance situations and the ratio of the majority- and minority-class samples. K-fashion is an unbalanced dataset with a long-tail distribution, so eight tail classes were selected as the minority classes, similar to the ratio between the majority and minority classes in [2, 102]. Furthermore, the ratio between the maximum number of samples in the majority classes and the minimum number of samples in the minority classes, ρ [51], is adopted to represent the degree of imbalance. The datasets were sampled

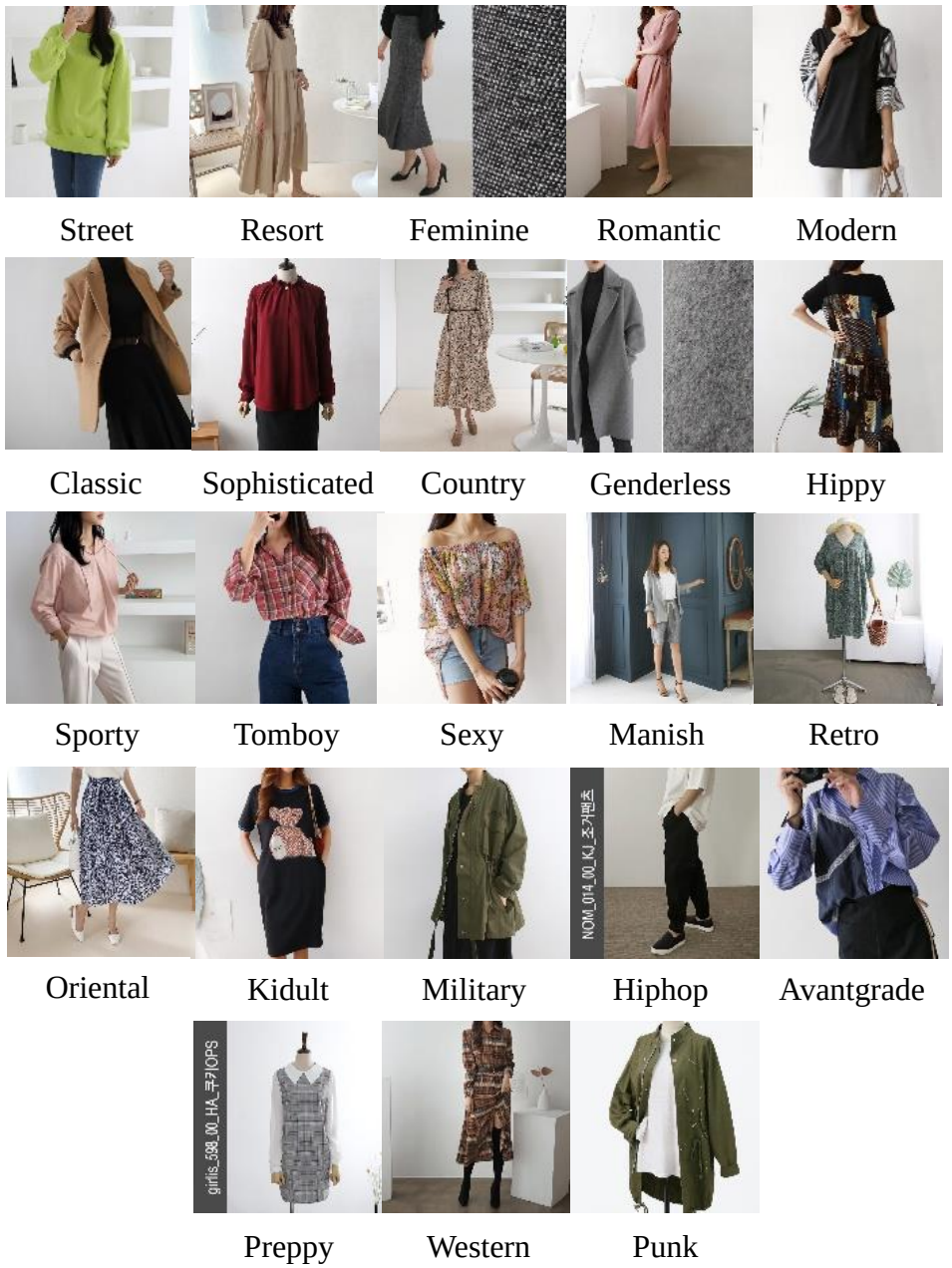


Figure 3.1: Examples of images from K-fashion [1] dataset.

with different degrees of imbalance at $\rho = 373.36$, 232.72, 110.84, 73.91, and 40.13. In addition, the maximum and minimum number of samples in classes belonging to the minority were not different by more than 100 samples. Table 3.1 shows the distribution of the number of samples by class when $\rho = 110.84$.

Table 3.1: Distribution of the sampled K-fashion dataset [1] used for the experiments.

Class split	Class name	Train					Test
		$\rho = 373.36$	$\rho = 232.72$	$\rho = 110.84$	$\rho = 73.91$	$\rho = 40.13$	
Majority classes	Street	4,107	4,189	4,101	4,287	4,494	30,593
	Resort	1,695	1,784	1,745	1,728	1,743	9,156
	Feminine	1,876	1,918	1,957	1,960	1,964	7,936
	Romantic	1,760	1,708	1,789	1,793	1,763	7,470
	Modern	1,946	1,890	1,909	1,924	2,002	7,702
	Classic	1,459	1,459	1,432	1,446	1,419	3,989
	Sophistcated	1,345	1,335	1,322	1,313	1,336	3,653
	Country	1,338	1,354	1,376	1,401	1,395	4,225
	Genderless	1,299	1,282	1,265	1,289	1,307	1,848
	Hippy	1,080	1,077	1,089	1,105	1,086	1,048
	Sporty	1,113	1,100	1,105	1,103	1,129	1,591
	Tomboy	1,096	1,090	1,098	1,096	1,104	838
	Sexy	1,087	1,090	1,097	1,091	1,090	839
	Manish	1,145	1,150	1,173	1,160	1,154	911
	Retro	1,096	1,104	1,104	1,092	1,115	893
Minority classes	Oriental	77	94	116	132	177	569
	Kidult	110	107	114	150	204	776
	Military	34	47	70	85	137	602
	Hiphop	49	58	73	100	166	340
	Avantgard	50	64	83	104	153	410
	Preppy	64	59	88	118	159	165
	Western	32	35	54	81	120	140
	Punk	11	18	37	58	112	88

3.2 Multimodal variational inference for fashion style classification

In this section, we first present a formal definition of the GFSL setting of the fashion style classification problem as follows. Let $\mathcal{G} = \{g_1, g_2, \ldots, g_{n_\mathcal{G}}\}$ denote the set of $n_\mathcal{G}$ modalities. For example, one of the possible modality sets is {Images, Foreground images, Attributes}. We are given a training set $\mathcal{D}^{tr} = \{(x, y)|x \in X, y \in Y\}$ where $X = \{x_g|g \in \mathcal{G}\}$ is the set of features, x_g, generated from modality g. In addition, Y is the union of the set of majority classes, Y_{maj}, and minority classes' set, Y_{min}:

$$Y = Y_{maj} \cup Y_{min}, \tag{3.1}$$

$$Y_{maj} = \{y|n_y > bound\},\ Y_{min} = \{y|n_y \leq bound\}, \tag{3.2}$$

where n_y is the number of samples in class y and *bound* means the boundary value that divides the majority and minority classes.

As depicted in Figure 3.2, our MVStyle training consists of two phases: CADA-VAE [2] training and the classifier training. Figure 3.2 (a) shows the CADA-VAE structure using the multimodal features. It helps the latent distribution of fashion images to contain information about different modalities. The final classifier is fitted using latent variables sampled from the latent distribution, depicted in Figure 3.2 (b). Here, the proposed cyclic oversampling controls the number of minority classes' latent variables provided to the classifier for each epoch, which prevents the model from being too biased towards either the majority or the minority. Moreover, the classifier adopts binary cross-entropy as a loss term for learning on a multi-label dataset. In subsection 3.2.1, we first present CADA-VAE and the classifier. Then, the process of generating multimodal inputs used in MVStyle is explained in subsec-

tion 3.2.2. Finally, the proposed cyclic oversampling is described in subsection 3.2.3.

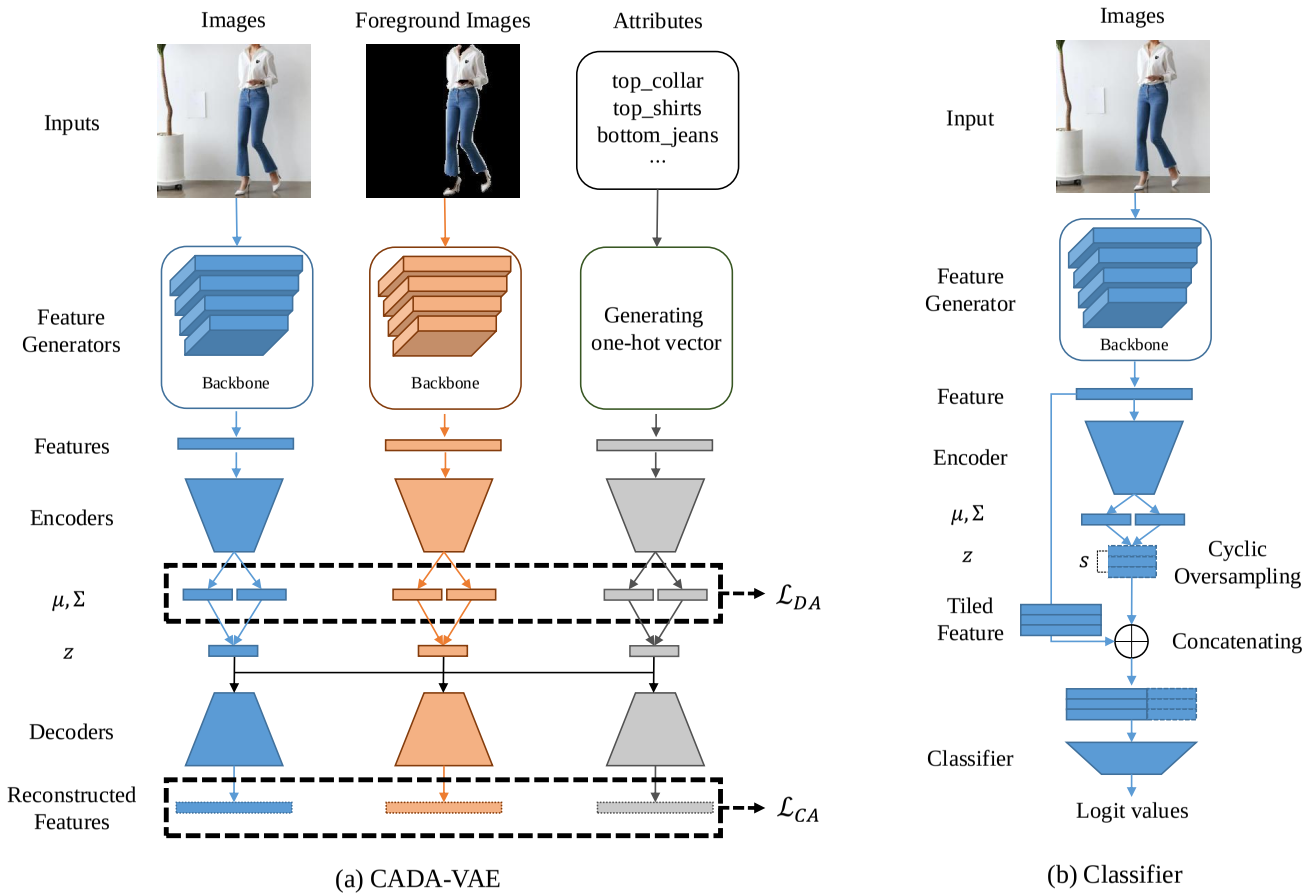


Figure 3.2: Visualization of CADA-VAE [2] and classifier for fashion style classification. (a) describes the structure of CADA-VAE using three modalities. (b) shows the style classifier using only the image encoder trained in (a). When fitting the classifier, the number of sampled latent variables (s) is adjusted by the proposed cyclic oversampling.

3.2.1 CADA-VAE

CADA-VAE [2] is characterized by training the encoder-decoder structure through VAE loss for each modality and allowing information from each modality to be mapped on the same latent space with the cross-alignment loss and distribution-alignment loss.

The encoder of modality g, E_g, maps the input feature x_g into a latent variable $z_g \sim N(\mu_g, \Sigma_g)$, where N is a standard multivariate Gaussian distribution. The VAE loss calculated for each modality is summed as follows:

$$\mathcal{L}_{VAE} = \sum_{g} \mathbb{E}_{q_\phi(z_g|x_g)}[\log p_\theta(x_g|z_g)] - \beta KLD(q_\phi(z_g|x_g)\|p_\theta(z_g)), \tag{3.3}$$

where β is the weight of the KL divergence [79].

The cross-alignment loss ($\mathcal{L}_{CA}$) calculates the reconstruction error through the latent variables obtained from different modalities. Through this term, latent distributions are induced to be aligned.

$$\mathcal{L}_{CA} = \sum_{g} \sum_{l \neq g} \|x_l - D_l(z_g)\|_1, \tag{3.4}$$

where D_l is the decoder of modality l.

The distribution-alignment loss ($\mathcal{L}_{DA}$) encourages alignment of the latent distribution. It is designed to minimize the distance between latent multivariate Gaussian distributions with the 2-Wasserstein distance [103].

$$\mathcal{L}_{DA} = \sum_{g} \sum_{l \neq g} (\|\mu_g - \mu_l\|_2^2 + Tr(\Sigma_g) + Tr(\Sigma_l) - 2(\Sigma_g^{\frac{1}{2}} \Sigma_g \Sigma_l^{\frac{1}{2}})^{\frac{1}{2}}). \tag{3.5}$$

Since the encoder generates the diagonal covariance matrices, this loss simplifies to:

$$\mathcal{L}_{DA} = \sum_{g} \sum_{l \neq g} (\|\mu_g - \mu_l\|_2^2 + \|\Sigma_g^{\frac{1}{2}} - \Sigma_l^{\frac{1}{2}}\|_{Frobenius}^2)^{\frac{1}{2}}. \tag{3.6}$$

The total loss $\mathcal{L}_{CADA\text{-}VAE}$ is the weighted sum of $\mathcal{L}_{VAE}$, $\mathcal{L}_{CA}$, and $\mathcal{L}_{DA}$:

$$\mathcal{L}_{CADA\text{-}VAE} = \mathcal{L}_{VAE} + \gamma_1 \mathcal{L}_{CA} + \gamma_2 \mathcal{L}_{DA}, \tag{3.7}$$

where γ_1 and γ_2 are the weighting factors of $\mathcal{L}_{CA}$ and $\mathcal{L}_{DA}$, respectively.

3.2.2 Generating multimodal features

As mentioned previously, we used a total of three modalities. Image-based modalities consist of an original image and a foreground image, which are known to be effective in classifying styles [30]. Foreground images were generated with non-fashion items removed to focus on the fashion items of the human body in the images. Furthermore, since some images collected in shopping malls include enlarged materials or patterns (see Figure 3.1, an example of K-fashion's Feminine label), we protect the model from such noise by using the foreground images for the model training. The mask for each fashion category was obtained using the Fashionpedia model [22], and all masks were merged to generate the foreground images. Examples of foreground images are shown in Figure 3.3.

In the case of the attribute modality, one-hot vectors, in which 1 was assigned to the attribute's index and 0 was assigned to the other index, were generated and fed to the model. The one-hot vector consists of 442 dimensions, which is equal to the sum of all attribute types corresponding to top, bottom, outer, and one-piece dresses. Since one fashion image can have multiple attributes, there are multiple one values in one-hot vector as many as the number of attributes. The process of generating these vectors is shown in Figure 3.4.

(a) Original Images (b) Foreground Images

Figure 3.3: Examples of foreground images.

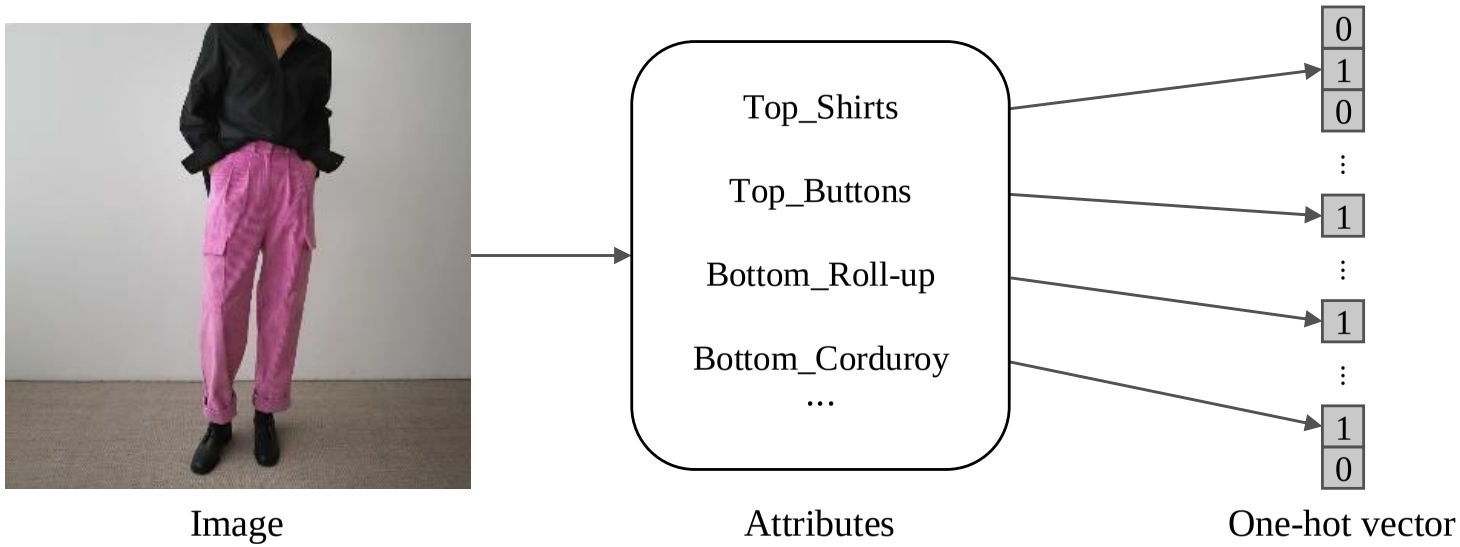


Figure 3.4: Generating one-hot vector representing the fashion attributes.

3.2.3 Classifier training with cyclic oversampling

Only the image modality is utilized for classifier training. Let $dim(z_I)$ and $dim(x_I)$ denote the dimension of latent variable z_I and feature x_I for the image modality, respectively. $z_I \in \mathbb{R}^{dim(z_I)}$ is sampled from the learned latent distribution of CADA-VAE, as shown in Figure 3.2 (b). By concatenating z_I with $x_I \in \mathbb{R}^{dim(x_I)}$ obtained from the image feature generator, an input vector $\mathbf{u} \in \mathbb{R}^{dim(z_I)+dim(x_I)}$ of the classifier is produced. Finally, $\mathbf{u}$ is passed through the fully connected layer to calculate the logit value for each class, and the model is trained with cross-entropy loss.

In addition to the above method, we adopt an oversampling technique to accelerate overcoming the class imbalance. Since the oversampling method causes overfitting for the minority classes [67, 69], accompanying data augmentation is known to help solve the overfitting [69, 104]. We alleviate the overfitting by performing augmentation at the feature level rather than the input data level through variational inference. However, excessive oversampling leads to a decrease in performance for the majority classes. Therefore, we propose the cyclic oversampling detailed in Algorithm 1 to determine the effective oversampling quantities. The cyclic oversampling first measures the validation loss of the minority class. If the loss is larger than the loss of the previous epoch, the number of sampled latent variables for the minority classes is increased in the training of the next epoch. In contrast, if the loss is smaller than the loss of the previous epoch, the number of sampled latent variables for minority classes is reduced in the training of the next epoch, increasing the majority's proportion during the classifier training. This adjustment is repeated until the training ends so that the classifier does not lean toward either the majority or the minority classes during the training.

Algorithm 1: Cyclic oversampling

Input : the number of training epochs n_e, classifier CLS,
number to sample for minority classes s_{min},
number to sample for majority classes s_{maj},
adjustment ratio adj,
training dataset:
$\mathcal{D}_I^{tr} = \{(x_I^{t_1}, y^{t_1}), \ldots, (x_I^T, y^T)\}$,
validation dataset:
$\mathcal{D}_I^{val} = \{(x_I^{v_1}, y^{v_1}), \ldots, (x_I^V, y^V)\}$,
pre-trained latent distribution $q_\phi(z_I|x_I)$

$s \leftarrow s_{min}$
$loss_b \leftarrow \infty$
for $e = 1$ **to** n_e **do**
  $Z \leftarrow$ new List()
  foreach $(x_I^t, y^t) \in D_I^{tr}$ **do**
    if *any class of* y^t *in minority classes* **then**
      sample z_I^t s times from $q_\phi(z_I|x_I)$
      append the all sampled z_I^t and y^t to Z
    else
      sample z_I^t s_{maj} times from $q_\phi(z_I|x_I)$
      append the all sampled z_I^t and y^t to Z
    end
  end
  train the classifier using Z
  $loss \leftarrow 0$
  foreach $(x_I^v, y^v) \in D_I^{val}$ **do**
    if *any class of* y^v *in minority classes* **then**
      $loss^v \leftarrow$ loss of x_I^v calculated from CLS
      $loss \leftarrow loss + loss^v$
    end
  end
  if $loss \geq loss_b$ **then**
    $s \leftarrow adj \times s$
  else
    $s \leftarrow \lfloor s/adj \rfloor$
  end
  $loss_b \leftarrow loss$
end

3.3 Experimental results for fashion style classification

In this section, we first introduce the implementation details of the proposed architecture, MVStyle. Second, the baselines adopted in the experiments are described, and the performance measurement results are reported for the K-fashion dataset. Then, we analyze whether MVStyle learns modality information well with example images. Finally, the effectiveness of the proposed cyclic oversampling is explained.

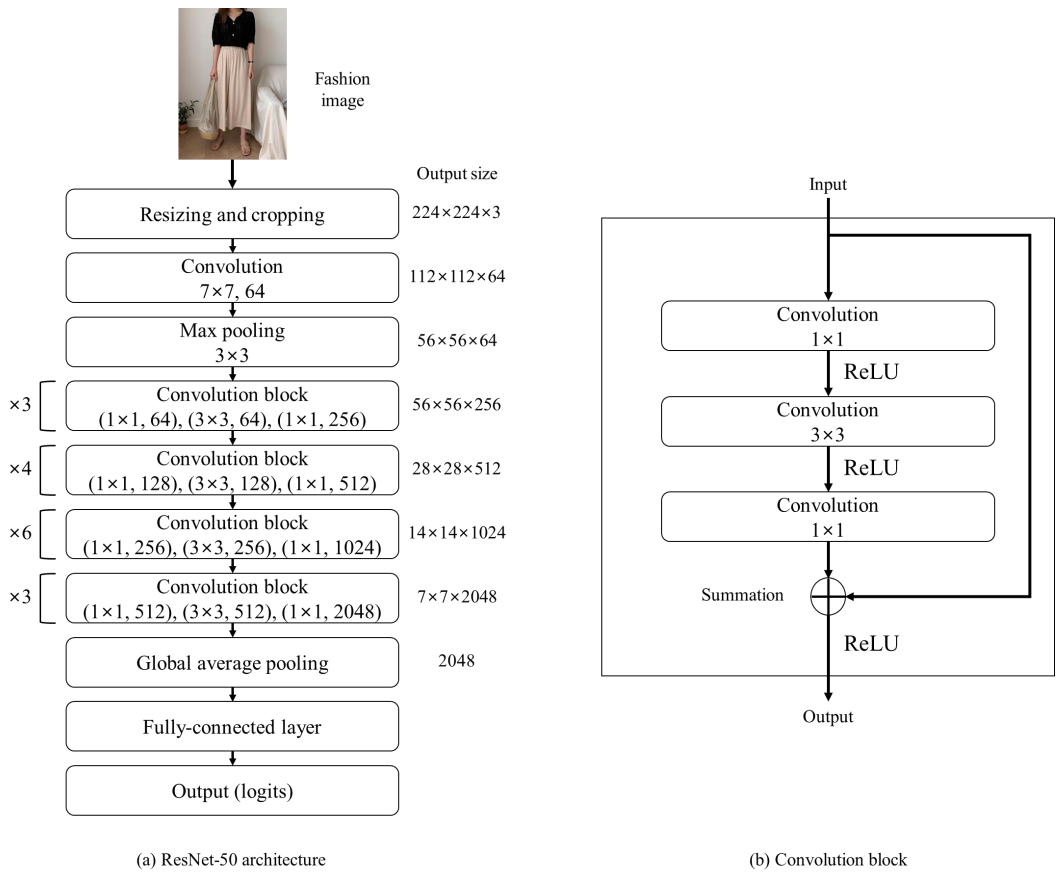
3.3.1 Implementation details

Preprocessing

Images and foreground images were resized to 256×256 pixels, and then the middle 224×224-pixel section was used, removing the images' outer part, which was highly unlikely to contain human body. These cropped images were represented with RGB channels and used as the inputs for the model. Here, the values corresponding to the RGB channels were normalized using the mean and standard deviation of the ImageNet [105] dataset for stable and fast model learning.

Network structure

For images and foreground images, ResNet-50 [106] is employed as the feature generator. ResNet-50 can extract image features based on a total of four convolution blocks, as shown in Figure 3.5 (a). The image generates 64 feature maps through convolution operation with a filter size of 7×7. The 64 feature maps, which are reduced in size through max pooling, are then passed through four convolution blocks. Each convolution block consists of convolution operations and the rectified linear unit (ReLU) [107] activation function, as shown in Figure 3.5 (b). The value calculated through the three convolution layers is added to the initial input value of the con-



(a) ResNet-50 architecture

(b) Convolution block

Figure 3.5: Illustration of ResNet-50 architecture.

volution block to generate the output value of the convolution block. This structure alleviates the gradient vanishing problem in which the gradient is close to 0, and the weights are not updated during the training. ResNet-50 utilizes a total of four types of convolutional blocks. Each convolutional block is repeated three, four, six, and three times to generate 256, 512, 1024, and 2048 feature maps, respectively. Finally, a 2048-dimensional vector is generated through global average pooling. Using this vector as the input value of the fully connected layer, the logit values corresponding to each class are calculated. ResNet-50 is pretrained with the ImageNet dataset and fine-tuned with our dataset to ensure that the feature generator's parameters capture fashion information well. Then, the ResNet-50 parameters are fixed, and ResNet-50 generates a 2048-dimensional feature vector from the output of the last convolution block using global average pooling.

All encoders and decoders of MVStyle are multilayer perceptrons with two hidden layers. Encoder layers for the images and foreground images each have 1560 hidden units, and layers for the attributes have 1460 hidden units. The 128-dimensional latent variable is reconstructed with the decoder having 1660 and 1160 hidden units for the images and foreground images and 660 and 460 hidden units for the attributes.

Optimization

CADA-VAE was trained using the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$ for 100 epochs. The initial learning rate was 1.5×10^{-4}. γ_1 and γ_2 were designed in a manner similar to [2]. Specifically, β increased from 1 epoch to 90 epochs at a rate of 5.62×10^{-6} per epoch, γ_1 increased from 21 epochs to 75 epochs at a rate of 8.15×10^{-4} per epoch, and γ_2 increased from 6 epochs to 37 epochs at a rate of 1.61×10^{-5} per epoch.

The classifier was trained for 10 epochs using stochastic gradient descent (SGD) with momentum parameter 0.9 and L2 regularization. The initial learning rate was 10^{-3} for the K-fashion dataset. For hyper-parameters of the cyclic oversampling, $adj = 2$, $s_{maj} = 1$, and $s_{min} = n_{maj}/n_{min}$ were used, where n_{maj} is the number of all image samples belonging to the majority classes, and n_{min} is the number of all image samples belonging to the minority classes.

Model selection

The classifier parameters to be employed for evaluation were selected based on the epoch with the best performance for each sampled K-fashion validation set.

3.3.2 Settings for experiments

Baselines

We employed baselines trained using only image modality with cross-entropy loss, focal loss [65], and label-distribution-aware margin (LDAM) loss [67] to compare the proposed model's performance on the K-fashion dataset. The model trained by the oversampling method was also adopted. The oversampling method was designed to sample the minority class s_{min} times and was trained through cross-entropy loss. In addition, two models were considered using multimodal input. One was the model trained by concatenating features of each modality (Concat). This model fed concatenated features, which are outputs of CADA-VAE's feature generators, to a classifier consisting of three fully connected layers. Then, it was backpropagated through LDAM loss to cope with the class imbalanced datasets. The other baseline was DRAGON [52], a state-of-the-art model in the GFSL task of the CUB [108] and SUN [109] datasets, which are single-label datasets. DRAGON can be trained using only images or images and other modalities simultaneously. Initially, DRAGON used only the attributes modality, but we experimented with attributes as well as foreground images. Similarly with [52], one fully connected layer was employed as the expert generating the classification probability from each modality input. The number of polynomial coefficients was selected as four.

Metrics

Top-3 accuracy (T3), which measures the frequency of all labels being included in the top-3 inference, was adopted as the metric on K-fashion with the multi-label

property. The formal definition of top-3 accuracy is as follows:

$$T3 = \frac{|\{\text{Samples with all labels in the top-3 inferences}\}|}{|\{\text{All samples}\}|} \tag{3.8}$$

The harmonic average between T3 for the majority classes and T3 for the minority classes (T3_H) was also considered, similar to the metric in [2]. If even one of the two labels was included in the minority classes, the sample was determined to belong to the minority classes.

3.3.3 Experimental results on K-fashion

Comparisons with the baselines

The results for each dataset are recorded in Table 3.2. Images, foreground images, and attributes are denoted by I, F, and A, respectively. The proposed model outperformed all baselines in the T3_H metric without significant performance degradation in the T3 metric, which is the total indicator of the dataset. The distribution of the K-fashion's test set has a long tail, as shown in Table 3.1. Therefore, the proportion of the majority increases in the T3 metric, whereas the proportion of the minority increases in the T3_H index. While the performance of the baselines is biased toward the majority or the minority, T3 and T3_H confirm that MVStyle maintained the majority's performance and outperformed the baselines for the minority.

Note that DRAGON is the model optimized for a single-label dataset. We modified and implemented the model to fit the multi-label situation, but the results show that the performance was biased toward the majority. We believe that DRAGON's reordering method of prediction scores is not suitable for the multi-label problem.

Ablation study

Table 3.3 shows the performances when the foreground images and attributes are combined with the images as the inputs. On average, the best performance was obtained when all modalities were used together as the inputs to the model. Additionally, it was found that when the foreground image was used together with the images, the performance was better than when the attribute was used together with the image. This suggests that the K-fashion dataset contains data that includes unnecessary information in the background, such as the examples of Feminine and

Table 3.2: Comparing MVStyle to the baselines on K-fashion. All numbers are percent accuracy, and the maximum value of each column is marked in bold.

Method	$\rho = 373.36$		$\rho = 232.72$		$\rho = 110.84$		$\rho = 73.91$		$\rho = 40.13$	
	T3	T3_H	T3	T3_H	T3	T3_H	T3	T3_H	T3	T3_H
CE	56.67	0.39	56.40	1.16	56.88	2.93	56.72	2.93	56.71	17.48
Focal loss	49.35	1.67	49.08	0.12	42.78	2.67	46.27	2.67	48.42	9.69
LDAM loss	39.80	8.61	41.51	9.59	43.56	20.07	31.87	20.07	44.26	27.10
Oversampling	46.83	29.02	49.10	28.88	45.77	33.12	46.91	33.12	46.46	24.75
Concat (I+F)	51.61	8.77	49.69	15.03	50.05	23.15	48.28	22.95	52.24	38.45
Concat (I+A)	54.90	22.13	54.89	24.23	54.66	29.86	53.68	36.35	53.36	40.92
Concat (I+F+A)	52.28	12.19	45.61	16.16	49.38	20.12	52.58	28.06	48.79	33.72
DRAGON (I)	56.43	0.07	56.46	0.39	56.64	0.33	55.66	1.48	56.16	5.54
DRAGON (I+F)	56.23	0.33	57.02	1.16	57.47	0.39	57.09	2.93	56.97	16.88
DRAGON (I+A)	**60.41**	0.26	**62.03**	0.97	**61.84**	0.97	**62.84**	3.06	**60.04**	17.29
MVStyle (I+F+A)	54.77	**33.65**	55.99	**38.55**	56.96	**39.55**	56.94	**41.01**	56.51	**44.43**

Genderless shown in Figure 3.1.

Table 3.3: Performances of MVStyle using various combinations of modalities. All numbers are percent accuracy, and the maximum value of each column is marked in bold. T3 avg and T3_H avg represent the average of T3 and T3_H for all sampled datasets, respectively.

Modality	$\rho = 373.36$		$\rho = 232.72$		$\rho = 110.84$		$\rho = 73.91$		$\rho = 40.13$		T3 avg	T3_H avg
	T3	T3_H	T3	T3_H	T3	T3_H	T3	T3_H	T3	T3_H		
I+F	**55.75**	**34.14**	55.83	38.24	56.78	38.89	56.87	39.55	55.76	**44.91**	56.20	39.15
I+A	55.32	33.1	55.15	34.3	56.33	37.99	55.89	40.03	56.41	44.63	55.82	38.01
I+F+A	54.77	33.65	**55.99**	**38.55**	**56.96**	**39.55**	**56.94**	**41.01**	**56.51**	44.43	**56.23**	**39.44**

3.3.4 Qualitative analysis

In this subsection, we visualize the cases correctly classified with additional modalities while wrongly identified without these modalities. For these experiments, we utilized the modality combinations of I+F and I+A with $\rho = 110.84$ on the K-fashion dataset. First, the cases that were correct for I+F only are presented in Figure 3.6. As previously mentioned, there are cases in which the clothing material is expanded and presented in K-fashion. CADA-VAE trained with the foreground images enables the model to focus on the areas containing the human body, not the areas where the material is enlarged. Next, examples that were correct only in I+A are shown in Figure 3.7. The left side of Figure 3.7 is an image with the label, Military. It is also annotated with Patchwork as one of the attributes; however, the corresponding part does not appear well on the image because of the angle. By using the attributes as a modality, the inference is corrected. The right side of Figure 3.7 is an image with the label, Sporty. If a person were wearing it, the tight shape of sportswear would appear on the image. However, the mannequin is wearing it, and the arm is loose. Since attributes such as Jersey and Tight were reflected in the image's latent distribution, this image could be classified as Sporty in the I+A combination model.

(a) Original Images (b) Foreground Images

Figure 3.6: Examples in which the generated foreground image captures human body and fashion items.

Figure 3.7: Examples that is judged as the correct answer when the attributes modality is included.

3.3.5 Effectiveness of the cyclic oversampling

To verify the effectiveness of the proposed cyclic oversampling, we compared our algorithm with various schedules that utilize the same oversampling ratio. The experiment was performed with the I+F+A modality combination, which produces the best performance on K-fashion's $\rho = 110.84$ dataset. Figure 3.8 shows the cyclic oversampling indicated by the blue line and other learning schedules.

The schedule for large amounts of oversampling in the first half and small amounts of oversampling in the second half is referred to as high to low (orange line). The opposite schedule is named low to high (green line). Increasing (red line) is a schedule that gradually increases the amount of oversampling, and decreasing (purple line) is a schedule that gradually decreases oversampling.

Table 3.4 shows the results of measuring the performance of the model trained

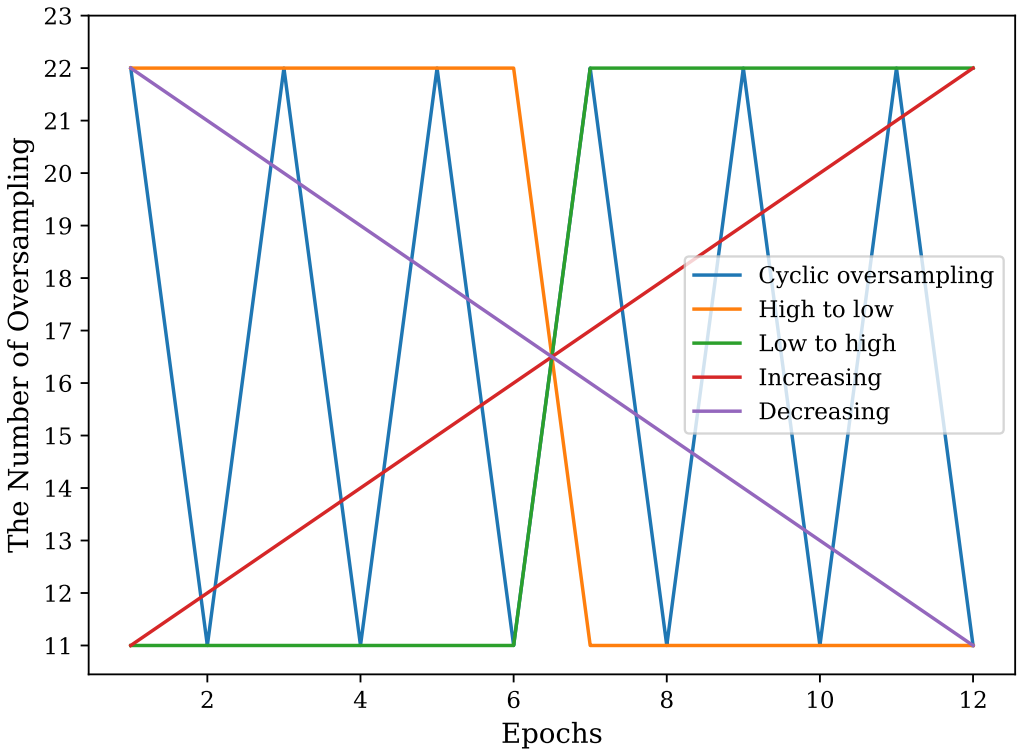


Figure 3.8: Sampling schedule of the cyclic oversampling and the comparisons. Each schedule method is drawn in a different color.

Table 3.4: Comparing the cyclic oversampling to the baselines on K-fashion $\rho = 110.84$. All numbers are T3_H represented in percent. The best performance is marked in bold.

Schedule	T3_H
High to low	28.49
Low to high	5.49
Increasing	5.55
Decreasing	27.57
Cyclic oversampling	**39.55**

with each schedule. The results of the model trained with the proposed cyclic oversampling achieved the best performance. The proposed algorithm continuously improves the performance of the minority class while reducing or increasing the frequency of sampling of the latent variable based on the validation loss of the minority class. In contrast, low to high or increasing schedule models fit the model for the majority at the beginning of the training, which was not overcome later. Additionally, for high to low or decreasing, the minority is sufficiently trained initially. Still, the frequency of oversampling for the minority gradually decreases, and the performance cannot be maintained, known as the catastrophic forgetting phenomenon [110].

Chapter 4

Motion Similarity Measurement

4.1 Datasets for motion similarity

Training the model to produce a motion embedding representation is performed with a synthetically created 3D motion dataset. Additionally, to demonstrate the generalization capabilities of the proposed model in evaluating the motion similarity of real-world data, we have manually annotated the NTU RGB+D 120 dataset. The latter is only used for performance evaluation.

4.1.1 Synthetic motion dataset: SARA dataset

We constructed a 3D motion dataset, named Synthetic Actors and Real Actions (SARA) to train a model to produce motion embeddings suitable for reasoning about motion similarity. Mixamo [5] was used for this.

Motion sequence data was generated by combining 18 different actors (i.e., action-performing characters). The characters were rendered in a skeleton shape with Adobe Fuse software. We selected four action categories (Combat, Adventure, Sport, and Dance), comprising several motion variations, where each action has a length of 32 frames or more. There are 4,428 base motions (e.g., dancing, jumping) in the SARA dataset. Intraclass variations were generated from these motions. Mixamo allows users to control various characteristics of each motion (e.g., **Energy**)

Table 4.1: SARA dataset overview.

Action category	Number of characters	Number of base motions	Number of variations
Combat	18	3,000	76,512
Adventure		264	3,390
Sport		306	4,485
Dance		858	18,756
Total	18	4,428	103,143

that can be adjusted to create variations for the dataset. Values of the characteristics variables are within the range of $[-1, 1]$, and in the SARA dataset, they are set to one of $\{-1, -0.5, 0, 0.5, 1\}$. This parameter is configured differently for each motion. Each sequence frame provides 3D coordinates of 17 joints from all body parts, and we generated samples through 2D projection. The dataset statistics are summarized in Table 4.1.

4.1.2 NTU RGB+D 120 similarity annotations

We collected motion similarity annotations for the NTU RGB+D 120 dataset to evaluate motion similarity in the real world. The NTU RGB+D 120 dataset is an action recognition dataset consisting of 114,480 videos covering 120 different actions of 106 people. While original videos from the dataset were used to obtain ground truth motion similarity from AMT, only the 2D skeleton sequences are utilized in our model to estimate the motion similarity.

Only a portion of the entire dataset was utilized because actions with small movements such as reading, writing, and talking on the phone also exist in the original NTU RGB+D 120 dataset. After filtering out these actions, 21 actions with large and well-defined movements were selected based on visual inspection. Then, two videos of 39 people for each action were sampled. The total number of sampled video clips was 1,638 (21 actions $\times$ 39 people $\times$ 2 videos).

We obtained motion similarity scores from humans through AMT [63] using the sampled videos. The motion similarity was scored on a 4-point scale ranging from 1 (utterly different motions) to 4 (the same movements) for each pair of video clips. The similarity score for a pair is an average of scores collected from at least 10 AMT workers. From all the possible candidates, the annotations for 20,093 randomly sampled video pairs were collected. We use all the annotations to evaluate the models, not to train them. These annotations are released on our project page. Some of the video pairs and the distribution of the annotated similarity scores are shown in Figure 4.1.

There are some imprecise skeleton data in NTU RGB+D 120. To cope with this problem and generate new 2D joint annotations, we used our reproduction of

MultiPoseNet [111] with an average precision of 0.709 for large objects in the COCO 2017 validation set to generate new 2D joint annotations.

More detailed information, including instructions, annotation guidelines provided to the workers, and 2D joint annotations generated using MultiPoseNet, can be found in Appendix A and Appendix B.

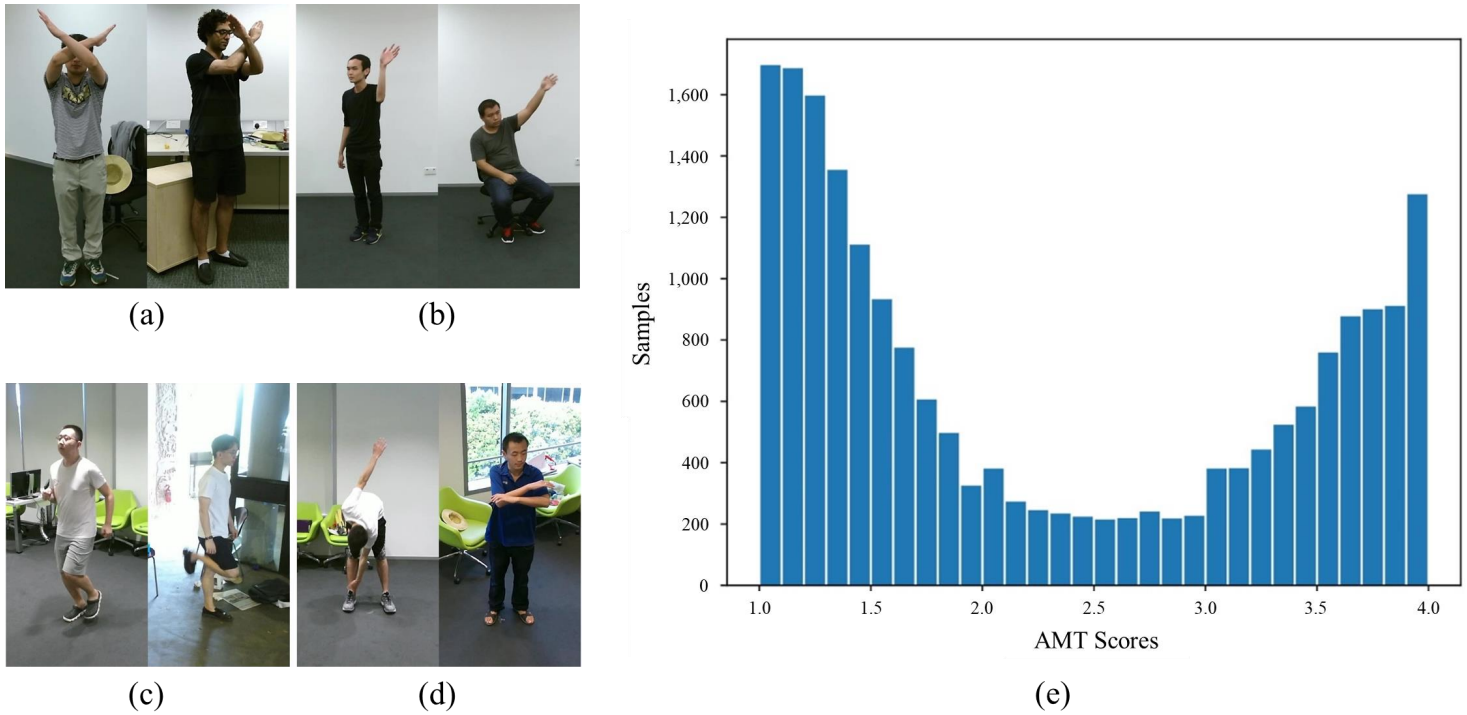


Figure 4.1: The examples of the AMT annotations pair from NTU RGB+D 120 [3] dataset. (a), (b), (c), and (d) are an example of scores 4, 3, 2, and 1 respectively; (e) is the histogram of the total collected scores.

4.2 Framework for measuring motion similarity

We propose a learning-based method to encode unique motion embeddings necessary for the motion similarity assessment. Inspired by the framework of [53], we extend the method by training a model to reconstruct each body part, rather than the whole body, to identify particular hand or foot movements. Furthermore, we propose a motion variation loss to calculate motion similarity robustly.

4.2.1 Body part embedding model

Network structure

Let $\mathcal{M}$, $\mathcal{S}$, and $\mathcal{C}$ denote the sets of motion, skeleton, and camera view attributes, respectively, in the training set. To calculate a total loss, we require $M = \{m, m', m''\}$, $S = \{s, s'\}$, and $C = \{c, c'\}$, which are subsets of $\mathcal{M}$, $\mathcal{S}$, and $\mathcal{C}$, respectively. In addition, m and m'' are required to be from the same motion class with different characteristics (i.e., motion variation), and m' to be a motion from a different class. For example, if m is a *Low jump*, then m'' is a *High jump* whereas m' is a *Sitting*. s and s' represent two skeletons with different body structures, and c and c' are viewing angles when the 3D motion is projected to 2D. We can generate a motion sequence by selecting and combining each element from M, S, and C. Let $\mathcal{X} = \{\mathbf{X}_{ijk} \in \mathbb{R}^{2\times J\times T} \mid i \in M, j \in S, k \in C\}$ be the set of 2D coordinate sequences where J is the number of joints of a skeleton, and T is the time length of the motion sequence. Among the elements of $\mathcal{X}$, $\mathbf{X}_{msc}$ and $\mathbf{X}_{ms'c}$ are the sequences of 2D joint coordinates representing the same motion m with the different skeletons s and s' at the same viewing angle c.

With set B, composed of $n_B = 5$ body parts, we decompose a skeleton to

construct body part embeddings (BPEs). Our case considers $B = \{Right\,Arm, Left\,Arm, Right\,Leg, Left\,Leg, Torso\}$, as depicted in Figure 4.2. Specifically, the motion sequence $\mathbf{X}_{msc}$ is decomposed into specific body parts $\mathbf{X}^b_{msc} \in \mathbb{R}^{2\times n_b\times T}$, where n_b is the number of joints in $b \in B$. X^b_{msc} is fed into body part motion encoder E^b_M and skeleton encoder E^b_S to produce embeddings. For global camera view encoder E_C, all the decomposed motion sequences are concatenated to generate the input of E_C. Let us denote this input as $\mathbf{X}^a_{msc} \in \mathbb{R}^{2\times n_c\times T}$, where n_c is the sum of the number of joints that make up each body part. The embeddings from the two types of encoders, E^b_M and E^b_S, respectively capturing the motion and skeleton features of body part b, are combined with the feature from the E_C. These combined features are decoded by body part decoder D^b to reconstruct the body part motion sequence. Since we are considering five body parts, each of the motion and skeleton encoders has five modules (i.e., one for each body part) that do not share weights with each other. This process is visualized in Figure 4.3.

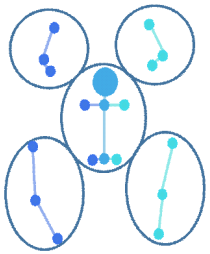
Losses

Aberman *et al.* [53] used a triplet loss to enforce separation between samples on the motion latent space. Let $\mathbf{z}^b_{msc} = E^b_M(\mathbf{X}^b_{msc})$ be the resulting motion embedding of $\mathbf{X}^b_{msc}$ obtained from E^b_M. Then, the motion triplet loss is:

$$\mathcal{L}^b_M(\mathbf{X}^b_{msc}, \mathbf{X}^b_{m's'c'}) = [d(\mathbf{z}^b_{ms'c'}, \mathbf{z}^b_{msc}) - d(\mathbf{z}^b_{ms'c'}, \mathbf{z}^b_{m's'c'}) + \delta]_+, \tag{4.1}$$

where $d(\cdot)$ is a distance metric, δ is the margin between the $\mathbf{X}^b_{msc}$ and $\mathbf{X}^b_{m's'c'}$ pair, and $[\cdot]_+$ is a hinge function [112]. The triplet loss ensures the distance between a reference and a positive sample is small and the distance between a reference and a negative sample is large. However, this loss measurement does not contain

Body parts decomposition



Body parts	Composition joints
Right Arm (RA)	Right shoulder, right elbow, right wrist
Left Arm (LA)	Left shoulder, left elbow, left wrist
Right Leg (RL)	Right hip, right knee, right ankle
Left Leg (LL)	Left hip, left knee, left ankle
Torso	Nose, neck, right shoulder, left shoulder, right hip, left hip

Figure 4.2: Body parts decomposition. Middle hip is the origin of the coordinate system.

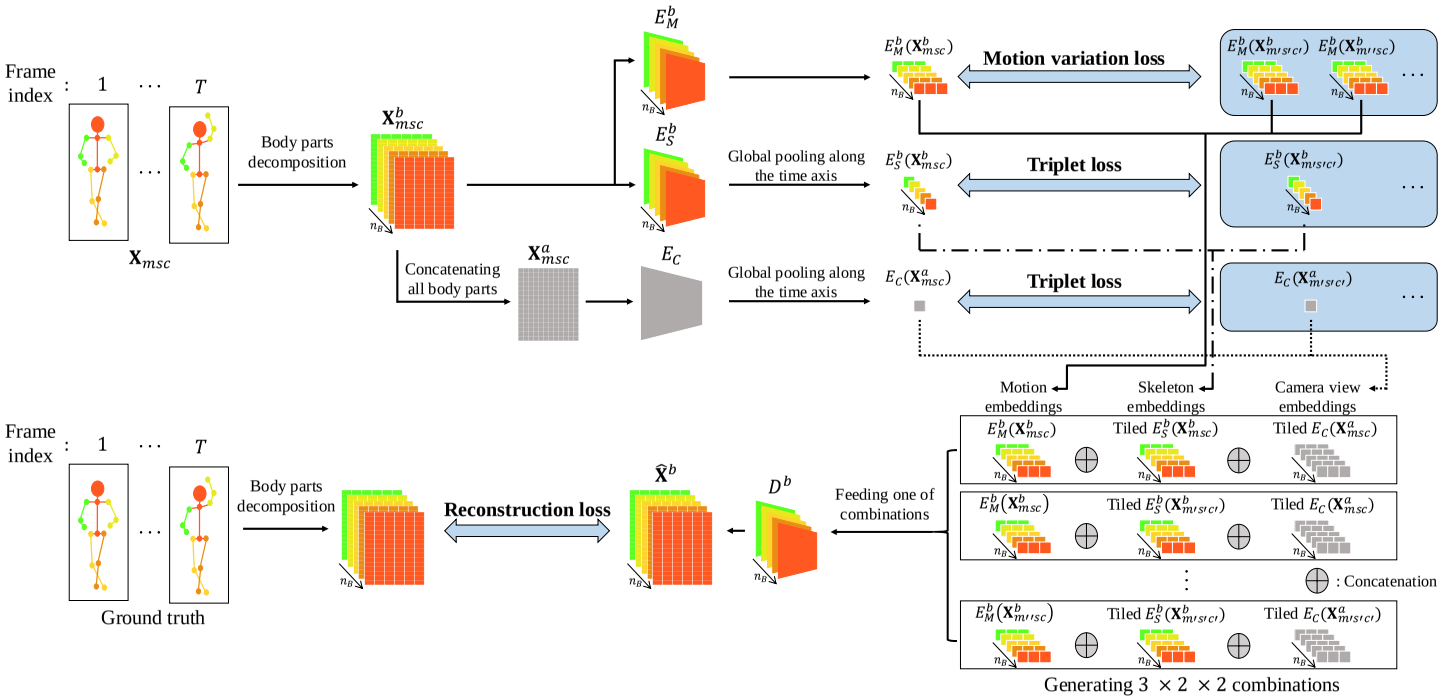


Figure 4.3: Visualization of the proposed model. Each body part is drawn in a different color.

information on the similarity of the samples.

To overcome this limitation of the triplet loss, we propose a loss term that utilizes a motion variation score between samples in the same action category. The proposed motion variation loss projects positive and semi-positive samples at a certain distance defined by the motion variation, as illustrated in Figure 4.4. Assuming that there are variables that can control the movement of the skeleton, such as **Energy**, **Distance**, and **Height** in Figure 4.5, we let $\mathbf{v}_m$ be the characteristic vector that has each element corresponding to one of these variables for motion m. Since m and m'' belong to the same motion class, $\mathbf{v}_m$ and $\mathbf{v}_{m''}$ have the same $n_{\mathbf{v}_m}$ number of variables. Then, the motion variation $var(m, m'')$ between m and m'' is defined as:

$$var(m, m'') = \frac{\|\mathbf{v}_m - \mathbf{v}_{m''}\|_1}{2 \times n_{\mathbf{v}_m}}. \tag{4.2}$$

The motion variation loss $\mathcal{L}^b_{var}$ is defined using the motion variation as:

$$\begin{aligned}\mathcal{L}^b_{var}(\mathbf{X}^b_{msc}, \mathbf{X}^b_{m's'c'}, \mathbf{X}^b_{m''sc}) = \mathcal{L}^b_M(\mathbf{X}^b_{msc}, \mathbf{X}^b_{m's'c'}) + \mathcal{L}^b_M(\mathbf{X}^b_{m''sc}, \mathbf{X}^b_{m's'c'}) \\ + \alpha_1\{d(\mathbf{z}^b_{msc}, \mathbf{z}^b_{m''sc}) - \alpha_2 \cdot var(m, m'')\}^2, \end{aligned} \tag{4.3}$$

where $d(\cdot)$ is a distance metric, and hyper-parameters α_1 and α_2 are respectively selected to 1 and 0.1 by grid search in our experiments. With this loss term, we expect the motion embedding vectors of positive and semi-positive samples to be dependent on the characteristic vector.

For the skeleton and camera view embeddings, triplet losses $\mathcal{L}^b_S$ and $\mathcal{L}_C$ can be obtained in the same manner as (4.1). These terms are then combined to complete

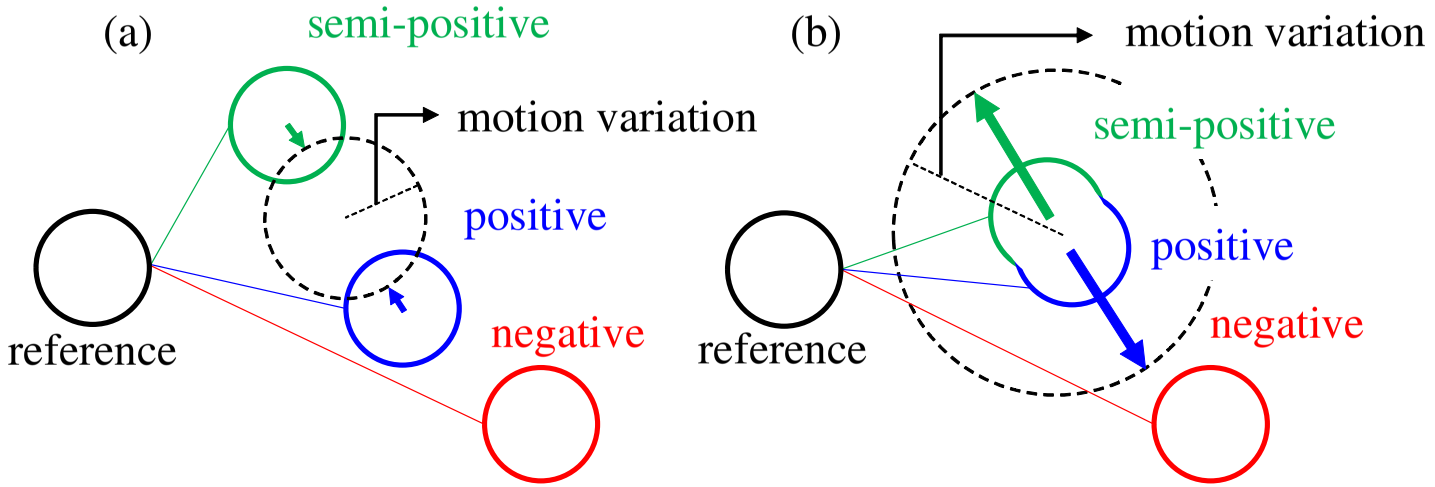


Figure 4.4: Visualization of the motion variation loss. (a) shows a situation where loss brings positive and semi-positive samples closer when they are mapped far; (b) indicates that the loss drives them far when they are mapped close.

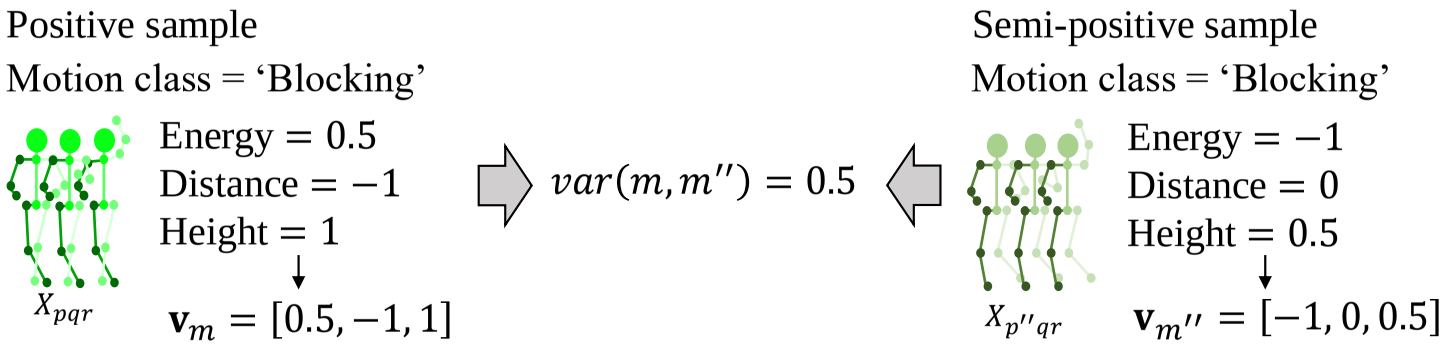


Figure 4.5: Visualization of the motion variation for positive and semi-positive samples of the SARA dataset. The motion variation is computed from two samples that belong to the same motion class but have different characteristics (e.g., **Energy**).

the final similarity loss term:

$$\mathcal{L}_{sim} = \sum_{b \in B} \mathcal{L}^b_{var} + \sum_{b \in B} \mathcal{L}^b_S + \mathcal{L}_C. \tag{4.4}$$

The estimate $\hat{\mathbf{X}}^b \in \mathbb{R}^{2 \times n_b \times T}$, which is the output of the D^b, can be obtained by providing the concatenation of motion, skeleton, and camera view embedding vectors to D^b. While the BPE model can accommodate 12 combinations specified by the different attributes of M, S, and C as inputs, only three inputs, $\mathbf{X}^b_{msc}$, $\mathbf{X}^b_{m's'c'}$, and $\mathbf{X}^b_{m''sc}$, are utilized to calculate the reconstruction error for computational efficiency. Specifically, motion embeddings, $E^b_M(\mathbf{X}^b_{msc})$, $E^b_M(\mathbf{X}^b_{m's'c'})$ and $E^b_M(\mathbf{X}^b_{m''sc})$; skeleton embeddings, $E^b_S(\mathbf{X}^b_{msc})$ and $E^b_S(\mathbf{X}^b_{m's'c'})$; and camera view embeddings, $E_C(\mathbf{X}^a_{msc})$ and $E_C(\mathbf{X}^a_{m's'c'})$ are concatenated into 12 different ways to build the inputs of D^b. Before the concatenation, the camera view embedding is copied a number of times equal to the number of body parts. Then, the skeleton embeddings and all the copied camera view embeddings are tiled along the time axis. The reconstruction error can then be calculated by comparing the output $\hat{\mathbf{X}}^b$ of the decoder D^b with the ground truth. The reconstruction error for each body part is defined as follows:

$$\mathcal{L}^b_{rec} = \frac{1}{12} \sum_{i \in M} \sum_{j \in S} \sum_{k \in C} (\hat{\mathbf{X}}^b_{ijk} - \mathbf{X}^b_{ijk})^2. \tag{4.5}$$

This reconstruction error term helps disentangle the motion, skeleton, and camera view embedding vectors.

Finally, the foot velocity loss $\mathcal{L}_f$ used in [53] is applied to prevent a foot skating phenomenon that causes a significant error in hands and feet. The final loss is the

weighted sum of the individual loss terms:

$$\mathcal{L} = \lambda_1 \sum_{b \in B} \mathcal{L}_{rec}^{b} + \lambda_2 \mathcal{L}_{sim} + \lambda_3 \mathcal{L}_{f}, \tag{4.6}$$

where the weights λ_1, λ_2, and λ_3 are respectively selected to 1, 1, and 0.5 by grid search in our experiments.

4.2.2 Measuring motion similarity

The measurement of similarity between motions is described in Algorithm 2. We use the outputs of the motion encoders only so that the model can generate predictions robust to differences in view-points or skeletons (e.g., different heights).

Let two sequences $\mathbf{X}_1 \in \mathbb{R}^{2\times J\times T_1}$ and $\mathbf{X}_2 \in \mathbb{R}^{2\times J\times T_2}$, which are targets for measuring motion similarity, have time lengths of T_1 and T_2, respectively. Furthermore, let $\mathbf{X}_1^b \in \mathbb{R}^{2\times n_b\times T_1}$ and $\mathbf{X}_2^b \in \mathbb{R}^{2\times n_b\times T_2}$, respectively, be the sequences corresponding to body part b of $\mathbf{X}_1$ and $\mathbf{X}_2$. A sliding window approach with window size w and stride r is applied to split $\mathbf{X}_1^b$ and $\mathbf{X}_2^b$ into patches, which are then put into the motion encoder E_M^b. Let $\mathcal{F}_1^b$ and $\mathcal{F}_2^b$ denote the sets of motion embeddings of the $\mathbf{X}_1^b$ and $\mathbf{X}_2^b$ patches, respectively. They are then fed as inputs to the DTW algorithm to determine the best alignment between the sequences. Subsequently, the similarity of each body part can be obtained through average cosine similarity between matching time frames on the DTW path. Based on each body part's similarity, the final similarity of the two sequences $\mathbf{X}_1$ and $\mathbf{X}_2$ can be calculated by averaging over the body parts and temporal timestamps.

Algorithm 2: Measuring motion similarity

Input : motion sequences $\mathbf{X}_1, \mathbf{X}_2$,
motion encoders
$E_M = \{E_M^1, \cdots, E_M^b, \cdots, E_M^{n_B}\}$,
video sampling window size w,
video sampling stride r

Output: similarity sim

divide $\mathbf{X}_1, \mathbf{X}_2$ into each body part $\{\mathbf{X}_1^1, \cdots, \mathbf{X}_1^b, \cdots, \mathbf{X}_1^{n_B}\}$, $\{\mathbf{X}_2^1, \cdots, \mathbf{X}_2^b, \cdots, \mathbf{X}_2^{n_B}\}$

for $b = 1$ **to** n_B **do**

- extract patches from $\mathbf{X}_1^b, \mathbf{X}_2^b$ using sliding window with w, r
- obtain embeddings $\mathcal{F}_1^b, \mathcal{F}_2^b$ from the extracted patches by using E_M
- $path \leftarrow DTW(\mathcal{F}_1^b, \mathcal{F}_2^b)$
- $sim^b \leftarrow$ average cosine similarity between the embedding pairs in $path$

$sim \leftarrow$ the average of $\{sim^b | b \in B\}$

4.3 Experimental results for measuring motion similarity

In this section, we first present implementation details for our model, then introduce the correlation measurements between the collected annotations for NTU RGB+D 120 pairs and the similarities produced by several models, including ours and the baselines. Next, we visualize the motion latent space of our model. Finally, we explain how our framework can be applied to real-world tasks. For all the experiments in this section, only the SARA dataset is used for training, and the NTU RGB+D similarity annotations are used for evaluation.

4.3.1 Implementation details

Preprocessing

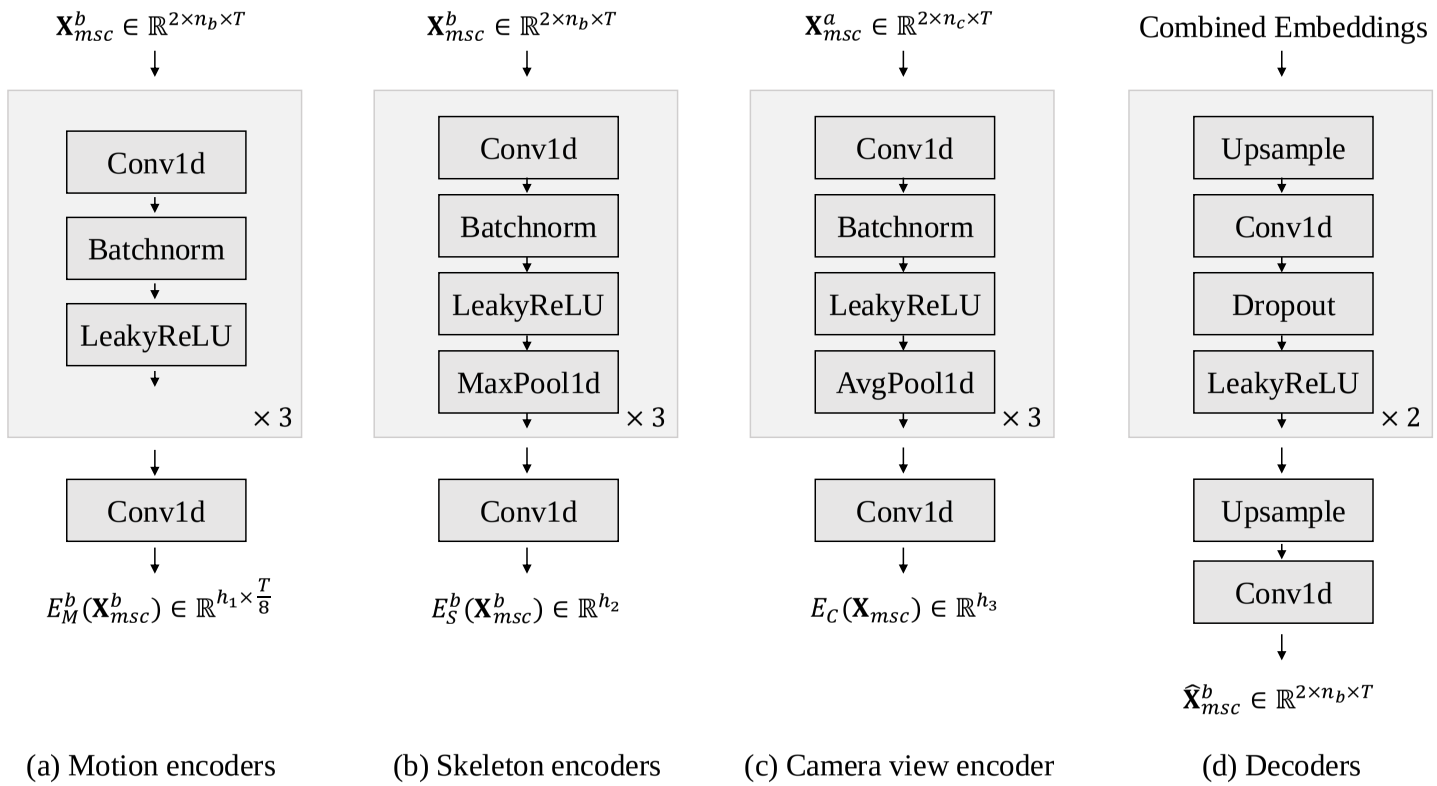
First, all motion sequences are divided into segments of 32 frames. Then, we split the SARA dataset into training and validation sets composed of different base motions of non-overlapping characters. This results in 455,028 motions, each with 32 frames,

from 12 characters for training and 64,218 motions, each with 32 frames, from 6 characters for validation.

In a real-world environment, the size of a person's projection varies depending on the distance from the camera. To address this problem, the skeleton size of the sequence is reduced or increased by a scale factor, which is randomly sampled between 0.5 and 1.5. After the scale adjustment, the reference joint for each body part is selected, and the coordinates of all joints are changed from absolute to relative coordinates. The reference joints for each body part are shoulders for arms, hips for legs, and the middle hip for a torso. Finally, the coordinates are normalized to produce the final input.

Network structure

The encoders and decoders in our BPE model depicted in Figure 4.6 are implemented as convolutional layers with a batch normalization [113] layer and a leaky rectified linear unit (Leaky ReLU) [114] activation function in between each layer. The motion encoder for each body part takes a 2D sequence for the corresponding body part as input. Let $\mathbf{X}^b_{msc} \in \mathbb{R}^{2\times n_b \times T}$ be a sequence fed into the encoder for body part b. Each motion encoder generates the embedding, denoted as $E^b_M(\mathbf{X}^b_{msc}) \in \mathbb{R}^{h_1 \times \frac{T}{8}}$, where $h_1 = 128$ for the torso motion encoder and $h_1 = 64$ for the other encoders. The torso embedding is set to a higher dimension because its number of joints is greater than the number of joints for other body parts. In the case of the skeleton encoders, the input is the same as the motion encoders. The difference is that it generates an embedding that compresses the temporal information using global max pooling. This embedding, denoted as $E^b_S(\mathbf{X}^b_{msc}) \in \mathbb{R}^{h_2}$, has dimension $h_2 = 32$ for torso and $h_2 = 16$ for the other body parts. The camera view encoder uses the concatenation



(a) Motion encoders (b) Skeleton encoders (c) Camera view encoder (d) Decoders

Figure 4.6: Network structure of encoders and decoders. Except for the camera view encoder, encoders and decoders are equal in number to body parts.

of the body parts as input. Unlike the skeleton encoder, we use average pooling to make the embedding with dimension $h_3 = 64$. The camera view embedding is copied a number of times equal to the number of body parts. Then, the generated skeleton and the camera view embeddings are tiled along the time axis to match the size of the motion embedding, $\frac{T}{8}$, and subsequently concatenated. The decoder then yields an estimate $\hat{\mathbf{X}}^b_{msc}$ for each body part using the concatenated embeddings. The size of the resulting output is the same as the input size of the encoders.

Optimization

The model was trained using Adam optimizer [115] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$. $L2$ regularization with a weight decay of 0.01 was also used to prevent overfitting. The initial learning rate was 10^{-3}, and we applied an exponential decay with a rate of 0.98 every 1/3 epoch. Using a single GPU (NVIDIA Tesla V100) and Intel Xeon 5120 @ 2.20 GHz, it took less than 20 minutes for the model to train one epoch with 12 workers and a batch size of 2,048.

Model selection

As previously mentioned, we trained the model only with the SARA dataset. The model parameters to be employed for evaluation were selected based on the epoch with the lowest total loss for the SARA validation set.

4.3.2 Experimental results on NTU RGB+D 120 similarity annotations

Comparisons with the baselines

We calculated the correlations between the model's predictions and the annotated similarity scores to determine how comparable the model is to human perception. Spearman's rank correlation was employed as an evaluation metric, and $20,093$ pairs were used to obtain the correlations. For the models relying on 2D coordinates, the symmetrical nature of the human body was utilized. In detail, each model predicted two different similarity scores for each motion pair: one from the original motion sequences and the other by using horizontally flipped ones from the two sequences, and the larger value was selected as the final similarity prediction. This procedure is referred to as *Body flip* in Tables 4.2 and 4.3.

Four approaches were considered as baselines. The first was a heuristic algorithm that calculated the Euclidean distance between the joints of the matching frames. DTW was used to align the frames of two motion sequences. As the second baseline, we used the algorithm of [40], where the authors proposed the similarity of 3D motion sequences between teacher and learner in a dance teaching situation. We used the 3D joint coordinates of the NTU RGB+D 120 as inputs for this method.

For the third baseline, we considered the approach of Coskun *et al.* [35], which was used to carry out the action recognition and retrieval tasks, since it learns the similarity between the motions through metric learning. We re-implemented the model of [35] and trained it with the SARA dataset for a fair comparison. Similar to Algorithm 2, DTW was employed to align the motion embedding patches. Finally, we trained the model of Aberman *et al.* [53] on the SARA dataset and used its motion

Table 4.2: Rank correlations with AMT scores.

Joints annotations	NTU RGB+D 120 [3]		Pose estimated joints	
Method	Original pair	Body flip	Original pair	Body flip
Kim and Kim [40]	0.1692	0.1601		
Joint distance	0.2222	0.1721	0.2634	0.1948
Coskun *et al.* [35]	0.2252	0.2335	0.2845	0.2996
Aberman *et al.* [53]	0.3207	0.3341	0.3545	0.3911
BPE with positional encoding	0.3972	0.4186	0.5032	0.5217
BPE (ours)	0.4345	0.4609	**0.5509**	**0.5970**

Table 4.3: Ablation study on the proposed loss function.

Method	With variation	With recons.	Original pair	Body flip
BPE (ours)	X	X	0.5264	0.5662
	X	O	0.5280	0.5740
	O	X	0.5363	0.5758
	O	O	**0.5509**	**0.5970**

embeddings for similarity measurement. Since [53] generated one motion embedding for the entire body's motion sequence, Algorithm 2 was modified to calculate the similarity for individual embedding vector.

Meanwhile, the BPE model was implemented with positional encoding attached to $\mathbf{X}^b_{msc}$. Positional encoding is an embedding that can include the positional information of a sequence, as proposed in Transformer [116]. We trained the BPE model by generating the model inputs, $(\mathbf{X}^b_{msc})_{pos} \in \mathbb{R}^{2\times(n_b+h_{pos})\times T}$, where h_{pos} is the dimension of the positional encoding vector, and we compared its performance with the other models.

Overall, the highest correlation results were achieved by the proposed BPE model, as shown in Table 4.2. Our method significantly improved the correlation results between the similarity estimation and human perception. There are two main reasons for this. One is that our method can estimate the similarity score for each body part. When people compare two motions, they tend to think that the whole body performs different motions even if only one body part moves differently. Furthermore, our loss term allows the model to catch subtle intraclass variations and enables the similarity estimation to be closer to human perception. The ablation study related to the loss term will be discussed.

Interestingly, in all the cases based on the proposed BPE model, the similarity correlation results produced with the *Body flip* had the best performance. This implies that horizontally flipped motions are considered the same motion in the human perspective. For example, people may not care whether a human throws a ball with their right hand and tend to focus only on the fact that a ball is thrown to determine whether the motions are similar.

Finally, we noted that using motion sequences corrected by MultiPoseNet results in a higher correlation score for every method. We believe that refining imprecisely annotated poses impacted this.

Ablation study

To verify the effectiveness of the proposed loss term, we carried out ablation experiments. Specifically, we separately removed the reconstruction and motion variation losses by excluding their contributions from the total loss. The results are outlined in Table 4.3.

The results show that the correlation scores increased when motion variation loss was applied. Unlike the triplet loss, motion variation loss forces the model to ensure that motion embeddings are separated even for slightly different motions from the same class. We argue that this property helps a model to generate similarity predictions close to human perception.

When we omitted the reconstruction loss, the correlation scores decreased. We claim that the reconstruction loss of our model forces the embedding to contain essential information of the motions, and when it is applied with a cross-reconstruction scheme, it can generate the embedding of the motion attribute independent of the skeleton or camera view.

The BPE model without the motion variation loss (shown in the second row of Table 4.3) performed better than Aberman *et al.* (shown in the fourth row of Table 4.2), suggesting the effectiveness of the proposed body part decomposition approach. The motion embedding for each body part appears to make it possible for the model to capture detailed motion information.

Motion similarity comparison by body part

Our model computed the motion similarity for each body part for a given pair of motions. Representative results from NTU RGB+D 120 are given in Table 4.4, with the corresponding visual references in Figure 4.7. Figure 4.7 (a) represents a case where both people raise their left hand. Our model predicted high similarity results in most body parts except the right hand for which their positions were different. Next, Figure 4.7 (b) shows a motion where one person raises both hands while the other raises a left hand. The model predicted a lower similarity score for the right arm and a high score for the remaining parts. Figure 4.7 (c) represents motion sequences with the same waving motion performed with a different hand. It was found that the motion similarities for both arms were lower than for the other body parts. In Figure 4.7 (d), the left person sits on the chair, and the right one performs squats. The model predicted lower similarities of the legs and torso than those of arms because the angles of the knees and hips were different. Finally, an example of a comparison between a person standing with raised arms and a person sitting is displayed in Figure 4.7 (e). The similarity scores in all parts were relatively low as the motions of all body parts' are different.

Figure 4.7: Pairs (from NTU RGB+D 120 [3]) for body part similarity in Table 4.4.

Table 4.4: Motion similarity by body part for the sample pairs in Figure 4.7. Body parts with relatively lower similarity scores are marked bold.

	Right Arm	Left Arm	Right Leg	Left Leg	Torso
(a)	**-0.0153**	0.6982	0.9297	0.9205	0.9429
(b)	**0.1740**	0.7338	0.8843	0.6937	0.8773
(c)	**0.1841**	**0.0648**	0.9423	0.9321	0.9160
(d)	0.8196	0.7210	**0.3999**	**0.5499**	**0.2366**
(e)	**0.2147**	**-0.2170**	**0.1040**	**0.1517**	**0.1057**

4.3.3 Visualization of motion latent clusters

The motion latent spaces for the SARA validation set and NTU RGD+D 120 are shown in Figure 4.8, visualized using t-distributed stochastic neighborhood embedding (t-SNE) [117]. t-SNE represents high-dimensional data as a two-dimensional graph by learning a two-dimensional embedding vector to preserve the neighbor structure between high-dimensional vectors.

Figure 4.8 (a) shows that despite the differences in the characters and camera views in the SARA dataset, the sequences with the same motion attributes are clustered together. It supports the claim that similarity can be measured by considering only the motion, independent of the humans or camera views. Furthermore, closely mapped motions corresponded to similar body part movements, as displayed on the right side of Figure 4.8 (a), even though they belonged to different classes. The plot on the left of Figure 4.8 (b) shows the motion latent space for the 21 sampled actions of NTU RGB+D 120. Overall, the samples with the same action class formed clusters. In some cases, however, different motions were mapped closely, similar to the aspect shown in the SARA dataset's case. The photos on the right of Figure 4.8 (b) are examples of such cases in which they are positioned closely due to their motion

classes' similarities from the human perspective.

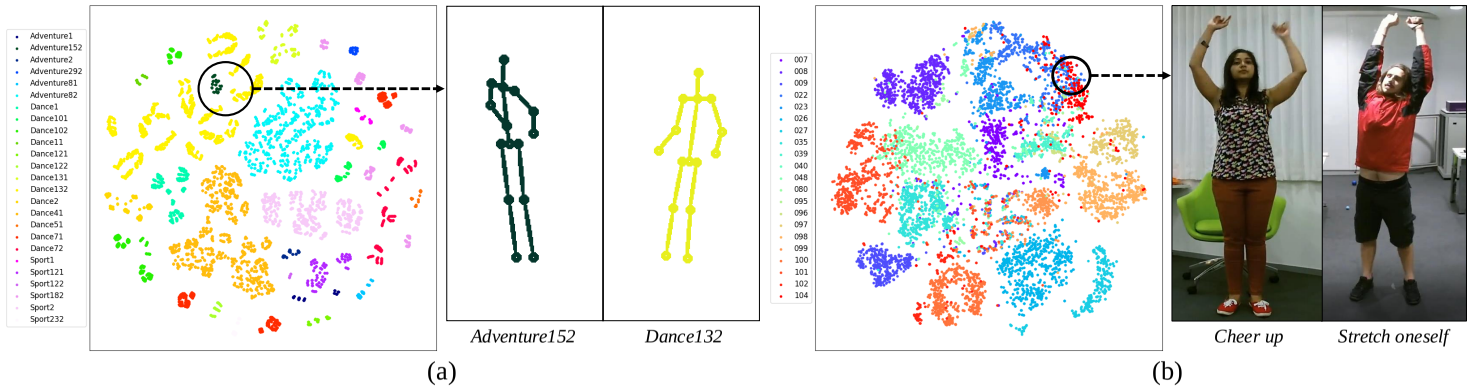


Figure 4.8: Visualization of motion latent vectors. The motion classes of the SARA validation set are clustered by colors in the left part of (a). The dark green (*Adventure152*) and light yellow (*Dance132*), circled in black, correspond to the similar motions that were performed while standing with the elbows bent and leaning back (shown in the right part of (a)). The visualization of 21 sampled actions of NTU RGB+D 120 [3] is made in the left part of (b). The blue (*cheer up*) and red (*stretch oneself*) positioned on the upper right, represent similar motions (shown in the right part of (b)).

4.4 Application

In this section, we provide a guideline for using the proposed motion similarity in the real-world. Evaluating an exercise (e.g., dance, yoga, and figure skating) performance is a natural application of the proposed model.

4.4.1 Real-world application with dancing videos

To evaluate the proposed motion similarity measurement framework with real-world data, we collected dancing videos from Korean idol audition programs. This dataset fits our goal to assess the performance periodically as the dance progresses and as the audition participants perform the same dance except for the section where they show their dancing individuality.

We compared two temporally aligned videos of people trying to perform the same dance. We used the method of [111] to extract the joints' location, although any human pose estimation algorithm is suitable. Algorithm 2 with window size $w = 32$ and stride $r = 32$ was used to obtain the motion similarities between two sequences. The parameters were set to provide feedback approximately every second. However, they could be defined arbitrarily based on the application or user preference.

To compare two video clips (3.5 min long, 24 fps), the proposed method took about 7.8 s (approximately 670 fps) on an Intel Xeon 5120 CPU @ 2.20 GHz without model or code optimization. This included joint data preprocessing, network inference, and motion similarity calculation but excluded the pose estimation extraction.

Figures 4.9 and 4.10 are examples that show interactive feedback of two similar dance motions. Since the two audition participants performed similar dance moves, the similarity was measured with high scores in all body parts. Figures 4.11 and

Figure 4.9: Illustration example comparing two similar dance sequences by body part. A threshold of 0.4 was chosen to separate similar (green) motions.

Figure 4.10: Illustration example comparing two similar dance sequences by body part. A threshold of 0.4 was chosen to separate similar (green) motions.

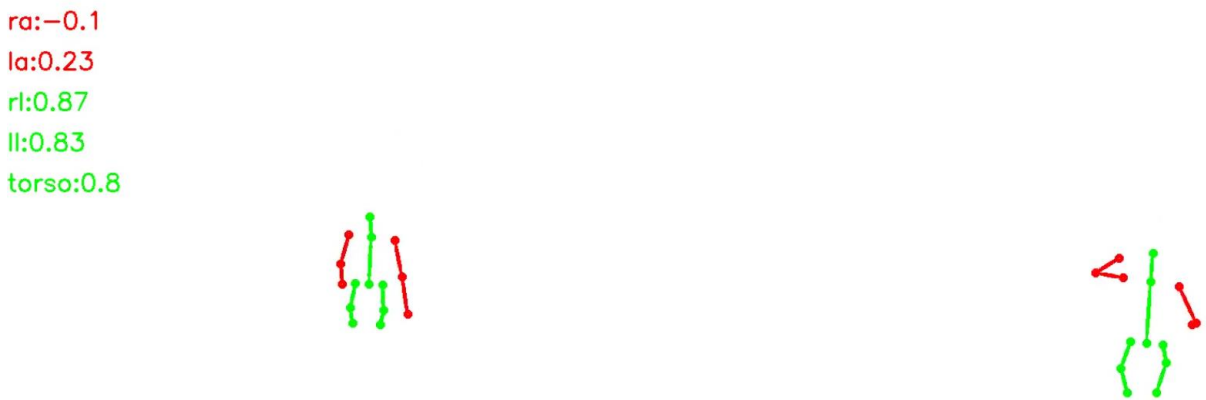


Figure 4.11: Illustration example comparing two dissimilar dance sequences by body part. A threshold of 0.4 was chosen to separate similar (green) and different (red) motions.

Figure 4.12: Illustration example comparing two dissimilar dance sequences by body part. A threshold of 0.4 was chosen to separate similar (green) and different (red) motions.

4.12 are examples that show interactive feedback of two partially different dance motions. When two audition participants momentarily showed different body part movements, the measured similarity score was lower.

Note that, in its current form, the proposed method neither provides feedback on the exact incorrect human joint location nor how to correct the location to make the action more similar. Nonetheless, our approach provides a similarity score on the sequences without requiring evaluation of the motion similarity based on manually defined rules, such as normalized distances between joints of an actor and the ground truth.

Meanwhile, the audition judges also evaluated and graded the participants based on the dance and the corresponding song in the audition program. We clustered motion embeddings through k-means clustering algorithm [118, 119] to check whether participants with similar dancing skills can form clusters. Since the dance and the song were evaluated comprehensively, we cannot directly compare the evaluation results of the judges with our framework, which generates motion embeddings with only dance movements. However, it is possible to compare the 16 highest-grade participants (grade 1) who were thought to have been relatively good at both dancing and singing, and the 14 lowest-grade participants (grade 5) who were considered to be relatively poor at both dancing and singing. As a result of clustering by the dance motion embeddings of the chorus part, one cluster consisted of 10 participants with an average grade of 1.4, and the other cluster consisted of 20 participants with an average grade of 3.6. This suggests that the highest-grade participants who performed dance movements accurately and produced similar motion embeddings formed the cluster with an average grade of 1.4.

4.4.2 Tuning similarity scores to match human perception

For a real-world application, it is necessary to know the meaning of the similarity measured from two motions. For example, if the measured similarity is 0.7, the number 0.7 should be able to imply how similar the two motions are. To convert the measured similarity into meaningful numbers, we utilized the similarity scores of the NTU RGB+D 120 similarity annotations which is collected from humans. The relationship between the score of NTU RGB+D 120 similarity annotation and the average value of the similarity measured from the proposed framework for each score is shown in the Figure 4.13. Since this relationship is approximated by the logistic function $f(x) = \frac{1}{1+e^{-(27x-15.5)}}$, the measured similarity can be converted to the criteria of NTU RGB+D 120 similarity annotation using this logistic function. The examples of each score on 4-point scale ranging from 1 (utterly different motions) to 4 (the same movements) is shown in Figure 4.1.

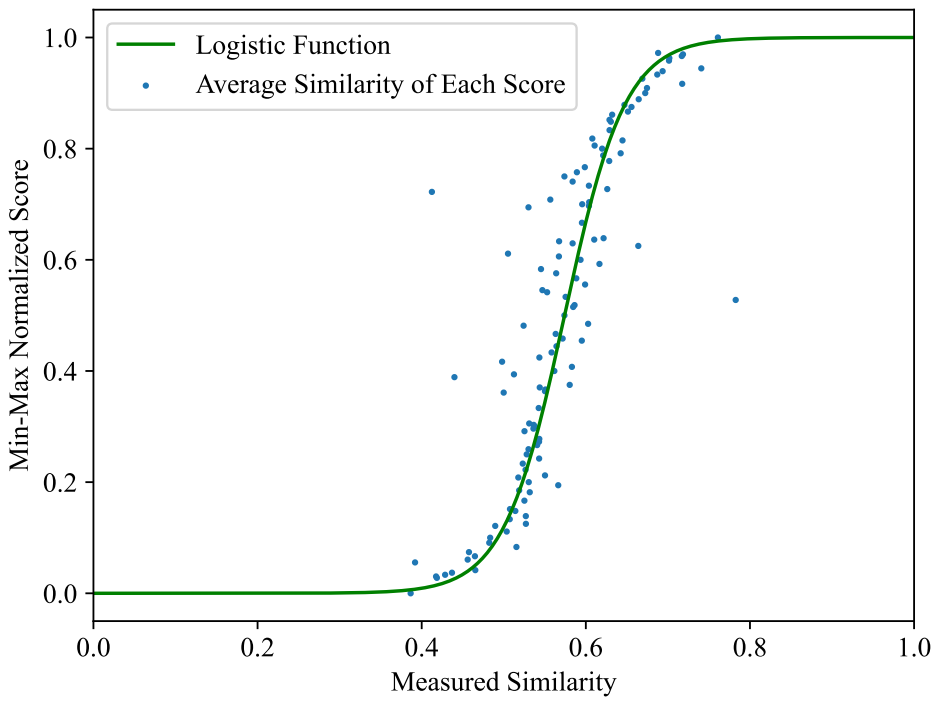


Figure 4.13: Relationship between the similarity scores of NTU RGB+D 120 similarity annotations and the similarity measured from the proposed framework. The x-axis represents the similarity measured from the proposed framework, and the y-axis represents the values obtained by normalizing the similarity scores of 1-4 points collected from humans to a 0-1 scale. The relationship can be approximated by the logistic function.

Chapter 5

Conclusions

5.1 Summary and contributions

This thesis proposes a method to improve the performance of two computer vision tasks, fashion style classification and measuring motion similarity using the visual information of the human body based on deep autoencoder architectures.

For the fashion style classification task, we proposed an architecture to classify the fashion styles by training the foreground images, which were cropped to include only the human parts with fashion items, and the fashion attributes represented in each image. The model was trained on GFSL problem settings, in which the class imbalance scenario is suitable for the fashion style classification domain. The CADA-VAE structure is adopted to overcome the class imbalance and utilize other modalities. After training CADA-VAE, the style classifier was trained through the proposed cyclic oversampling. The results showed that the proposed architecture, MVStyle, outperformed the baselines.

For motion similarity measurement utilized in various human-related computer vision tasks such as action recognition, human performance evaluation, and anomaly detection, we generated motion embedding vectors for five body parts, and a motion variation loss term was introduced to distinguish similar motions. Additionally,

a synthetic dataset to train the model was constructed. For evaluation purposes, we collected real-world annotations of the NTU RGB+D 120 dataset. The evaluation indicated that our method outperformed all the baseline models considered. An example application in which the proposed framework can be utilized by measuring the similarity of dance motions was also presented using the joint-coordinate sequences measured from real-world dance videos.

5.2 Limitations and future research

Training the model using additional modalities will be our future work for the fashion style classification task. Foreground images will be able to be specified, such as a top, a bottom, and accessories. Human pose, which was used to measure motion similarity, will help the model to recognize the clothing shape accurately. It is also possible that attributes can be represented in a format other than a one-hot vector. If the soft grouping approach proposed in [120] is applied, even information not included in the attribute labels can be reflected in learning. Because this study was conducted on women's fashion images, its performance was not confirmed on men's fashion images. Furthermore, it was trained using shopping mall images, not images from real life. Therefore, it will be our additional future work to train the model with different fashion style datasets.

Because our approach for measuring motion similarity depends on the joint-coordinate sequences, the similarity model performs best when precise pose estimation is available. However, the pose estimation may not be satisfactory in challenging situations (e.g., occlusions and crowded scenes), and our future work will seek to measure the similarity accurately even in these challenging situations. Adding noise to the training inputs for generating motion embeddings is one method that can overcome these challenging situations. Extending the model to learn temporal alignment is also an important future work. We expect the extended model to produce better similarity predictions using both aligned and non-aligned action datasets and data-driven sequence alignment. Finally, evaluating the performance of existing tasks such as action recognition or person re-identification by applying motion similarity is also viable. In fact, the aforementioned tasks already take advantage of the concept

of motion similarity, and it would be interesting to see how the proposed method can contribute to those tasks.

Appendices

Appendix A

NTU RGB+D 120 Similarity Annotations

A.1 Data collection

We collected 20,093 motion similarity scores of NTU RGB+D 120 [4, 3] action sample pairs using Amazon Mechanical Turk (AMT). The annotation task (each HIT) consisted of 21 sets. One set was composed of a query video and 10 candidate videos, forming 10 pairs. Thus, there were 210 sample pairs per task. With random sampling, the distribution of scores would have been focused near a score of 1 (i.e. utterly different motions) since it is unlikely that a particular sample movement would have been similar to the movement of the sample from any other action category. To solve this problem, we adjusted the configuration of the pair selection strategy so that a wider variety of similarity scores was collected. When constructing a set, we included action videos in a fixed ratio of same (40%) and different (60%) actions. The different actions for the query were chosen from the action groups likely having similar movements to the query video (e.g. *capitulate* and *cheer up*). We created 100 tasks and performed a survey based on the tasks to obtain human evaluation. At least 10 AMT annotators examined each pair, and the final score was determined by averaging similarity scores for a particular pair.

Several examples of the motion pairs in AMT can be observed in Figure 4.1.

Figure 4.1 (a) is an example of score 4 because both people perform the same action - crossing hands in front. The two videos in Figure 4.1 (b) were given scores of 3. They share the same hand waving motion, but the arm movements are slightly different. In Figure 4.1 (c), the video on the left is the running-on-the-spot motion, while the right one is the butt-kicks motion. While they were in different action categories, the pair was scored 2, as there was similar movement between them. Lastly, Figure 4.1 (d) represents completely different motions. It was given a score of 1. Like the pair of videos in Figure 4.1, the instructions with a sample test were given to the AMT annotators before proceeding with the scoring (see Figures A.1 and A.2). In total, the final dataset after cleaning consisted of 20,093 pairs.

A.2 AMT score analysis

The average scores for each action are depicted in Figure A.3. The blue bar displays the average score of sample pairs belonging to the same action category, and the red bar displays the average score for all cases in which one of the pairs belongs to the corresponding action category and the other belongs to a different action category. Scores are generally high for pairs of the same action and mostly low for pairs of different actions. However, there are a few exceptions where low scores are observed for the same actions. For instance, two actions, A048 (*nausea/vomiting*) and A104 (*stretch oneself*), are scored relatively lower than other action pairs in the same action. We infer that the aforementioned actions can be expressed with various motions, causing the annotator to treat them as different.

On the other hand, two actions score relatively higher than other actions in a different action group: A095 (*capitulate*) and A104 (*stretch oneself*). For instance,

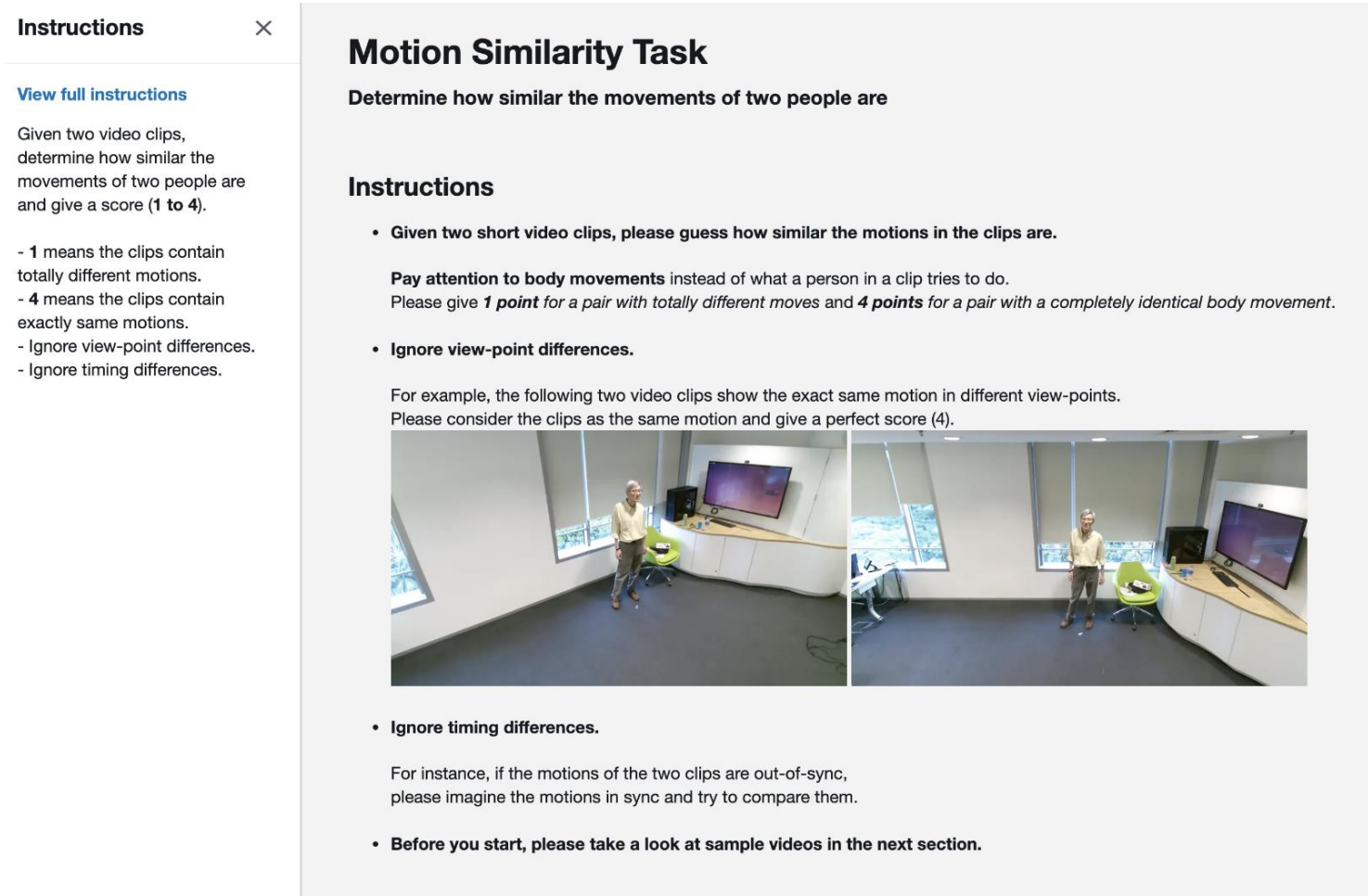


Figure A.1: Web page for AMT instructions. All video clips are from NTU RGB+D 120 [4, 3].

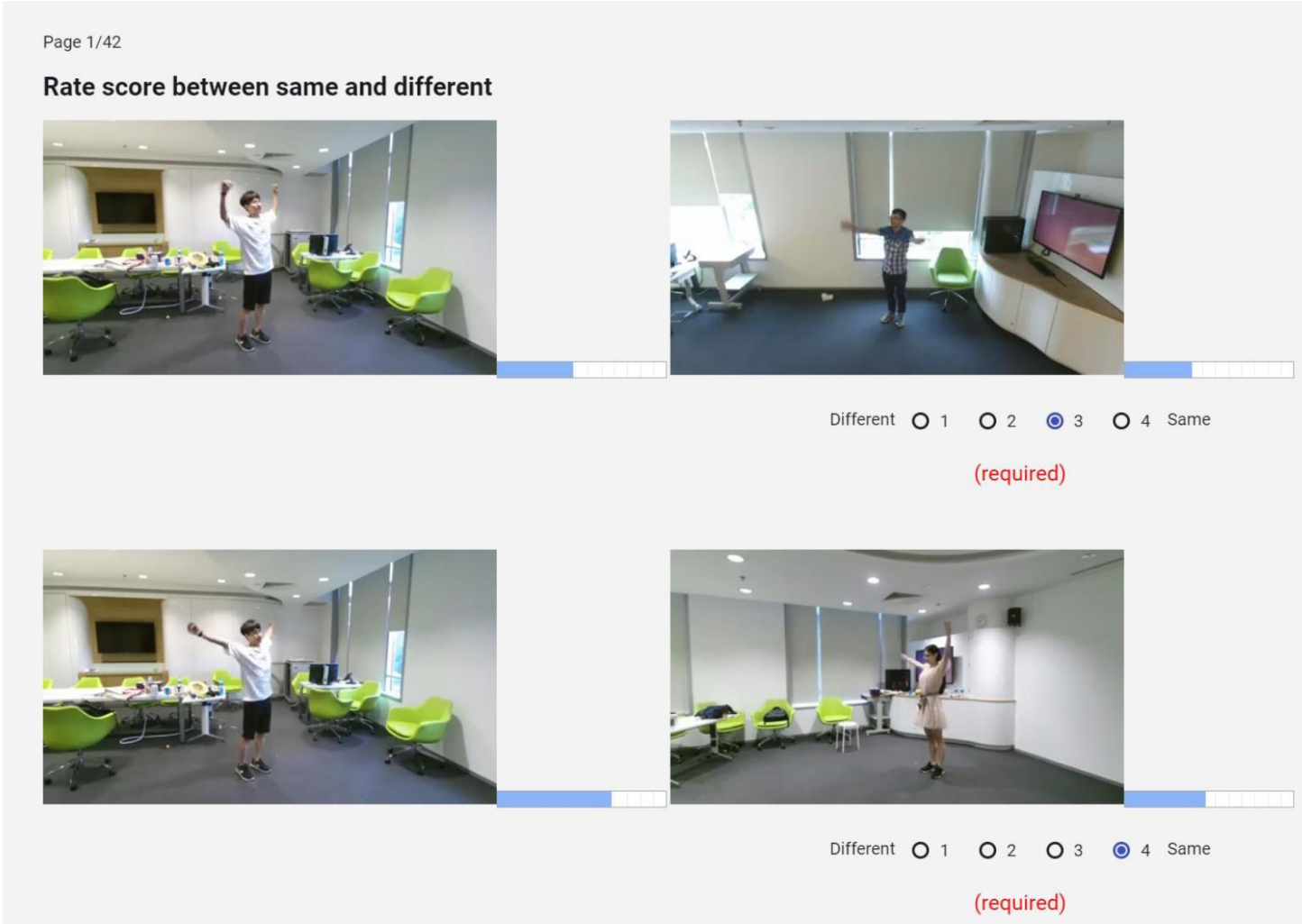


Figure A.2: Web page for AMT scoring. All video clips are from NTU RGB+D 120 [4, 3].

the *capitulate* action resembles A022 (*cheer up*) and A040 (*cross hands in front*) since both arms are raised while performing these actions. A104 (*stretch oneself*) is similar to A022 (*cheer up*) and A095 (*capitulate*) since these actions share movements such as lifting and stretching the arms. Because the images in an action pair with a high score have certain movements in common, we believe that our proposed method captures the motion similarity for each body part.

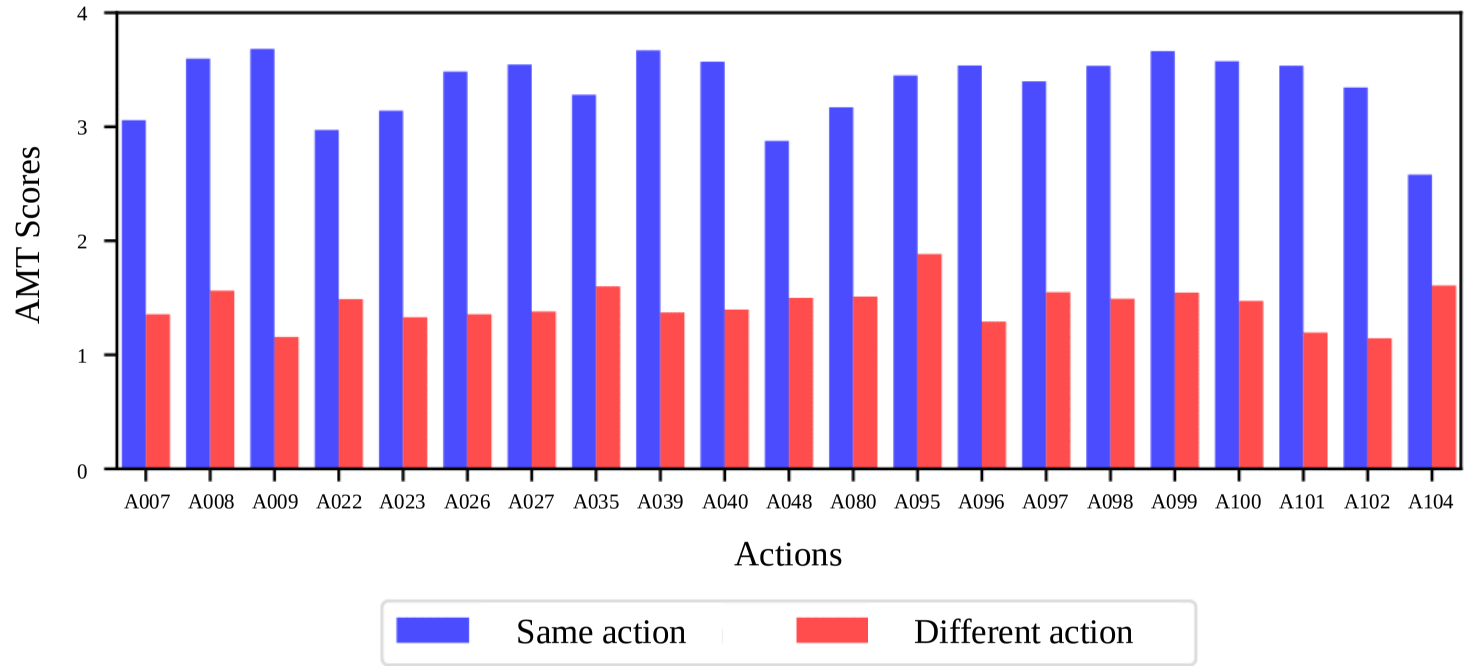


Figure A.3: Average AMT scores per action category. The blue and red bars mean the average scores when the similarity is measured with samples belonging to the same action category and different action categories, respectively.

Appendix B

Data Cleansing of NTU RGB+D 120 Skeletal Data

We noticed that some skeletal data in NTU RGB+D 120 is annotated imprecisely. Misannotated files can be classified into three cases. In the first case, a skeleton representing a human and a skeleton representing nonhuman objects are stored together, as shown on the left side of Figure B.1 (a). The problem is that we cannot identify a correct human skeleton without looking at the original video. In the second case, as shown on the left side of Figure B.1 (b), the skeleton is incorrectly located. In the last case, the joints have invalid annotations, as represented on the left side of Figure B.1 (c). To cope with these issues, we used our reproduction of MultiPoseNet [111] to generate new 2D joint annotations. More accurate skeleton data was obtained as a result, as shown on the right side of Figure B.1 (a), (b) and (c).

Figure B.1: Example of cleansing NTU RGB+D 120 [4, 3] skeletal data: In each example, the left side image is the original skeleton data from NTU RGB+D and the right side image is estimated joint annotations by our estimation model.

Appendix C

Motion Sequence Generation Using Mixamo

As mentioned in the paper, we utilized Adobe Mixamo [5] to generate motion sequences. Figure C.1 shows an example of generating a *Hip Hop Dancing* motion class. By adjusting the value of the slider on the right side, we could change the characteristic of the motion.

Figure C.1: Motion generation using the Mixamo [5] tool.

Bibliography

[1] K-fashion image dataset for artificial intelligence. `https://www.aihub.or.kr/aidata/7988`. Accessed: 2021-05-24.

[2] Edgar Schonfeld, Sayna Ebrahimi, Samarth Sinha, Trevor Darrell, and Zeynep Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8247–8255, 2019.

[3] Jun Liu, Amir Shahroudy, Mauricio Perez, Gang Wang, Ling-Yu Duan, and Alex C Kot. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2684–2701, 2019.

[4] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019, 2016.

[5] Adobe mixamo. `https://www.mixamo.com`. Accessed: 2021-05-22.

[6] Weibo Liu, Zidong Wang, Xiaohui Liu, Nianyin Zeng, Yurong Liu, and Fuad E Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.

[7] Antonio Brunetti, Domenico Buongiorno, Gianpaolo Francesco Trotta, and Vitoantonio Bevilacqua. Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing*, 300:17–33, 2018.

[8] Abdulla Al-Kaff, David Martin, Fernando Garcia, Arturo de la Escalera, and José María Armingol. Survey of computer vision algorithms and applications for unmanned aerial vehicles. *Expert Systems with Applications*, 92:447–463, 2018.

[9] Abhilash Nandy, Sushovan Haldar, Subhashis Banerjee, and Sushmita Mitra. A survey on applications of siamese neural networks in computer vision. In *International Conference for Emerging Technology (INCET)*, pages 1–5. IEEE, 2020.

[10] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. State-of-the-art in artificial neural network applications: A survey. *Heliyon*, 4(11):e00938, 2018.

[11] Muna O Almasawa, Lamiaa A Elrefaei, and Kawthar Moria. A survey on deep learning-based person re-identification systems. *IEEE Access*, 7:175228–175247, 2019.

[12] Kajaree Das and Rabi Narayan Behera. A survey on machine learning: Concept, algorithms and applications. *International Journal of Innovative Research in Computer and Communication Engineering*, 5(2):1301–1309, 2017.

[13] Sultan Daud Khan and Habib Ullah. A survey of advances in vision-based vehicle re-identification. *Computer Vision and Image Understanding*, 182:50–63, 2019.

[14] Jin Liu, Yi Pan, Min Li, Ziyue Chen, Lu Tang, Chengqian Lu, and Jianxin Wang. Applications of deep learning to mri images: A survey. *Big Data Mining and Analytics*, 1(1):1–18, 2018.

[15] Andre Esteva, Katherine Chou, Serena Yeung, Nikhil Naik, Ali Madani, Ali Mottaghi, Yun Liu, Eric Topol, Jeff Dean, and Richard Socher. Deep learning-enabled medical computer vision. *NPJ Digital Medicine*, 4(1):1–9, 2021.

[16] Bernard Marr. *Artificial intelligence in practice: How 50 successful companies used AI and machine learning to solve problems.* John Wiley & Sons, 2019.

[17] Annemarie J Nanne, Marjolijn L Antheunis, Chris G van der Lee, Eric O Postma, Sander Wubben, and Guda van Noort. The use of computer vision to analyze brand-related user generated image content. *Journal of Interactive Marketing*, 50:156–167, 2020.

[18] Alexandru Capatina, Maher Kachour, Jessica Lichy, Adrian Micu, Angela-Eliza Micu, and Federica Codignola. Matching the future capabilities of an artificial intelligence-based software for social media marketing with potential users' expectations. *Technological Forecasting and Social Change*, 151:119794, 2020.

[19] Most popular content categories on TikTok worldwide as of july 2020, by number of hashtag views. `https://www.statista.com/statistics/1130988/`

most-popular-categories-tiktok-worldwide-hashtag-views. Accessed: 2021-05-22.

[20] Ziwei Liu, Ping Luo, Shi Qiu, Xiaogang Wang, and Xiaoou Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1096–1104, 2016.

[21] Wenguan Wang, Yuanlu Xu, Jianbing Shen, and Song-Chun Zhu. Attentive fashion grammar network for fashion landmark detection and clothing category classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4271–4280, 2018.

[22] Menglin Jia, Mengyun Shi, Mikhail Sirotenko, Yin Cui, Claire Cardie, Bharath Hariharan, Hartwig Adam, and Serge Belongie. Fashionpedia: Ontology, segmentation, and an attribute localization dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 316–332. Springer, 2020.

[23] John Martinsson and Olof Mogren. Semantic segmentation of fashion images using feature pyramid networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 0–0, 2019.

[24] Yongwei Miao, Gaoyi Li, Chen Bao, Jiajing Zhang, and Jinrong Wang. Clothingnet: Cross-domain clothing retrieval with feature fusion and quadruplet loss. *IEEE Access*, 8:142669–142679, 2020.

[25] Ivona Tautkute, Tomasz Trzciński, Aleksander P Skorupa, Łukasz Brocki, and Krzysztof Marasek. Deepstyle: Multimodal search engine for fashion and interior design. *IEEE Access*, 7:84613–84628, 2019.

[26] Weiqian Li and Bugao Xu. Aspect-based fashion recommendation with attention mechanism. *IEEE Access*, 8:141814–141823, 2020.

[27] Cairong Yan, Yizhou Chen, and Lingjie Zhou. Differentiated fashion recommendation using knowledge graph and data augmentation. *IEEE Access*, 7:102239–102248, 2019.

[28] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989.

[29] Moeko Takagi, Edgar Simo-Serra, Satoshi Iizuka, and Hiroshi Ishikawa. What makes a style: Experimental analysis of fashion prediction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2247–2253, 2017.

[30] Ryusuke Miyamoto, Takeshi Nakajima, and Takuro Oki. Accurate fashion style estimation with a novel training set and removal of unnecessary pixels. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2019.

[31] M Hadi Kiapour, Kota Yamaguchi, Alexander C Berg, and Tamara L Berg. Hipster wars: Discovering elements of fashion styles. In *Proceedings of the*

European Conference on Computer Vision (ECCV), pages 472–488. Springer, 2014.

[32] Baoxin Wu, Chunfeng Yuan, and Weiming Hu. Human action recognition based on context-dependent graph kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2609–2616, 2014.

[33] Arridhana Ciptadi, Matthew S Goodwin, and James M Rehg. Movement pattern histogram for action recognition and retrieval. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 695–710. Springer, 2014.

[34] Ionut Cosmin Duta, Bogdan Ionescu, Kiyoharu Aizawa, and Nicu Sebe. Spatio-temporal vector of locally max pooled features for action recognition in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3097–3106, 2017.

[35] Huseyin Coskun, David Joseph Tan, Sailesh Conjeti, Nassir Navab, and Federico Tombari. Human motion analysis with deep metric learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 667–683, 2018.

[36] Zan Gao, Leming Guo, Tongwei Ren, An-An Liu, Zhi-Yong Cheng, and Shengyong Chen. Pairwise two-stream convnets for cross-domain action recognition with small data. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[37] Zan Gao, Leming Guo, Weili Guan, An-An Liu, Tongwei Ren, and Shengyong Chen. A pairwise attentive adversarial spatiotemporal network for cross-domain few-shot action recognition-r2. *IEEE Transactions on Image Processing*, 30:767–782, 2020.

[38] Fahn Chin-Shyurng, Shih-En Lee, and Meng-Luen Wu. Real-time musical conducting gesture recognition based on a dynamic time warping classifier using a single-depth camera. *Applied Sciences*, 9(3):528, 2019.

[39] Amani Elaoud, Walid Barhoumi, Ezzeddine Zagrouba, and Brahim Agrebi. Skeleton-based comparison of throwing motion for handball players. *Journal of Ambient Intelligence and Humanized Computing*, 11(1):419–431, 2020.

[40] Yeonho Kim and Daijin Kim. Real-time dance evaluation by markerless human pose estimation. *Multimedia Tools and Applications*, 77(23):31199–31220, 2018.

[41] Michalis Raptis, Darko Kirovski, and Hugues Hoppe. Real-time classification of dance gestures from skeleton animation. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 147–156, 2011.

[42] Rodrigo Schramm, Cláudio Rosito Jung, and Eduardo Reck Miranda. Dynamic time warping for music conducting gestures evaluation. *IEEE Transactions on Multimedia*, 17(2):243–255, 2014.

[43] Kai-Wen Cheng, Yie-Tarng Chen, and Wen-Hsien Fang. Video anomaly detection and localization using hierarchical feature representation and gaussian

process regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2909–2917, 2015.

[44] Romero Morais, Vuong Le, Truyen Tran, Budhaditya Saha, Moussa Mansour, and Svetha Venkatesh. Learning regularity in skeleton trajectories for anomaly detection in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11996–12004, 2019.

[45] Xinfeng Zhang, Su Yang, Xinjian Zhang, Weishan Zhang, and Jiulong Zhang. Anomaly detection and localization in crowded scenes by motion-field shape description and similarity-based statistical learning. *arXiv preprint arXiv:1805.10620*, 2018.

[46] Yumin Suh, Jingdong Wang, Siyu Tang, Tao Mei, and Kyoung Mu Lee. Part-aligned bilinear representations for person re-identification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 402–419, 2018.

[47] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6479–6488, 2018.

[48] Lin Wu, Yang Wang, Junbin Gao, and Xue Li. Where-and-when to look: Deep siamese attention networks for video-based person re-identification. *IEEE Transactions on Multimedia*, 21(6):1412–1424, 2018.

[49] Wei Zhang, Yimeng Li, Weizhi Lu, Xinshun Xu, Zhaowei Liu, and Xiangyang Ji. Learning intra-video difference for person re-identification. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(10):3028–3036, 2018.

[50] Han-Jia Ye, Hexiang Hu, and De-Chuan Zhan. Learning adaptive classifiers synthesis for generalized few-shot learning. *International Journal of Computer Vision*, pages 1–24, 2021.

[51] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

[52] Dvir Samuel, Yuval Atzmon, and Gal Chechik. From generalized zero-shot learning to long-tail with class descriptors. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 286–295, 2021.

[53] Kfir Aberman, Rundi Wu, Dani Lischinski, Baoquan Chen, and Daniel Cohen-Or. Learning character-agnostic motion for motion retargeting in 2d. *ACM Transactions on Graphics (TOG)*, 38(4):75, 2019.

[54] Fanjia Li, Aichun Zhu, Yonggang Xu, Ran Cui, and Gang Hua. Multi-stream and enhanced spatial-temporal graph convolution network for skeleton-based action recognition. *IEEE Access*, 8:97757–97770, 2020.

[55] Yun Han, Sheng-Luen Chung, Qiang Xiao, Wei You Lin, and Shun-Feng Su. Global spatio-temporal attention for action recognition based on 3d human skeleton data. *IEEE Access*, 8:88604–88616, 2020.

[56] Hongye Yang, Yuzhang Gu, Jianchao Zhu, Keli Hu, and Xiaolin Zhang. Pgcn-tca: Pseudo graph convolutional network with temporal and channel-wise at-

tention for skeleton-based action recognition. *IEEE Access*, 8:10040–10047, 2020.

[57] Yingfu Wang, Zheyuan Xu, Li Li, and Jian Yao. Robust multi-feature learning for skeleton-based action recognition. *IEEE Access*, 7:148658–148671, 2019.

[58] Weizhi Nie, Wei Wang, and Xiangdong Huang. Srnet: Structured relevance feature learning network from skeleton data for human action recognition. *IEEE Access*, 7:132161–132172, 2019.

[59] Yanbo Fan, Shuchen Weng, Yong Zhang, Boxin Shi, and Yi Zhang. Context-aware cross-attention for skeleton-based human action recognition. *IEEE Access*, 8:15280–15290, 2020.

[60] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 143–152, 2020.

[61] Hao Yang, Dan Yan, Li Zhang, Dong Li, YunDa Sun, ShaoDi You, and Stephen J Maybank. Feedback graph convolutional network for skeleton-based action recognition. *arXiv preprint arXiv:2003.07564*, 2020.

[62] Konstantinos Papadopoulos, Enjie Ghorbel, Djamila Aouada, and Björn Ottersten. Vertex feature encoding and hierarchical temporal modeling in a spatio-temporal graph convolutional network for action recognition. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 452–458. IEEE, 2021.

[63] Amazon mechanical turk. `https://www.mturk.com`. Accessed: 2021-05-22.

[64] Yihui Ma, Jia Jia, Suping Zhou, Jingtian Fu, Yejun Liu, and Zijian Tong. Towards better understanding the clothing fashion styles: A multimodal deep learning approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.

[65] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[66] Buyu Li, Yu Liu, and Xiaogang Wang. Gradient harmonized single-stage detector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8577–8584, 2019.

[67] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9268–9277, 2019.

[68] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *arXiv preprint arXiv:1906.07413*, 2019.

[69] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[70] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *Proceedings of the International Conference on Intelligent Computing*, pages 878–887. Springer, 2005.

[71] Chris Drummond, Robert C Holte, et al. C4. 5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *Workshop on Learning from Imbalanced Datasets II*, volume 11, pages 1–8. Citeseer, 2003.

[72] Yijie Huang, Zhenghong Deng, and Tao Wu. Learning discriminative latent features for generalized zero-and few-shot searning. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020.

[73] Yongqin Xian, Bruno Korbar, Matthijs Douze, Bernt Schiele, Zeynep Akata, and Lorenzo Torresani. Generalized many-way few-shot video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 111–127. Springer, 2020.

[74] Yao-Hung Hubert Tsai, Liang-Kang Huang, and Ruslan Salakhutdinov. Learning robust visual-semantic embeddings. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3571–3580, 2017.

[75] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[76] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[77] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in Neural Information Processing Systems*, 28:3483–3491, 2015.

[78] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[79] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[80] Erdenebileg Batbaatar, Kwang Ho Park, Tsatsral Amarbayasgalan, Khishigsuren Davagdorj, Lkhagvadorj Munkhdalai, Van-Huy Pham, and Keun Ho Ryu. Class-incremental learning with deep generative feature replay for dna methylation-based cancer classification. *IEEE Access*, 8:210800–210815, 2020.

[81] In-Chul Yoo, Keonnyeong Lee, Seonggyun Leem, Hyunwoo Oh, Bonggu Ko, and Dongsuk Yook. Speaker anonymization for personal information protection using voice conversion techniques. *IEEE Access*, 8:198637–198645, 2020.

[82] Andrea Asperti and Matteo Trentin. Balancing reconstruction error and kullback-leibler divergence in variational autoencoders. *IEEE Access*, 8:199440–199448, 2020.

[83] Ahlem Drif, Houssem Eddine Zerrad, and Hocine Cherifi. Ensvae: Ensemble variational autoencoders for recommendations. *IEEE Access*, 8:188335–188351, 2020.

[84] Vittorio Ferrari, Manuel Marin-Jimenez, and Andrew Zisserman. Pose search: Retrieving people using their pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2009.

[85] Nataraj Jammalamadaka, Andrew Zisserman, Marcin Eichner, Vittorio Ferrari, and CV Jawahar. Video retrieval by mimicking poses. In *Proceedings of the ACM International Conference on Multimedia Retrieval*, pages 1–8, 2012.

[86] Andreas Aristidou, Daniel Cohen-Or, Jessica K Hodgins, Yiorgos Chrysanthou, and Ariel Shamir. Deep motifs and motion signatures. *ACM Transactions on Graphics (TOG)*, 37(6):1–13, 2018.

[87] Yijun Shen, Longzhi Yang, Edmond SL Ho, and Hubert PH Shum. Interaction-based human activity comparison. *IEEE Transactions on Visualization and Computer Graphics*, 26(8):2620–2633, 2019.

[88] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7024–7033, 2018.

[89] Xiao Guo and Jongmoo Choi. Human motion prediction via learning local structure representations and temporal dependencies. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2580–2587, 2019.

[90] An-An Liu, Yu-Ting Su, Ping-Ping Jia, Zan Gao, Tong Hao, and Zhao-Xuan Yang. Multiple/single-view human action recognition via part-induced multi-task structural learning. *IEEE Transactions on Cybernetics*, 45(6):1194–1208, 2014.

[91] Yong-Lu Li, Liang Xu, Xinpeng Liu, Xijie Huang, Yue Xu, Mingyang Chen, Ze Ma, Shiyi Wang, Hao-Shu Fang, and Cewu Lu. Hake: Human activity knowledge engine. *arXiv preprint arXiv:1904.06539*, 2019.

[92] Nataraj Jammalamadaka, Andrew Zisserman, and CV Jawahar. Human pose search using deep networks. *Image and Vision Computing*, 59:31–43, 2017.

[93] Greg Mori, Caroline Pantofaru, Nisarg Kothari, Thomas Leung, George Toderici, Alexander Toshev, and Weilong Yang. Pose embeddings: A deep architecture for learning to match human poses. *arXiv preprint arXiv:1507.00302*, 2015.

[94] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.

[95] Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3d pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3109–3118, 2015.

[96] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[97] Sungyeon Kim, Minkyo Seo, Ivan Laptev, Minsu Cho, and Suha Kwak. Deep metric learning beyond binary supervision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2288–2297, 2019.

[98] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. Beyond triplet loss: A deep quadruplet network for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 403–412, 2017.

[99] Jeany Son, Mooyeol Baek, Minsu Cho, and Bohyung Han. Multi-object tracking with quadruplet convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5620–5629, 2017.

[100] Richard Bellman and Robert Kalaba. On adaptive control processes. *IRE Transactions on Automatic Control*, 4(2):1–9, 1959.

[101] Francisco Javier Torres Reyes. *Human motion: Analysis of similarity and dissimilarity using orthogonal changes of direction on given trajectories.* University of Colorado at Colorado Springs, 2016.

[102] Yongqin Xian, Christoph H Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, 2018.

[103] Clark R Givens and Rae Michael Shortt. A class of wasserstein metrics for probability distributions. *The Michigan Mathematical Journal*, 31(2):231–240, 1984.

[104] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 289–305, 2018.

[105] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[106] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[107] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, page 807–814. Omnipress, 2010.

[108] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010.

[109] Genevieve Patterson and James Hays. Sun attribute database: Discovering, annotating, and recognizing scene attributes. In *Proceedings of the IEEE Con-*

ference on Computer Vision and Pattern Recognition (CVPR), pages 2751–2758. IEEE, 2012.

[110] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Elsevier, 1989.

[111] Muhammed Kocabas, Salih Karagoz, and Emre Akbas. Multiposenet: Fast multi-person pose estimation using pose residual network. In *Proceedings of the European conference on computer vision (ECCV)*, pages 417–433, 2018.

[112] Claudio Gentile and Manfred KK Warmuth. Linear hinge loss and average margin. *Advances in Neural Information Processing Systems*, 11:225–231, 1998.

[113] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 448–456. PMLR, 2015.

[114] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 30, page 3. Citeseer, 2013.

[115] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[116] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[117] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008.

[118] Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[119] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[120] Yuval Atzmon and Gal Chechik. Probabilistic and-or attribute grouping for zero-shot learning. *arXiv preprint arXiv:1806.02664*, 2018.

국문초록

컴퓨터 비전은 딥러닝 학습 방법론이 강점을 보이는 분야로, 다양한 태스크에서 우수한 성능을 보이고 있다. 특히, 사람이 포함된 이미지나 동영상을 딥러닝을 통해 분석하는 태스크의 경우, 최근 소셜 미디어에 사람이 포함된 이미지 또는 동영상 게시물이 늘어나면서 그 활용 가치가 높아지고 있다.

본 논문에서는 사람과 관련된 컴퓨터 비전 태스크 중 패션 스타일 분류 문제와 동작 유사도 측정에 대해 다룬다. 패션 스타일 분류 문제의 경우, 데이터 수집 시점의 패션 유행에 따라 스타일 클래스별 수집되는 샘플의 양이 달라지기 때문에 이로부터 클래스 불균형이 발생한다. 본 논문에서는 이러한 클래스 불균형 문제에 대처하기 위하여, 소수 샘플 클래스와 다수 샘플 클래스를 학습 및 평가에 모두 사용하는 일반화된 퓨샷러닝으로 패션 스타일 분류 문제를 설정하였다. 또한 변분 오토인코더 기반의 모델을 통해, 신체 및 패션 아이템 부분만 잘라낸 전경 이미지 모달리티와 패션 속성 정보 모달리티가 패션 이미지의 임베딩 학습에 반영되도록 하였다. 학습 및 평가를 위한 데이터셋으로는 한국 패션 쇼핑몰에서 수집된 K-fashion 데이터셋을 사용하였다.

한편, 동작 유사도 측정은 행위 인식, 이상 동작 감지, 사람 재인식 같은 다양한 분야의 하위 모듈로 활용되고 있지만 그 자체가 연구된 적은 많지 않은데, 이는 같은 동작을 수행하더라도 신체 구조 및 카메라 각도에 따라 다르게 표현될 수 있다는 점으로 부터 기인한다. 학습 및 평가를 위한 공개 데이터셋이 많지 않다는 점 또한 연구를 어렵게 하는 요인이다. 따라서 본 논문에서는 학습을 위한 인공 데이터셋을 수집하여 오토인코더 구조를 통해 신체 구조 및 카메라 각도 요소가 분리된 동작 임베딩을 학습하였다. 이때, 각 신체 부위별로 동작 임베딩을 생성할 수 있도록하여 신체 부위별로 동작 유사도 측정이 가능하도록 하였다. 두 동작 사이의 유사도를 측정할 때에는 동적 시간

워핑 기법을 사용, 비슷한 동작을 수행하는 구간끼리 정렬시켜 유사도를 측정하도록 함으로써, 동작 수행 속도의 차이를 보정하였다. 평가를 위한 유사도 점수 데이터셋은 행위 인식 데이터셋인 NTU-RGB+D 120의 영상을 활용하여 크라우드 소싱 플랫폼을 통해 수집되었다.

두 가지 태스크의 제안 모델을 각각의 평가 데이터셋으로 검증한 결과, 모두 비교 모델 대비 우수한 성능을 기록하였다. 패션 스타일 분류 문제의 경우, 모든 비교군에서 소수 샘플 클래스와 다수 샘플 클래스 중 한 쪽으로 치우치지 않는 가장 균형잡힌 추론 성능을 보여주었고, 동작 유사도 측정의 경우 사람이 측정한 유사도 점수와 상관계수에서 비교 모델 대비 더 높은 수치를 나타내었다.

주요어: 신체, 패션, 동작 분석, 컴퓨터 비전 응용, 오토인코더, 산업공학

학번: 2016-21106