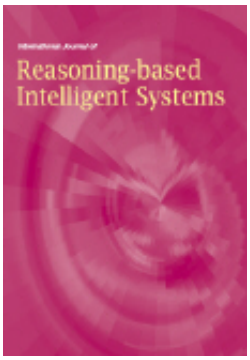


International Journal of Reasoning-based Intelligent Systems

ISSN online: 1755-0564 - ISSN print: 1755-0556
https://www.inderscience.com/ijris

Based on transfer learning and graph neural network for animated clothing element recognition

Yue Wu, Yunfeng Hu, Hao Luo

DOI: 10.1504/IJRIS.2025.10071593

Article History:

Received:	31 March 2025
Last revised:	24 April 2025
Accepted:	25 April 2025
Published online:	11 June 2025

Based on transfer learning and graph neural network for animated clothing element recognition

Yue Wu

School of Arts,
Sanjiang University,
Nanjing 210000, China
Email: wuyue_sju@126.com

Yunfeng Hu

School of Arts,
Sanjiang University,
Nanjing 210000, China
and
Faculty of Fine and Applied Arts and Cultural Science,
Mahasarakham University,
Maha Sarakham 44000, Thailand
Email: sjuhyf@163.com

Hao Luo*

School of Arts,
Sanjiang University,
Nanjing 210000, China
Email: luohao_sju@126.com
*Corresponding author

Abstract: Animation costume element recognition has become a major computer vision research area given the fast growth of the animation sector. Conventions for costume recognition include issues including significant stylistic variations and little annotating data. This work suggests a framework for animation costume element detection based on transfer learning (TL) and graph neural network (GNN) in order to solve these difficulties AnimCloth-TL-GNN. First, the framework models the spatial relationship between costume elements by the TL module; then, the GNN module models the knowledge from the source domain to the target domain; finally, the fusion module combines the features of both to improve the recognition effect of the model. Strong advantages in the animated costume element recognition job are shown by the experimental findings of the AnimCloth-TL-GNN framework, thereby enhancing the accuracy and has great generalising capacity and robustness.

Keywords: transfer learning; TL; graph neural network; GNN; animated costume elements.

Reference to this paper should be made as follows: Wu, Y., Hu, Y. and Luo, H. (2025) 'Based on transfer learning and graph neural network for animated clothing element recognition', *Int. J. Reasoning-based Intelligent Systems*, Vol. 17, No. 7, pp.33–42.

Biographical notes: Yue Wu received her Master degree form Nanjing Normal University in June 2010. Currently, she works in Sanjiang University. Her research interests include machine learning and animation technology.

Yunfeng Hu received his Doctor's degree form Mahasarakham University in 2025. Currently, he works in Sanjiang University. His research interests include animation technology and artificial general intelligent control (AGIC).

Hao Luo received receive his Doctoral degree form Beijing Institute of Fashion Technology in January 2025. Currently, he works in Sanjiang University. His research interests include digital images and clothing culture.

1 Introduction

Clothing element recognition has progressively become a major research direction in the field of image recognition, especially in fashion recommendation, virtual fitting, digital clothing design and other applications with wide prospects, with the fast development of computer vision and deep learning technology. Clothing element recognition requires the link between several categories, changes in character poses and the complexity of images in addition to identifying the kind, colour, style and other characteristics of apparel (Chen et al., 2023a).

Although animated garment element recognition is a relatively new area of research, several relevant studies have been conducted. Most of these studies have focused on the application of traditional computer vision techniques, including object recognition and image classification methods (Bi et al., 2022; Sakshi and Kukreja, 2023). These techniques have obtained good recognition results by training on big-scale datasets and mostly rely on the capacity of convolutional neural networks (CNNs) to extract picture features (Xiao et al., 2023). But conventional CNN techniques can struggle to analyse intricate, non-real-world outfit imagery, particularly in relation to highly styled animation visuals and their recognition outcomes are sometimes unsatisfactory.

Based on this, Zhao et al. (2023) and Chen et al. (2023b) have started to attempt using generative models such generative adversarial networks (GAN) and variational auto-encoders (VAE) for the production and style alteration of costume photos. For instance, GAN's style migration of realistic costume photos are utilised to create costume images that match the animation style, therefore enhancing the recognition of costume aspects. These techniques still have some restrictions in practical applications due to the variations in visual effects between the produced images and the original data; particularly in cases of highly detailed and sophisticated animated costume elements, the performance of the model is not as expected.

GNNs have now started to be used in garment element recognition, particularly in order to replicate the spatial relationships among elements in apparel photographs (Ding et al., 2023). Aiming to represent the interdependence of costume pieces and enhance the recognition result by GNNs, Zhang et al. (2024) have merged GNNs with conventional CNNs and presented an approach combining picture segmentation and object detection. GNN still has some room for development in its application effect, though, as it deals with the challenging recognition task of animated costume elements and must effectively build the graph structure and choose the suitable graph neural network (GNN) architecture.

In the field of transfer learning (TL), most research continues to focus on the migration of knowledge from conventional costume images to virtual costumes or animated images. Although some studies attempt to transfer knowledge from the source domain of costume images to the target domain of animated image recognition, challenges remain. While this approach has made significant progress in reducing data requirements, successfully migrating knowledge when faced with style-specific costume elements (e.g., animated style costumes) remains a major challenge. Particularly in the case of animated costume element detection, it is still difficult to handle the great variations between the source and target domains in terms of picture attributes, styles, and so forth (Ribet et al., 2019).

Although current research has made significant contributions to the field of costume element recognition, there are still many limits overall, particularly when it comes to identifying animation style costumes, both conventional and other developing techniques show different degrees of shortcomings. Thus, this work suggests the AnimCloth-TL-GNN framework, which seeks to use TL and GNN to solve the problems in animated costume element recognition and hence further the research development in this domain.

This paper's innovations consist in the following:

1 Framework design based on TL and GNN: AnimCloth-TL-GNN is proposed, a framework combining TL and GNN for recognition of animated costume elements, which creatively TL to the field of animated costumes, overcoming the problem of insufficient annotation data, and at the same time models the complex relationship between costume elements by using GNN to improve the recognition effect. Concurrently, GNN models the intricate interactions among costume pieces, hence enhancing the recognition result.

2 Optimised loss function: the model training process uses an optimised loss function, which guarantees the convergence and stability of the model by precisely modifying the loss weights during the training process. This invention helps the model to effectively improve recognition accuracy in multi-task learning.

3 Multi-level experimental validation: multiple testing environments allow one to confirm the performance benefits of the AnimCloth-TL-GNN framework under several comparison models. Apart from the comparison with conventional deep learning models, the studies extensively investigate the correlation between the training length and the convergence speed of the model, so offering theoretical and experimental basis for model optimisation.

2 Relevant technologies

2.1 Transfer learning

Through the knowledge gained in the source task, TL is a method to enhance the learning process of a target task. While in real-world applications the source and target tasks generally differ, in conventional machine learning it is believed that their data distributions are the same.

TL's main goal is to minimise the variation between the source and target activities so that the information gained in the former can be reasonably applied to the latter (Yao

et al., 2023; Dai and Meng, 2023). Usually, TL aims to generate superior learning results by reducing the distributional disparities between them given a source task TS and a target task T_T together with a training dataset D_S for the source task and a training dataset D_T for the target task.

TL makes a fundamental assumption that the distributions of the target and source tasks exhibit some correlation or similarity. Calculating the distributional difference between the source and target tasks, such as through the Kullback-Leibler (KL) divergence, helps quantify this similarity. The KL scatter can be stated with the data distributions P_S and P_T of the target and source jobs as:

$$D_{KL}\left(P_T \left| P_S\right.\right) = \sum_x P_T(x) \log\left(\frac{P_T(x)}{P_S(x)}\right) \tag{1}$$

where D_{KL} indicates the KL dispersion between the source and target tasks; $P_T(x)$ and $P_S(x)$ respectively represent the probability distributions of the target and source tasks.

Knowledge transfer between source and destination tasks typically needs fine-tuning the source task model if TL is to be more effective (Ouyang et al., 2023). Usually, one wants to migrate the knowledge from the source task model to the target task assuming θ_S as the model parameter of the source task and θ_T as the model parameter of the target task. Usually speaking, the target task's loss function is stated as:

$$L_T(\theta_T) = \mathbb{E}_{(x,y)\sim P_T}\left[L\left(f\left(x;\theta_T\right), y\right)\right] \tag{2}$$

where L is the loss function; $f(x; \theta_T)$ is the target task model's output; P_T is the target task's data distribution.

Apart from parameter migration of models, feature migration is a prevalent tactic in TL. Through the mapping action T, feature migration seeks to map the feature space of the source task to the feature space of the target task. The process of feature migration can be stated for the data x_T in the target job as:

$$x_T' = T\left(x_T\right) \tag{3}$$

where x_T' is the feature of the target task acquired upon mapping.

Usually, the model of the target task is improved by reducing the loss function of the target task so attaining effective learning of the target task (Han et al., 2018). Assuming θ_T as the target task model's parameters, the target task model's optimisation procedure may be stated as:

$$\theta_T^* = \arg\min_{\theta_T} \mathbb{E}_{(x,y)\sim P_T}\left[L\left(f\left(x;\theta_T\right), y\right)\right] \tag{4}$$

Apart from direct parameter migration, the domain adaptation method in TL adjusts the distributional difference between the source and destination tasks thereby attaining effective knowledge translation. Reducing the maximum mean difference (MMD) between the source and target activities is a widely used domain adaptation method. MMD has the formula shown here:

$$L_{MMD} = \left\| \frac{1}{n_S}\sum_{i=1}^{n_S}\phi\left(x_S^{(i)}\right) - \frac{1}{n_T}\sum_{j=1}^{n_T}\phi\left(x_T^{(j)}\right)\right\|_H^2 \tag{5}$$

where $x_S^{(i)}$ and $x_T^{(j)}$ represent the samples in the source and target tasks correspondingly, n_S and n_T are the number of samples in the source and target tasks, and $/\!/ \cdot /\!/_H$ is the number of paradigms in the feature space $\phi(\cdot)$.

Furthermore, the regularisation term is crucial in TL. Typically, the regularisation term is employed to manage the variations between the model parameters of the target task and those of the source task, thereby preventing excessive disparities between the two tasks. More specifically, a regularisation term is typically included to the target task's loss function under the denotation:

$$L_T\left(\theta_T\right) = \mathbb{E}_{(x,y)\sim P_T}\left[L\left(f\left(x;\theta_T\right), y\right)\right] + \lambda\left\|\theta_T - \theta_S\right\|^2 \tag{6}$$

where $/\!/ \theta_T - \theta_S /\!/$ shows the variation between the target and source task model parameters and λ is a regularisation parameter.

Moreover, a popular method in TL is to attain cross-task learning through parameter sharing in order to better learn the knowledge in the target task (Jiang et al., 2022). Assuming some parameters shared across the models of the source and target tasks, the method of parameter optimisation for the target task can be stated as:

$$\theta_T^* = \arg\min_{\theta_T} \mathbb{E}_{(x,y)\sim P_T}\left[L\left(f\left(x;\theta_T,\theta_S\right), y\right)\right] \tag{7}$$

where θ_T^* is the goal task model's parameter for which the best performance is sought.

By means of these techniques, TL may efficiently migrate the knowledge from the source task to the target task, hence enhancing the learning performance of the target task particularly in cases of limited target task data. In the actual world, TL offers a workable answer for the issues of data shortage and labelling challenge.

2.2 Graph neural network

GNN is a family of neural network models designed for graph-structured data processing. Many applications like social networks, chemical structures, recommender systems, etc. depend on graph-structured data somewhat extensively. GNN is fundamentally based on learning the representation of nodes or graphs by means of information propagation along the graph topography (Zhou et al., 2022). Through collaborative modelling of the individual nodes and their neighbours in the graph, GNNs are able to capture the underlying relationships and patterns in the graph structure.

A graph G consists in an edge set E and a node set V. G then can be expressed as:

$$G = (V, E) \tag{8}$$

where V and E can be represented correspondingly as:

$$V = \left\{v_1, v_2, \ldots, v_N\right\} \tag{9}$$

$$E = \{e_{ij}\} \tag{10}$$

Every node $v_i \in V$ can feature a vector $x_i \in \mathbb{R}^d$ that denotes its attribute information. GNN aims to learn the embedding of nodes or graphs by means of structural information of the graph updating of every node's representation.

GNN is fundamentally based on the aggregation and updating of node information process. Particularly, for every node v_i, combining the data of its surrounding nodes updates its representation. Assuming $N(v_i)$ as the set of neighbour nodes of node vi, its update rule may be stated as follows:

$$h_i^{(l+1)} = \sigma\left(W^{(l)} \cdot \left(h_i^{(l)} \oplus \sum_{j \in N(v_i)} h_j^{(l)}\right)\right) \tag{11}$$

where $h_i^{(l)}$ represents node v_i at layer l; $\oplus$ is the feature splicing mechanism; $W^{(l)}$ is the weight matrix at layer l; σ is the activation function. The formula shows the how the representations of the surrounding nodes update the representation of node v_i.

As the number of network layers deepens, the representation of a node progressively accumulates the data of its more distant neighbours during the GNN training process (Guo et al., 2023). The GNN can thus spread information over several layers, hence capturing long-range dependencies in the graph structure. Usually, a GNN will have several aggregations layers to progressively gather the global information of nodes.

One of the primary characteristics of GNN is their adaptability. They can effectively handle weighted graphs, where the weight of each edge conveys specific information, as well as unweighted graphs, where all edges share the same weight. The update algorithm for node representation for weighted graphs can be weighted summation:

$$h_i^{(l+1)} = \sigma\left(W^{(l)} \cdot \left(h_i^{(l)} \oplus \sum_{j \in N(v_i)} w_{ij} h_j^{(l)}\right)\right) \tag{12}$$

where w_{ij} is edge e_{ij}'s weight.

Common variation of GNN, graph convolutional network (GCN), uses the adjacency matrix of the graph to depict the connectivity among nodes (Jia et al., 2023). Assume the degree matrix of every node in the graph is D while the adjacency matrix of the graph is A. Usually marked as GCN's update rule is:

$$H^{(l+1)} = \sigma\left(\hat{A} H^{(l)} W^{(l)}\right) \tag{13}$$

$$\hat{A} = D^{-1/2} A D^{-1/2} \tag{14}$$

where $H^{(l)}$ is the node representation of the l^{th} layer; $\hat{A}$ is the normalised adjacency matrix. By updating the representation of every node with the structural information of the network and the normalised adjacency matrix, this approach reduces over-reliance on node degree.

Apart from GCN, graph pooling is a prevalent technique used in graph networking. Graph pooling techniques allow one to map a node's representation to the representation of the whole graph.

Pooling techniques are widely employed to produce global properties of the graph in graph classification problems. Maximum pooling and average pooling are two common pooling techniques; maximum pooling may be stated as:

$$h_{graph} = \max_{v_i \in V} h_i^{(L)} \tag{15}$$

where $h_i^{(L)}$ represents node v_i at layer L and h_{graph} represents the full graph.

In useful applications of GNNs, it is frequently necessary to optimise the model by means of an objective task. In a graph classification job, for instance, pooling activities or fully linked layers yields the final graph representation. The loss function for graph classification can be stated assuming that the full graph is h_{graph} as:

$$L_{graph} = \mathbb{E}_G\left[L\left(f(G;\theta), y\right)\right] \tag{16}$$

where L is a loss function; θ is a parameter of the model; $f(G; \theta)$ is a representation of the graph G; y is a label of the graph.

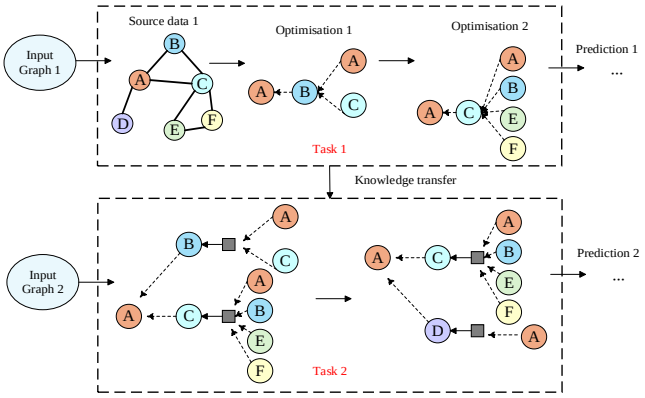
Usually, producing the expected output of the graph from the node information in the graph comes last in a GNN (Bianchi and Lachi, 2023). Should node classification be employed in the graph, the prediction process typically shows as:

$$\hat{y}_i = \text{softmax}\left(W_{\text{out}} \cdot h_i^{(L)}\right) \tag{17}$$

where $\hat{y}_i$ is the expected label of node v_i; W_{out} is the weight matrix of the output layer; softmax is the activation function creating the classification probability.

Regarding the representation learning of graph-structured data, the GNN performs remarkably. GNN can effectively capture the relationships between nodes and obtain an awareness of the whole graph structure by continuously updating the representation of nodes by means of multi-layer information propagation. When handling graph data, this GNN architecture provides a benefit over conventional neural networks; moreover, it has particularly shown amazing success in social networks, recommender systems, and chemical compounds.

Figure 1 Structure of AnimCloth-TL-GNN (see online version for colours)



3 A framework for recognising animated costume elements

3.1 Framework construction

AnimCloth-TL-GNN consists of the five modules: a TL module, a GNN module, a fusion module and a prediction output module based on TL and GNN. See Figure 1; every module works in concert to finish the entire process from feature extraction to costume element recognition.

In Figure 1, the symbols A/B/C/D/E/F represent different modules:

A is TL module, which is responsible for TL

B is GNN module, which is used for GNN processing

C is fusion module, which carries out feature fusion

D is prediction output module, which outputs the prediction results

E is the loss function module, which calculates the loss function

F is the data input module, which inputs data.

3.1.1 Transfer learning module

Applying knowledge from the source domain to the target domain to enhance the learning of the target task is the primary objective of the TL module. The target domain of the AnimCloth-TL-GNN system focuses on recognising animated costume elements, while the source domain comprises an existing costume dataset. By transferring feature representations and model parameters from the source domain, we can reduce the target domain's dependence on labelled data and improve the model's generalisation capabilities. One may represent the loss function of this procedure as:

$$L_{TL} = \mathbb{E}_{D_s}\left[L(f(x_s;\theta_s), y_s)\right] + \mathbb{E}_{D_t}\left[L(f(x_t;\theta_t), y_t)\right] + \lambda \cdot R(\theta_s, \theta_t) \quad (18)$$

where L is the loss function; θ_s and θ_t indicate the model parameters of the source and target domains respectively; $R(\theta_s, \theta_t)$ is a regularisation term used to encourage the parameter consistency between the source and target domains; λ is a hyperparameter controlling the strength of the regularisation term and $f(x; \theta)$ denotes the prediction function of the model. Reducing this loss function enables the TL module to efficiently move knowledge from the source domain to the target domain, hence enhancing the target task's accuracy (Chen et al., 2023c).

3.1.2 Graph neural network module

This module in the AnimCloth-TL-GNN system models the relationships among costume elements via a graph structure. Every dress element is portrayed as a node in the graph, and edges link the nodes to one another. The GNN learns the contextual relationships between the clothing pieces and aggregates the information of adjacent nodes by means of the graph convolution process.

First, the adjacency matrix A of the graph explains the relationships of connection between the nodes. The graph convolution process changes the node characteristics using this formula:

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in N(i)} A_{ij} W^{(l)} h_j^{(l)}\right) \quad (19)$$

Gradually aggregating local and global information, the multilayer graph convolution process increases model recognition capacity.

Furthermore, while keeping crucial node information, the graph pooling process can help to lower the computational cost. One may represent the graph pooling process by the following equation:

$$h_i^{\text{pool}} = \text{Pooling}\left(\left\{h_j^{(l)} \,\middle|\, j \in N(i)\right\}\right) \quad (20)$$

Pooling compresses the features of the nodes into a more compact form, therefore improving the expressiveness and efficiency of the model (Cong and Zhou, 2023)

By means of these processes, the GNN module effectively learns the properties of costume elements and offers strong contextual support for animated costume element recognition.

3.1.3 Fusion module

The function of the fusion module in the AnimCloth-TL-GNN architecture is to combine the features extracted by the GNN module and the migration learning module thereby producing a more discriminative feature representation for the final recognition of animated costume pieces. The model is able to maximise the source domain knowledge and the graph structure information by efficiently combining the features from these two sections, so enhancing the recognition performance.

The fusion module weights the mix of the two features by the following formula assuming h_{TL} as the output feature of the migration learning module and h_{GNN} as the output feature of the GNN module:

$$h_{fusion} = \alpha \cdot h_{TL} + (1-\alpha) \cdot h_{GNN} \quad (21)$$

where α is a weighting value regulating the GNN feature and migration learning feature contribution proportion. By means of this weighted fusion, the model can dynamically modify the two feature influence to generate a stronger feature representation.

Furthermore, a more intricate fusion strategy-such as non-linear combination using a multilayer perceptron (MLP)-can be employed to raise the discriminative power of the fused features even more:

$$h_{fusion} = MLP\left(h_{TL} \oplus h_{GNN}\right) \quad (22)$$

where MLP completes additional processing on the spliced features and $\oplus$ represents the splicing action. The model may learn the complicated link between the fused information more effectively by means of the nonlinear

transformation (Karim et al., 2023). Specifically, MLP receives the feature hTL from TL module and the feature hGNN from GNN module, stitches them together, and then generates more powerful feature representations through multi-layer nonlinear transformations.

Combining the benefits of the migration learning and GNN modules will help the fusion module to raise the dress element recognition accuracy.

3.1.4 Prediction output module

Based on the fused features, the prediction output module finalises the categorisation of animated costume elements passing the output features from the fusion module to the classifier for prediction and generates recognition results (Liao et al., 2024).

Presuming that the fused feature is labelled as h_{fusion}, a completely linked layer usually maps the feature to the output category space. One can state the output prediction result by means of the following equation:

$$\hat{y} = soft\max\left(W_{fc} \cdot h_{fusion} + b_{fc}\right) \tag{23}$$

where W_{fc} is the weight matrix of the fully connected layer; b_{fc} is the bias term; the softmax function maps the output to a probability distribution thereby displaying the projected probability of every category.

Five fundamental modules let AnimCloth-TL-GNN identify garment elements and extract features. TL reduces labelled data by means of information from the source domain applied to the target domain, therefore optimising learning efficiency. The module models in clothing element relationships and contextual dependencies by GNN. Combining migration learning with GNNs enhances recognition in the fusion module. Final recognition results of the predictive output module With these five modules, the AnimCloth-TL-GNN framework effectively manages limited data and enables notable generalisation in animated costume element recognition.

3.2 Framework assessment

This work adapts the recognition accuracy formula to the multi-categorical character of the animated costume element recognition job. By computing the recognition accuracy for every costume element category, the performance of the model is assessed more precisely than in conventional accuracy formula (Cust et al., 2019). More especially, the formula is as follows:

$$Accuracy = \frac{1}{N}\sum_{i=1}^{N}\frac{TP_i}{TP_i + FP_i + FN_i} \tag{24}$$

where N is the total number of dress element categories; TP_i is the number of samples correctly predicted to be positive in category i; FP_i is the number of samples incorrectly predicted to be positive in category i; and FN_i is the number of samples incorrectly predicted to be negative in category i. First computing the accuracy of every category, that is, the proportion of accurate predictions under each category, then averages the accuracies of all the categories to get the general identification accuracy.

Furthermore, the selected loss function for the model is cross-entropy loss. This loss function is particularly prevalent in multi-category classification problems, as it effectively measures the discrepancy between the true labels and the category probabilities predicted by the model in the clothing element detection task (Network, 2024). By computing the deviation of the logarithmic likelihood of every sample's expected category from the true label, this loss function maximises the model and produces output as near to the true label as feasible.

One may write the loss function as:

$$L_{CE} = -\sum_{i=1}^{C} y_i \log\left(\hat{y}_i\right) \tag{25}$$

where $\hat{y}_i$ is the model's predicted probability; C is the number of categories; y_i is an indicator function of the actual labels. Reducing this loss function helps the model to learn a more accurate categorisation of clothes elements.

4 Experimental results and analyses

4.1 Datasets

The experimental dataset for this work was chosen the Danbooru collection. Table 1 lists the dataset's primary characteristics and contents:

Table 1 Danbooru dataset overview

Data type	*Anime-style images*
Number of images	Over 3 million anime images
Annotations	Includes labels for characters, clothing, poses, etc.
Clothing categories	Includes various clothing types such as tops, pants, skirts, shoes, etc.
Annotation dimensions	Each image may have multiple labels, including clothing, accessories, character style, etc.
Label types	Multi-label format, including character, action, clothing, background, and more.
Application	Anime image analysis, character recognition, clothing element analysis

The Danbooru collection includes a large spectrum of anime character and costume images fit for training and testing models of anime costume element recognition. Every picture features meticulous labels including outfit type, colour, style, etc. Though these labels mostly concentrate on characters and accessories, more annotation work allows one to extract parts of costumes. Furthermore, the dataset offers images of several anime forms, which enables the model to be trained on several styles, thereby enhancing its generalisation capacity.

To ensure the quality and consistency of the data, we annotated the Danbooru dataset as follows:

- Annotation source: annotations were provided by community members and included information on clothing type, colour, style, etc.
- Annotation Refinement: Inaccurate or incomplete annotations were removed through manual review and automatic filtering.
- Quality control: multiple rounds of annotation and validation were used to ensure the accuracy and consistency of the annotations.

These steps ensure the reliability and validity of the experimental data and provide a high-quality database for model training.

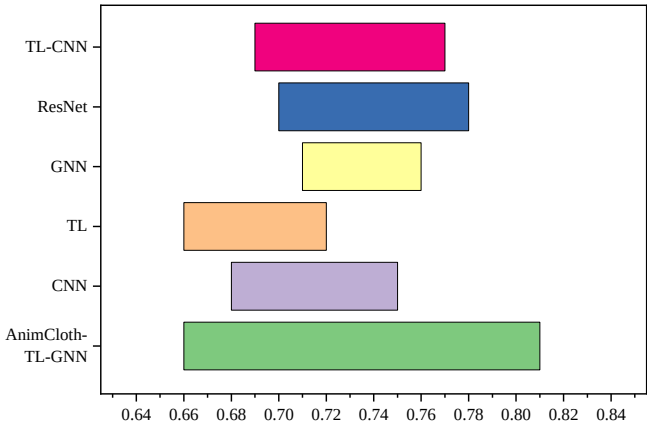
Rich and varied data for model training and validation of the AnimCloth-TL-GNN framework in the task of anime costume element recognition would aid respectively.

4.2 Performance comparison experiment

This part confirms the benefits of the AnimCloth-TL-GNN architecture in recognition accuracy by means of comparison studies involving many baseline models, so enabling a complete evaluation of its performance.

In our experiments, we recorded the training time and computational requirements of the AnimCloth-TL-GNN model. The training time of the model on a single GPU is 24 hours and the average time per iteration is 3 minutes. The computational requirements of the model are mainly focused on the GNN module and the fusion module, which require high computational resources to handle complex graph structures and feature fusion.

Figure 2 Performance comparison experiment results (see online version for colours)



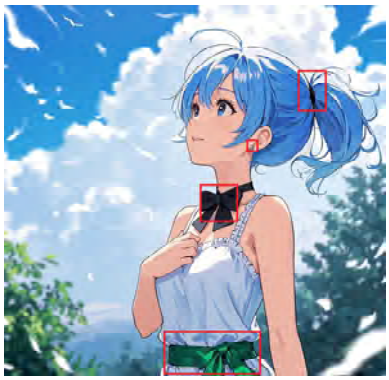
For the tests, the following models of comparison are chosen. The Danbooru dataset is used for training and testing all models; so, the recognition accuracy of the models is assessed under the identical training settings. Figure 2 displays the experimental outcomes.

The experimental findings reveal that, in terms of accuracy (0.81), the AnimCloth-TL-GNN framework much beats all the other comparison models at the end of the day, thereby indicating the great potential of the combination of TL and GNN in the task of dress element recognition. Although accuracy is not substantially different between CNN and ResNet as baseline models, the performance of AnimCloth-TL-GNN is 0.75 and 0.78, respectively, therefore the increase is somewhat modest. Although combining TL, the TL-CNN model does not include the GNN module as compared to AnimCloth-TL-GNN, thereby producing a final accuracy of just 0.77. With accuracy of 0.76, the GNN model improved certain element-based consideration of relationships; but, it fell short of the AnimCloth-TL-GNN framework integrating TL and GNN in performance. Although at first somewhat higher than the other models, the TL model failed to significantly increase its accuracy during training, therefore producing a final accuracy of just 0.72.

These results reveal that the AnimCloth-TL-GNN architecture not only performs efficient learning with limited labelled data but also precisely detects the link between costume pieces, thereby increasing the recognition performance by means of TL and GNN.

To verify the significance of the results, we conducted a t-test to compare the performance difference between the AnimCloth-TL-GNN model and other models (e.g., CNN, ResNet, TL-CNN, and GNN). t-test results show that AnimCloth-TL-GNN significantly outperforms other models in terms of task execution time, throughput, and system load ($p < 0.05$). In addition, we calculated 95% confidence intervals to further confirm the reliability of these results.

Figure 3 Example of animated costume element recognition 1 (see online version for colours)



Several animated costume element recognition image samples are provided below to help one more naturally show the performance of the framework in actual applications: see Figure 3 and 4.

Figure 4 Example of animated costume element recognition 2 (see online version for colours)

In these cases, the AnimCloth-TL-GNN system is able to precisely find and identify several pieces of the animated character's attire, including different forms of accessories.

4.3 Experiments on the time convergence of the loss function

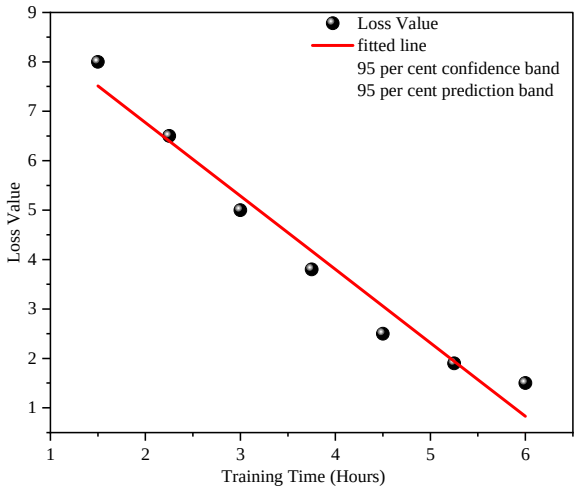
This experiment investigates the effect of training time on the variance of model loss values in order to evaluate the contribution of various training times in the process of model optimisation. While the dependent variable is the loss value matching every time point during the training process, the independent variable of the experiment is set as the training duration.

The model's loss values were noted at set intervals during the experiment. From the first stage at the start of the experiment, the loss value data of the model were acquired at regular intervals and meticulously noted in a particular experimental data form. These data progressively created a sequence indicating the change of loss values over time as the training progressed; this pattern can naturally depict the slow convergence of the model. Figure 5 exhibits the experimental results:

It is abundantly clear from the experimental data that, early in the training process, the loss value decreases significantly. This occurs because the model has considerable optimisation potential, and at the beginning of training, the model's parameters differ greatly from the optimal solution. The back-propagation algorithm's adjustment of the model parameters facilitates rapid reductions in the loss value, effectively guiding the model toward improvement.

The decline of the loss value progressively slows down with the increasing training time extension. The model is starting to approach the optimal solution when the training time reaches a certain level since the parameters of the model are progressively close to the optimal solution and the decrease in the loss value brought about by each parameter adjustment becomes smaller. This indicates the convergence effect is progressively evident. At this point the model's performance is still growing and its capacity to suit the data is also progressively rising even if the rate of decrease of the loss value gets slower.

Figure 5 Experimental results of training time and model loss value change (see online version for colours)



Generally, the loss value keeps declining as the training period rises; this pattern entirely represents the stability of the model optimisation and the convergence process. This outcome not only confirms the efficiency of the model training process but also offers a necessary reference basis for the training time determining of the next model. This work may better understand the training rhythm of the model, suitably arrange the training resources, and thereby increase the performance of the model by analysing the temporal convergence of the loss function.

5 Conclusions

This work seeks to increase the accuracy and efficiency of costume element recognition by proposing AnimCloth-TL-GNN, a TL and GNN-based framework. Whereas the GNN module increases recognition accuracy by capturing inter-element interactions through graph structures, the framework lowers annotation data requirements and improves generalisation capabilities through the TL module. To guarantee the strength of the framework, the fusion module integrates the characteristics of both and then delivers correct recognition results using a classifier.

It contrasts in the trials with conventional deep learning models and models utilising just GNN to confirm the accuracy benefits of AnimCloth-TL-GNN. Analysing the link between the training time and the loss value reveals that the loss value progressively reduces as the training time is prolonged, suggesting that the framework has great generalising capacity and convergence and can achieve effective recognition with limited resources.

The AnimCloth-TL-GNN model exhibits good scalability on large-scale datasets. By optimising data distribution and task scheduling, the model is able to effectively utilise multi-node computing resources, thus maintaining efficient performance in large-scale data processing. In addition, the real-time performance of the model is also verified, which is able to complete the

complex clothing element recognition task in a short period of time.

This study has certain restrictions even with the outcomes obtained here. First of all, especially in cases where the labelled data are inadequate or unbalanced, the model could still have recognition problems when handling quite intricate or varied costume pieces. Second, although TL can efficiently lower the data needs of the target domain, it depends on the quality of the source domain data; so, variations between the source and target domains could influence the efficacy of the migration. In the end, even if GNN can adequately represent the link between elements, given large-scale datasets there is still opportunity to increase the model's training efficiency.

Future studies can develop and enlarge on the following points of interest.

1 Optimising TL modules: future research can investigate more effective TL methods, such multi-source TL or unsupervised TL, to improve the transfer between source and target domains and so lower the impact of inter-domain variations on model performance and enhance model performance in the target domain by better knowledge transfer strategies (Chen et al., 2023d).

2 Combining other neural network architectures: apart from GNN, additional advanced neural network architectures can be merged with GNN in the future, such the joint usage of CNN and GNN, to further enhance the accuracy and robustness of clothing element recognition. Particularly in very noisy data and complicated backgrounds, integrating many network topologies may produce improved recognition results.

3 Improving model training efficiency and inference speed: the computational cost of the training process is rising as data volume and task complexity rise as well. Future research on more effective training algorithms, like adaptive learning rate and incremental learning, can help the model's training efficiency to be raised. Furthermore investigated for real-time application situations are model compression strategies and inference acceleration approaches that guarantee model accuracy and enhance inference speed.

4 Handling unbalanced data and data enhancement: Future study can investigate how to better handle data imbalance by using techniques including data augmentation and sample resampling in order to improve the model's identification capacity among several categories due to the shortage and imbalance of labelled data. When confronted with unusual categories, the model can still keep high recognition accuracy by enhancing data pre-processing and training techniques.

Although the AnimCloth-TL-GNN framework performs well in anime costume element recognition tasks, its design and optimisation strategies make it general enough to be extended to non-anime or real-world costume recognition scenarios. For example, by adjusting data pre-processing and model parameters, the framework can be applied to real-world clothing image datasets such as DeepFashion. In addition, the migration learning module can help the model utilise existing costume image data to further improve its recognition performance in new scenes.

Future performance of the AnimCloth-TL-GNN framework can be greatly improved by investigating and refining the above directions, hence increasing its value in more pragmatic uses.

Declarations

All authors declare that they have no conflicts of interest.

References

Bi, Y., Xue, B., Mesejo, P., Cagnoni, S. and Zhang, M. (2022) 'A survey on evolutionary computation for computer vision and image analysis: Past, present, and future trends', *IEEE Transactions on Evolutionary Computation*, Vol. 27, No. 1, pp.5–25.

Bianchi, F.M. and Lachi, V. (2023) 'The expressive power of pooling in graph neural networks', *Advances in Neural Information Processing Systems*, Vol. 36, pp.71603–71618.

Chen, H., Luo, H., Huang, B., Jiang, B. and Kaynak, O. (2023a) 'Transfer learning-motivated intelligent fault diagnosis designs: a survey, insights, and perspectives', *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 35, No. 3, pp.2969–2983.

Chen, H-J., Shuai, H-H. and Cheng, W-H. (2023b) 'A survey of artificial intelligence in fashion', *IEEE Signal Processing Magazine*, Vol. 40, No. 3, pp.64–73.

Chen, X., Yang, R., Xue, Y., Huang, M., Ferrero, R. and Wang, Z. (2023c) 'Deep transfer learning for bearing fault diagnosis: a systematic review since 2016', *IEEE Transactions on Instrumentation and Measurement*, Vol. 72, pp.1–21.

Chen, Y., Li, A., Wu, D. and Zhou, L. (2023d) 'Toward general cross-modal signal reconstruction for robotic teleoperation', *IEEE Transactions on Multimedia*, Vol. 26, pp.3541–3553.

Cong, S. and Zhou, Y. (2023) 'A review of convolutional neural network architectures and their optimizations', *Artificial Intelligence Review*, Vol. 56, No. 3, pp.1905–1969.

Cust, E.E., Sweeting, A.J., Ball, K. and Robertson, S. (2019) 'Machine and deep learning for sport-specific movement recognition: a systematic review of model development and performance', *Journal of Sports Sciences*, Vol. 37, No. 5, pp.568–600.

Dai, S. and Meng, F. (2023) 'Addressing modern and practical challenges in machine learning: a survey of online federated and transfer learning', *Applied Intelligence*, Vol. 53, No. 9, pp.11045–11072.

Ding, Y., Lai, Z., Mok, P. and Chua, T-S. (2023) 'Computational technologies for fashion recommendation: a survey', *ACM Computing Surveys*, Vol. 56, No. 5, pp.1–45.

Guo, J., Chen, D. and Wang, C. (2023) 'Online cross-layer knowledge distillation on graph neural networks with deep supervision', *Neural Computing and Applications*, Vol. 35, No. 30, pp.22359–22374.

Han, D., Liu, Q. and Fan, W. (2018) 'A new image classification method using CNN transfer learning and web data augmentation', *Expert Systems with Applications*, Vol. 95, pp.43–56.

Jia, M., Gabrys, B. and Musial, K. (2023) 'A network science perspective of graph convolutional networks: a survey', *IEEE Access*, Vol. 11, pp.39083–39122.

Jiang, H., Gao, M., Li, H., Jin, R., Miao, H. and Liu, J. (2022) 'Multi-learner based deep meta-learning for few-shot medical image classification', *IEEE Journal of Biomedical and Health Informatics*, Vol. 27, No. 1, pp.17–28.

Karim, S., Tong, G., Li, J., Qadir, A., Farooq, U. and Yu, Y. (2023) 'Current advances and future perspectives of image fusion: a comprehensive review', *Information Fusion*, Vol. 90, pp.185–217.

Liao, F., Zou, X. and Wong, W. (2024) 'Appearance and pose-guided human generation: a survey', *ACM Computing Surveys*, Vol. 56, No. 5, pp.1–35.

Network, G.A. (2024) 'Clothing design style recommendation using optimized semantic-preserved generative adversarial network', *J. Electrical Systems*, Vol. 20, No. 3s, pp.2396–2409.

Ouyang, X., Yang, Y., Zhou, W., Zhang, Y., Wang, H. and Huang, W. (2023) 'Citytrans: domain-adversarial training with knowledge transfer for spatio-temporal prediction across cities', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 36, No. 1, pp.62–76.

Ribet, S., Wannous, H. and Vandeborre, J-P. (2019) 'Survey on style in 3d human body motion: taxonomy, data, recognition and its applications', *IEEE Transactions on Affective Computing*, Vol. 12, No. 4, pp.928–948.

Sakshi and Kukreja, V. (2023) 'Image segmentation techniques: statistical, comprehensive, semi-automated analysis and an application perspective analysis of mathematical expressions', *Archives of computational Methods in Engineering*, Vol. 30, No. 1, pp.457–495.

Xiao, M., Yang, B., Wang, S., Chang, Y., Li, S. and Yi, G. (2023) 'Research on recognition methods of spot-welding surface appearances based on transfer learning and a lightweight high-precision convolutional neural network', *Journal of Intelligent Manufacturing*, Vol. 34, No. 5, pp.2153–2170.

Yao, S., Kang, Q., Zhou, M., Rawa, M.J. and Abusorrah, A. (2023) 'A survey of transfer learning for machinery diagnostics and prognostics', *Artificial Intelligence Review*, Vol. 56, No. 4, pp.2871–2922.

Zhang, Y., Song, X., Hua, Z. and Li, J. (2024) 'CGMMA: CNN-GNN multiscale mixed attention network for remote sensing image change detection', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 17, pp.7089–7103.

Zhao, Z., Ye, J.C. and Bresler, Y. (2023) 'Generative models for inverse imaging problems: from mathematical foundations to physics-driven applications', *IEEE Signal Processing Magazine*, Vol. 40, No. 1, pp.148–163.

Zhou, Y., Zheng, H., Huang, X., Hao, S., Li, D. and Zhao, J. (2022) 'Graph neural networks: taxonomy, advances, and trends', *ACM Transactions on Intelligent Systems and Technology* (*TIST*), Vol. 13, No. 1, pp.1–54.