International Journal of Engineering and Technology Innovation, vol. 14, no. 2, 2024, pp. 216-230

Personalized Clothing Prediction Algorithm Based on Multi-modal Feature Fusion

Rong Liu[1,2], Annie Anak Joseph[1,*], Miaomiao Xin[2], Hongyan Zang[2],
Wanzhen Wang[2], Shengqun Zhang[2]

[1]Faculty of Engineering, Universiti Malaysia Sarawak, Kota Samarahan, Sarawak, Malaysia

[2]Computer and Information Engineering, Qilu Institute of Technology, Jinan, China

Received 18 February 2024; received in revised form 05 March 2024; accepted 06 March 2024

DOI: https://doi.org/10.46604/ijeti.2024.13394

Abstract

With the popularization of information technology and the improvement of material living standards, fashion consumers are faced with the daunting challenge of making informed choices from massive amounts of data. This study aims to propose deep learning technology and sales data to analyze the personalized preference characteristics of fashion consumers and predict fashion clothing categories, thus empowering consumers to make well-informed decisions. The Visuelle's dataset includes 5,355 apparel products and 45 MB of sales data, and it encompasses image data, text attributes, and time series data. The paper proposes a novel 1DCNN-2DCNN deep convolutional neural network model for the multi-modal fusion of clothing images and sales text data. The experimental findings exhibit the remarkable performance of the proposed model, with accuracy, recall, F1 score, macro average, and weighted average metrics achieving 99.59%, 99.60%, 98.01%, 98.04%, and 98.00%, respectively. Analysis of four hybrid models highlights the superiority of this model in addressing personalized preferences.

Keywords: fashion consumers, image, text data, personalized, multi-modal fusion

1. Introduction

Personalized fashion classification forecasting technology has garnered considerable attention in the e-commerce and fashion industries. Employing deep learning technology to tackle this challenge is a crucial issue that demands attention. Previous studies have delved into the challenges and potential solutions in this field. For instance, Wu and Zhu [1] proposed deep learning technology and the anatomically constrained neural networks (ACNN) method to enhance the accuracy and efficiency of garment product shape recognition, thus introducing innovations and breakthroughs to fashion prediction. Yuan et al. [2] designed and implemented a clothing matching and recommendation system based on clothing pictures and customer historical behavior data using deep learning technology and data mining to meet consumer demand and boost sales. The target detection technology and the deep residual network (ResNet) extract comprehensive clothing features, addressing the issue of interfering factors in clothing image recognition through multi-depth feature fusion [3]. This method fully leverages global, primary, and local area attributes, directing the recognition process towards the clothing itself, thereby significantly improving the accuracy rate of clothing image recognition.

The multi-feature fusion algorithm is applied to accurately retrieve multi-scale clothing images [4]. Retrieval accuracy is considerably enhanced by thoroughly addressing interference factors in clothing image recognition through the comprehensive exploration of global-to-local multi-scale features of clothing images in conjunction with deep learning and traditional feature

* Corresponding author. E-mail address: jannie@unimas.my

English language proofreader: Yen-Chun Hsieh

extraction methods. The algorithm offers an excellent and workable approach for apparel picture retrieval by combining multi-scale convolutional neural network (CNN) features with conventional characteristics to optimize the ordering of search results. This study proposes a CNN model for identifying categories of clothing design, enhancing classification accuracy by effectively learning image attributes [5]. The model surpasses the performance of traditional hand-designed feature coding methods and existing CNN models in category recognition of clothing design. These findings suggest that deep learning and multi-feature fusion algorithms are well-positioned to enhance the accuracy of personalized fashion classification predictions significantly. Consequently, to address the challenges in customized clothing forecasting, this study proposes introducing a multi-modal fusion algorithm to meet the market demand for customized fashion classification forecasting.

In personalized clothing classification prediction, current research grapples with two primary challenges. Firstly, there is a need to effectively represent and consolidate multi-modal data in a manner that exploits the complementary nature of diverse modalities. This entails developing methodologies for efficiently fusing information from distinct modalities to enable models to understand clothing characteristics better. For instance, while text descriptions provide insights into clothing seasons and types, images offer detailed visual characteristics and labels. Integrating these sources provides a more comprehensive garment description, enhancing personalized classification accuracy.

Secondly, research must address the challenge of data transformation between different modalities. In personalized clothing classification prediction, converting text descriptions into image features or vice versa is essential. This study aims to tackle these challenges by developing practical algorithms capable of learning mapping relationships between modalities and seamlessly integrating information. This transformation process considers the heterogeneity of data and the subjective relationships between modalities, presenting a formidable challenge. However, overcoming this obstacle can lead to more effective utilization of multi-modal data, thereby enhancing the comprehensiveness of personalized classification. Building upon these challenges, the optimization of parameter classification is proposed, with the subsequent steps constructed for this purpose.

The significant contributions of this paper include:

(1) Propose a multi-modal weighted combination method primarily utilizing visual information to achieve a more comprehensive representation of fashion items through weighted fusion of multi-modal features encompassing graphical and textual data.

(2) Propose a personalized clothing prediction algorithm based on multi-modal feature fusion, comprising four main components: feature extraction, feature joint representation, feature fusion, and customized clothing category prediction.

(3) A hybrid model that combines a 1D convolutional neural network (1DCNN) and a 2D convolutional neural network (2DCNN) is introduced to enhance the performance of the fashion prediction model. The proposed model is compared with the same dataset's mainstream models such as ResNet and the temporal convolutional network (TCN).

The structure of the remainder of the paper is as follows. Section 2 delineates the preceding endeavors undertaken by researchers within this domain. Section 3 outlines the pertinent methodologies proposed to tackle the challenges of multi-modal personalized fusion research. The outcomes of experimental and comparative analyses are elucidated in Section 4. Section 5 encapsulates the empirical findings and underscores the contributions of this paper.

2. Related Worked

As societal values evolve and individual preferences become increasingly nuanced, the imperative of personalization grows more pronounced. The interactive bisection algorithm proposed by certain studies achieves attribute differentiation through a bottom-up information genetic algorithm (GA) method. It defines an average classification fuzzy subset to facilitate

a comprehensive decision-making process [6]. The distinct style of traditional Chinese ethnic costumes has emerged as a fashion expression. In the realm of personalized recommendation algorithms, an enhanced collaborative filtering algorithm has been introduced to address issues impacting the accuracy of joint filtering recommendations. By assigning weights to clothing classification attributes and amalgamating clothing category preference similarity with consumer feature similarity, the accuracy of joint filtering recommendations is enhanced, promoting holistic similarity [7].

Numerous researchers are dedicated to addressing the challenge of personalized clothing category prediction, focusing on aligning the compatibility of textual and visual features in the embedding space, particularly emphasizing visual feature compatibility [8] or treating visual features as sequences [9]. The classification of fashion apparel and items encounters the obstacle of integrating category or sub-category classification with diverse fashion item attribute prediction within a concise multi-task learning framework. Certain studies have proposed the fashion sub-category and attribute prediction (FSCAP) model, achieving an accuracy rate of 94% [10].

However, existing research exhibits certain limitations in fully leveraging the complementarity and compatibility of modal fusion. A new framework for clothing matching based on item compatibility is proposed [11]. Instead of relying on visual features, it exclusively utilizes textual descriptions, particularly title sentences, to construct basic features. Feature embeddings of title sentences are generated using long short-term memory (LSTM) networks. These embeddings are subsequently integrated into a style compatibility space characterized by a compatibility matrix. The framework is evaluated on three large clothing datasets to demonstrate its effectiveness compared to baseline approaches. A personalized clothing compatibility prediction model is proposed [12], which encodes users' clothing preference visual features and measures users' attention and clothing compatibility in different regions. Finally, the two modules are jointly optimized to enhance the model's prediction ability.

Nonetheless, there is no mention of the comprehensiveness of multi-modal fusion, specifically regarding the model's effective integration and balance of text and visual information [13]. This paper presents a novel approach to address the challenges mentioned earlier: a multi-modal weighted combination method. This method primarily relies on visual information, aiming to provide a more comprehensive representation of fashion items through the weighted fusion of multi-modal features, which include both visual and textual information [14]. In addition, this paper proposes a personalized clothing prediction algorithm based on multi-modal feature fusion [15-16]. This algorithm consists of four main components. The feature extraction module employs 2D convolution for visual feature extraction and 1D convolution for textual feature extraction. Its objective is to capture essential information embedded within images and descriptions.

3. Methodology

This section details the personalized clothing prediction algorithm based on multi-modal fusion, along with related loss functions and evaluation indicators. Elaborates on the dataset used, including Structured image data and unstructured text data. The algorithm comprises four main components: feature extraction, joint representation, fusion, and prediction. Specifically, a 2D model extracts image features, a 1D model extracts text features, and these features are fused before the final prediction. Using evaluation metrics like accuracy, precision, recall, and F1 score demonstrates the effectiveness of the personalized clothing recommendation algorithm.

3.1. Dataset

The study utilized sales records from the Italian fast fashion company Nunalie from October 2016 to December 2019, as its dataset [17]. This dataset encompasses sales data for 5,355 items, providing extensive multi-modal information, including product descriptions, images, and prices. The dataset's diversity enhances the comprehensive analysis and prediction of fashion product sales performance.

The ratio of training, test, and validation sets in the dataset is 8:1:1, with image data divided into six folders. They are AI17, AI18, AI19, PE17, PE18, and PE19. These folders contain 864, 999, 1,084, 955, 776, and 899 fashion clothing pictures. The text data consists of 24 columns of sales data, including clothing type, color, image code, and season.

The data collection method entails organizing image data within a folder containing the "images" directory and a comma-separated values (CSV) file listing the image file name. Textual data is stored within the columns of this CSV file. During data preprocessing, information is extracted from the CSV file, irrelevant columns such as "image path," "release date," and "fabric" are discarded, and the remaining features are labeled accordingly.

Validation sets are employed throughout model training to monitor and adjust model performance. The model checkpoint callback function saves the model parameters that exhibit the best performance on the validation set. Upon completion of training, the best-performing model is loaded and evaluated using the test dataset. Confusion matrices and classification reports are generated to assess the model's performance and generalization ability. Within the dataset, picture category labels are replaced with numerical labels; for instance, "AI19" is replaced with "2" and "PE19" with "5". The algorithm implementation utilizes lambda and map functions.

(1) Image data

Each product is associated with an RGB image ranging in resolution from 256 to 1,193 in width and 256 to 1,172 in height, with median values of 575 and 722, respectively. These images were captured in a controlled environment to ensure color accuracy and minimize potential prediction bias. Each image depicts a clothing item against a white background and includes a binary foreground mask to aid in distinguishing the item from the background. Fig. 1 illustrates clothing images. Preprocessing of the image data involves loading image files, resizing, and normalization. Subsequently, the image data is stored as NumPy arrays and loaded into variables to train the image and test the image. Each image array has a shape of (64, 64, 3), representing the width, height, and number of RGB channels.

Fig. 1 Examples of images per category

(2) Text data

Each product is associated with multiple labels extracted by the Nunalie team after validation. Products are sold during a specific season and released on a particular day. This temporal information is recorded as a text string. During training, the code snippet initially removes image paths, release dates, and structural details from the training and test dataset. The non-numeric data is labeled and converted into an array format suitable for model input.

3.2. Personalized clothing prediction algorithm based on multi-modal feature fusion

In response to the limitation of traditional single-factor models in fully utilizing time series-related information, resulting in poor accuracy and reliability in time series prediction, a time series prediction model based on multi-modal information fusion is proposed. This model aims to fuse multiple textual and numerical data in modal data for prediction. Proposed visual recognition involves supervised learning of human-annotated image label data and language-image contrastive learning of network-extracted image-text pairs [18]. A novel formula is introduced by combining image labels and text into a common image-text-label space. This article integrates the perspectives proposed in these two studies to combine time series table data and image label data for predicting fashion clothing categories, aiming to recommend fashion categories to consumers.

In this article, the two types of data, image-label and image-text, can be represented in a unified form: the (image, text, label) triplet [19]. Specifically, for image-label data, the text describes the category name corresponding to each label, while the label indicates the discrete label for each category. In contrast, for image-text data, the text denotes the textual description of each image, and the label pertains to each pair of matching images. It's important to note that the text pairs are all unique. By merging these two data types, as depicted in Fig. 2, a matrix can be formed where the filled areas represent positive samples and the remaining regions represent negative samples. In image-label data, the images and texts corresponding to the same categories constitute positive samples, while in image-text data, the positive samples are represented by diagonal lines.

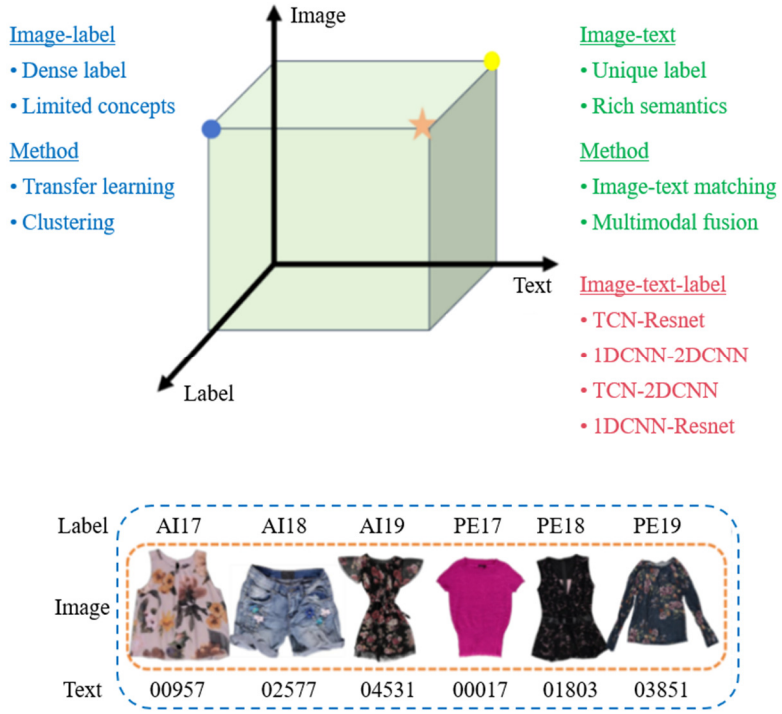

Fig. 2 Annotation labels match images and text

The personalized clothing prediction algorithm utilizes multi-modal fusion technology to predict clothing categories by integrating images and text. Initially, it conducts a multi-modal intelligent analysis of clothing images, extracting multi-dimensional layout information like texture, fabric, and garment profile. This information is then mapped to the visual space to enhance understanding of the garment's appearance and style [20].

During the multi-modal fusion process, the algorithm incorporates textual data such as user preferences, purchase history, and season to enrich the model's training data further. This integration captures the visual attributes of the clothing and reflects user preferences and trends. Moreover, advanced data processing techniques like pre-training models expedite model training and enhance generalization capabilities. The personalized clothing prediction algorithm achieves high-precision category prediction through multi-modal fusion technology, integrating information from diverse sources such as images, text, and user characteristics.

3.3. Triplets in cross-entropy loss function

Let the characteristics of the sample be $\vec{x}$ tagged with y, let the classifier weight be $W = [\vec{w}_1^T, \vec{w}_2^T, \cdots, \vec{w}_c^T]$. Among them, represents the vertical connection of vectors, and c is the number of categories of the classifier. The one hot encoding of y is $\overrightarrow{e_y} = [1\{y = 1\}, \cdots, 1\{y = c\}]$, is a c-dimensional vector, and $1\{.\}$ is the indicator function. $\vec{x}$ the output after classifier and normalization is $\vec{q} = soft\ max(w\vec{x})$, It is also a c-dimensional vector. The cross-entropy loss function of the sample is:

$$-\left\langle \vec{e}_y, \log^{\circ} \vec{q} \right\rangle = \log\left[1 + \sum_{j=1, j\neq y}^{e} \exp\left(\vec{w}_j^T \vec{x} - \vec{w}_y^T \vec{x}\right)\right] \tag{1}$$

Suppose the classifier weight is regarded as the feature of the label. In that case, the current sample feature $\vec{x}$ and the current sample label feature $\vec{w}_j$ form a positive sample pair and current sample characteristics $\vec{x}$ and non-current label features $\vec{w}_j (j = 1, \cdots, C\ and\ j \neq y)$ form an opposing sample pair. The formula shows that triples are hidden in the cross-entropy loss function (including one positive sample pair and $c - 1$ opposing sample pairs).

3.4. Evaluation index

The Confusion Matrix, also called an error matrix, is a valuable tool for evaluating the performance of models in classification problems within supervised learning. In this article, the confusion matrix represents the disparities between the algorithm model's classification outcomes and the actual scenario, primarily applied to address binary classification problems. The confusion matrix is shown in Table 1, which provides a detailed explanation and analysis. True positive (TP) denotes the count of samples accurately predicted as positive by the model. False positive (FP) signifies the count of samples erroneously classified as positive by the model, despite being negative. True negative (TN) represents the count of samples accurately predicted by the model as the negative class. False negative (FN) indicates the count of samples inaccurately classified as negative by the model, although they are positive.

Table 1 Confusion matrix

Class	Predict positive class	Predict negative class
Actual positive class	TP	FN
Actual negative class	FP	TN

Through the confusion matrix, a variety of performance indicators can be calculated to evaluate the performance of the classification model, such as accuracy, precision, recall or true positive rate (TPR), false positive rate (FPR) F1 score, etc. The advantage of the confusion matrix is that it not only provides the number of misclassifications but also illustrates which categories have the most prediction errors, which is very helpful for improving classification algorithms and model tuning.

Accuracy is the most intuitive performance indicator, and its calculation formula is:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

Precision focuses on how many of the samples predicted as positive are positive. The calculation formula is:

$$Precision = \frac{TP + TN}{TP + FP} \tag{3}$$

Recall or true rate (TPR) focuses on how many of all TP samples are predicted as positive by the model. The calculation formula is:

$$Recall = \frac{TP + TN}{TP + FN} \tag{4}$$

The F1 score is the harmonic mean of precision and recall and is calculated as:

$$F1\ score = \frac{2\times(Precicion\times Recall)}{Precicion + Recall} \tag{5}$$

Macro avg, which is the average precision of all classes, where n represents the class, is calculated by:

$$Macro\ avg = \frac{1}{n}\sum_{i=1}^{n} P_i \tag{6}$$

The weighted average is an improvement on the macro average, taking into account the proportion of the number of samples of each category in the total sample:

$$Weighted\ avg = \sum_{i} w_i P_i \tag{7}$$

When combining one-dimensional convolution with two-dimensional convolution, consider the case of processing input images with multiple channels. Assume that the input image is X, and its size is $H \times W \times C$, where H represents the height of the image, W represents the width of the image, and C represents the number of channels of the input image. The convolution kernel is W, and its size is $H \times W \times C$, where M represents the height of the convolution kernel, N represents the width of the convolution kernel, and C represents the number of channels in the input image. Then, the calculation formula for the combination of one-dimensional and two-dimensional convolution is as follows, $(X \times W)[i,j]$ representing the pixel value of the output image $i = 1,2,3,\cdots,H - M + 1, J = 1,2,\cdots,W - N + 1$.

$$(W \times W)[i,j] = \sum_{u=1}^{M}\sum_{v=1}^{N}\sum_{c=1}^{C} X[i+u-1, j+v-1, c]\cdot W[u,v,c] \tag{8}$$

In this study, factors evaluated for the model's running speed included computational effort floating point operations (FLOPs), runtime frames per second (FPS), and the number of parameters (Params). To comprehensively assess performance, the experiment adopted a deep learning architecture with convolutional, pooling, and fully connected layers. Layers were added with different numbers of nodes and layers. Rectified linear unit (ReLU) activation functions and the Adam optimizer enhanced generalization and stability. An appropriate initial learning rate was set, and Dropout layers were added to mitigate overfitting and improve regularization. ReLU activation functions and the Adam optimizer were used to enhance generalization and stability. An appropriate initial learning rate was set, and dropout layers were added to mitigate overfitting and improve regularization.

In data preprocessing, input data is encoded, features are selected, and image data is scaled. Additionally, pre-trained model parameters are loaded. Initially, the model's accuracy is assessed, and once a certain accuracy threshold is reached, a speed index is introduced to evaluate performance. Despite utilizing CPU-only training without GPU support, hardware resource demands for forward propagation are still considered. Moreover, parameters post-training reflect the model's memory footprint. As convolutional operations entail substantial computation during forward propagation, FLOPs are employed to calculate FLOPs per second. This enables fine-tuning of the model structure and comparing computational complexity among different models. C_i, C_0 represents input and output channels, K represents the size of the convolution kernel, and H and W are the size of the output feature map.

$$FLOPs = \left(2\times C_i \times K^2 - 1\right)\times H \times W \times C_0 \tag{9}$$

$$\left(2\times C_i \times K^2 - 1\right) = \left(C_i \times K^2\right) + \left(C_i \times K^2 - 1\right) \tag{10}$$

While FLOPs estimate computational complexity, the execution time can be influenced by factors like hardware performance and data size. Therefore, a time function was incorporated to calculate training time, representing training speed and providing a direct understanding of the model's performance on CPU hardware.

4. Results and Discussion

This paper comprehensively evaluates and compares the four proposed models (TCN-ResNet, 1DCNN-ResNet, TCN-2DCNN, and 1DCNN-2DCNN). The results show that the 1DCNN-ResNet and 1DCNN-2DCNN models show the best performance. In this section, the structure and construction process of these models are discussed in detail, and their advantages, limitations, and directions for future improvements are pointed out to ensure the validity and applicability of the models.

4.1. Model analysis and construction

The model analysis section comprehensively evaluates and compares four different models. First, a series of performance evaluation metrics, including accuracy, precision, recall, and F1 score, will be used to measure the performance of each model. Second, the experimental design, including each model's selection and tuning process of hyperparameters, is described below. Then, the four models will be compared in detail to analyze their advantages and disadvantages. Through in-depth analysis of experimental results, explore the performance of each model under specific conditions, as well as its stability and generalization ability. Finally, visualization tools such as charts and curves will be used to visually demonstrate the performance and behavior of the models, facilitating a clearer understanding of the differences, strengths, and weaknesses between them. The steps of model construction are illustrated in Fig. 3.

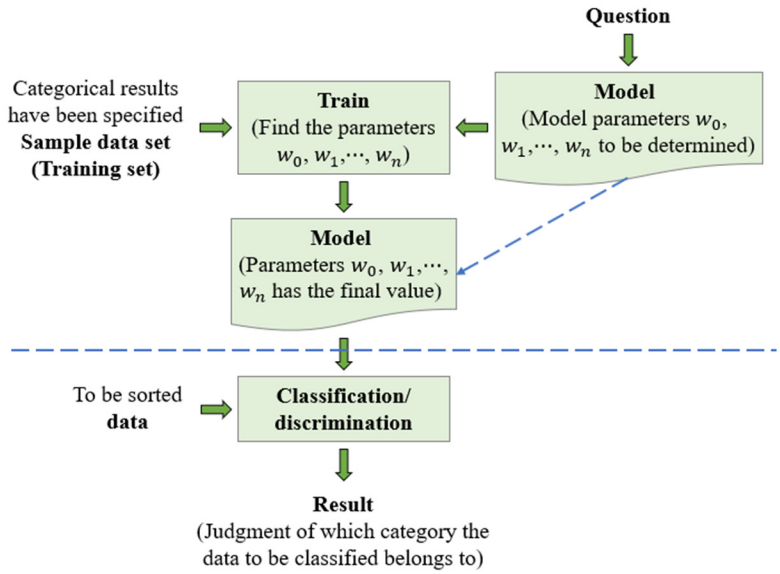

Fig. 3 Model construction steps

4.2. TCN-ResNet model

The application of TCN in sequence modeling tasks has gradually garnered attention in recent years. Compared with previously commonly used recurrent neural networks (RNNs) such as LSTM and GRU, TCN has demonstrated advantages in processing sequence data more efficiently and even outperforming them in specific tasks. As the foundation of TCN, CNN can extract low/medium/high-level features. Increasing the number of network layers enables the extraction of richer features at different levels. Additionally, deeper network layers yield more abstract extracted features and richer semantic information.

However, increasing the network depth may lead to vanishing or exploding gradient problems. The TCN part utilizes three residual block (ResBlock) modules, each containing a convolutional layer and residual connection. The network structure is constructed by adjusting parameters such as the number of filters, convolutional kernel size, and dilation rate. With filter numbers set at 32 and 16 and a fixed kernel size of 3, the dilation rate is incrementally adjusted to expand the receptive field. Processing time series data, the TCN integrates with the ResNet segment handling image data, connecting the two via a Concatenate layer, ultimately generating predictions using a fully connected layer with Softmax activation.

The ResNet architecture has inspired image classification tasks, where the ResNet design is realized using convolution blocks (ConvBlock) and identity blocks (IdentityBlock). These components integrate convolutional layers, batch normalization, and skip connections (residual connections) to process image input data. The TCN part is implemented through the ResBlock function, which constructs a ResBlock for time series data. These ResBlocks are applied to time series input data (inputs2), enabling the model to extract features from time series data effectively. By combining with the design concept of ResNet, the entire model can better handle the joint modeling of images and time series data tasks. Based on the TCN-ResNet model shown in Fig. 4, image and time series data are used for personalized clothing classification prediction.

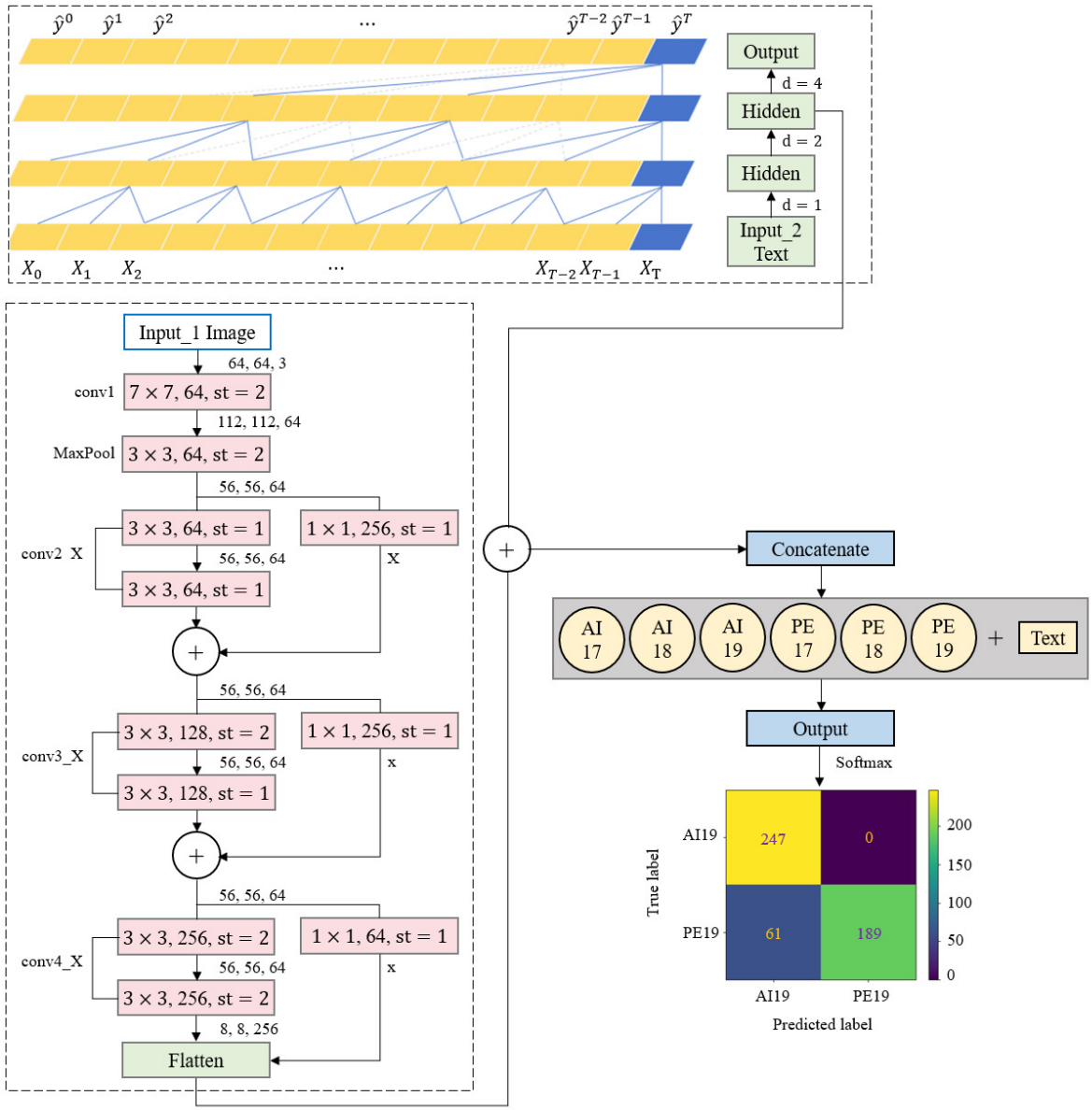

Fig. 4 TCN-ResNet model

The CSV files of the training set and test set were loaded using the Pandas library, and the data were preprocessed and labeled. Subsequently, a multi-input and multi-output deep learning model was constructed using the Keras functional API. The image data was processed using the ResNet50V2 pre-trained network and custom ConvBlock, while the time series data was processed using the TCN. The pre-trained parameter was used in Keras to load the ResNet50V2 model along with its pre-trained parameters. Keras provides API to load both the architecture and weights of the ResNet50V2 model. Once loaded, the model is trained using the model. Fit () method by passing in the training data and labels. Parameters such as the number of epochs, batch size, and validation data are set for training. The model checkpoint callback function is also employed to save the models that demonstrate the best performance on the validation set.

4.3. 1DCNN-ResNet model

This model combines the dimensional model (1DCNN) as shown in Fig. 5 and the ResNet architecture for personalized clothing classification prediction tasks. The model structure of 1DCNN-ResNet is shown in Fig. 6. The training and test datasets are initially loaded by reading CSV files, followed by data preprocessing steps, including label encoding and reshaping operations. Then, ResNet's convolution block and identity block functions are defined, and a deep learning model with two input branches is constructed.

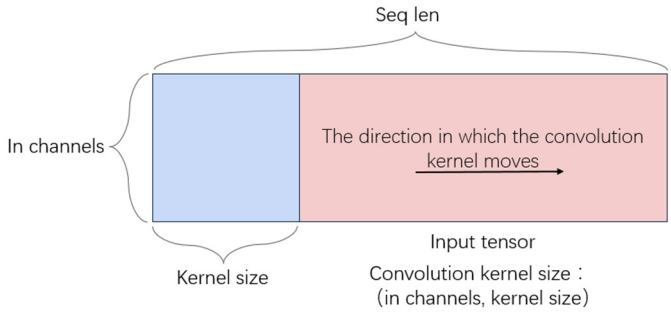

Fig. 5 Computation of one-dimensional convolution

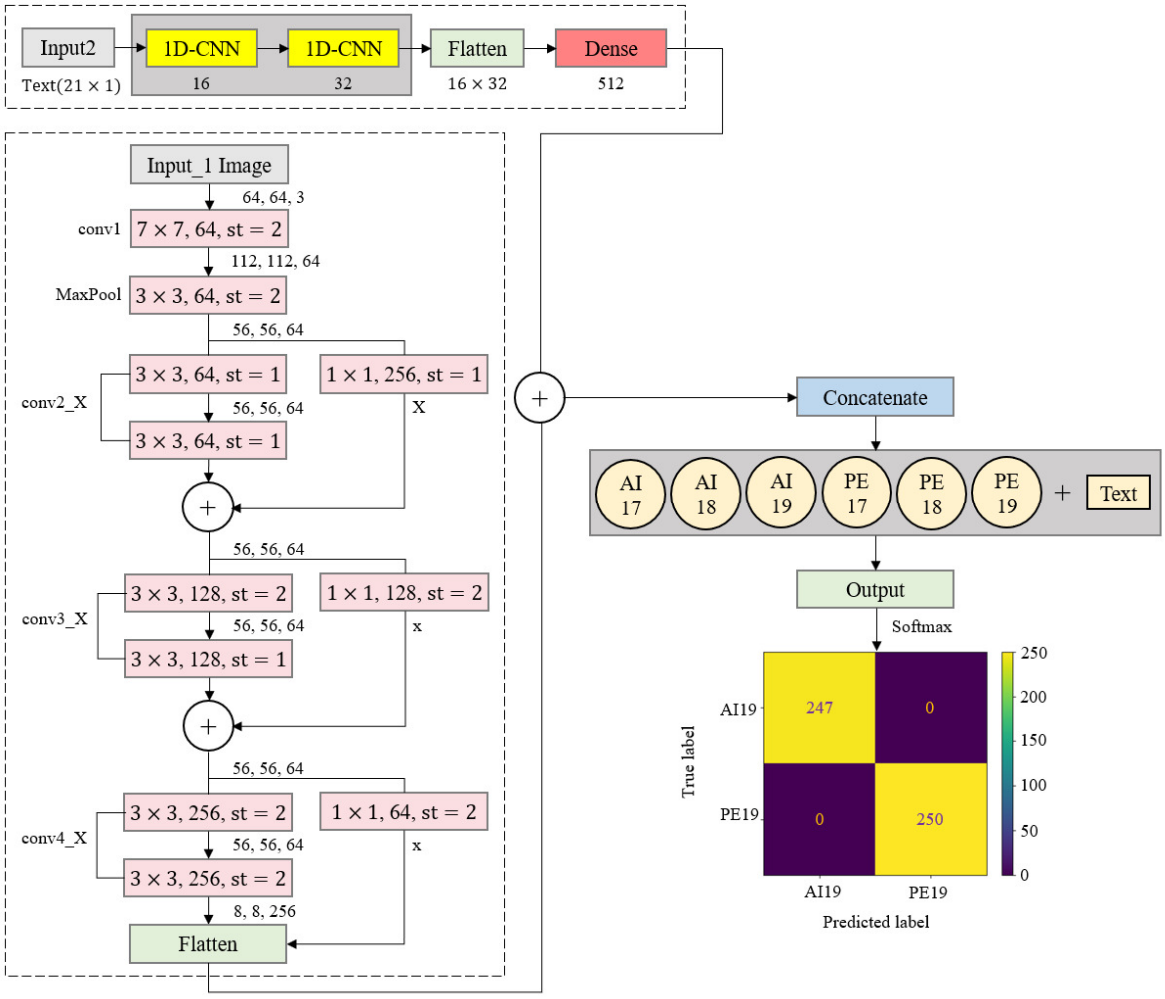

Fig. 6 1 DCNN-ResNet Model

One input branch processes image data, extracting features through a sequence of convolution and pooling operations, and ultimately outputs feature vectors via a fully connected layer. The other input branch handles text data, generating features after passing through one-dimensional convolution and fully connected layers. Subsequently, the feature vectors from both branches are concatenated, and the final classification result is produced through a fully connected layer. The model checkpoint

callback function is employed throughout the training process to save the best model, while loss and accuracy curves are plotted to monitor the training progress. Ultimately, the best model is loaded to make predictions, and the confusion matrix and classification report are generated to assess the model's performance. The research pair is shown in Table 2. Comparing the TCN-ResNet model and the 1DCNN-ResNet model, it is found that the prediction accuracy of the 1DCNN-ResNet model is significantly improved compared to the TCN-ResNet model, but the speed is slower.

Table 2 Comparison between TCN-ResNet model and 1DCNN-ResNet model

Model	Category	Precision	Recall	Model (F1-score)	Support	Macro avg	Weighted avg	Confusion matrix	Model accuracy	Prediction class	Repaid
TCN-Resnet	2	1.0000	0.8901	0.8901	247	0.9010	0.8780	0.8756	0.8773	497	4.952s
	5	1.0000	0.7560	0.8610	250	0.9016	0.8773	0.8755			
1DCNN-Resnet	2	1	1	1	247	1	1	1	1	497	11.6s
	5	1	1	1	250	1	1	1			

4.4. TCN-2DCNN model

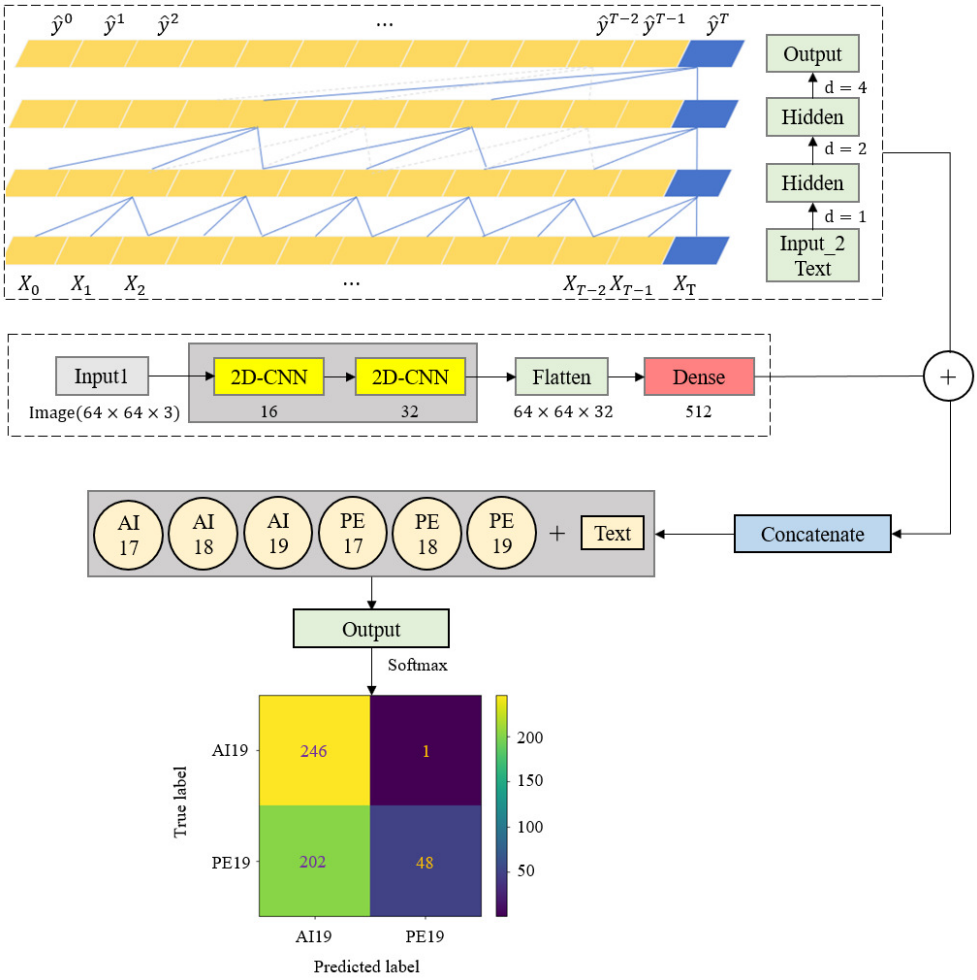

Fig. 7 TCN-2DCNN model

In this model, the ResBlock structure addresses the challenges of gradient vanishing and exploding gradients during deep network training. The ResBlock function delineates the composition of the ResBlock, comprising two convolutional layers with an identity mapping interposed between them. This residual connection is established by directly adding the input to the output of the convolutional layer. Such architecture enables the model to learn residual information, facilitating smoother training of the deep network. Within each ResBlock, a Conv1D convolutional layer is initially applied to extract features, followed by a ReLU activation function for nonlinear transformation. Subsequently, another convolutional layer is applied to refine the features further. Finally, the output of the second convolutional layer is combined with the input and undergoes

additional processing via the activation function to yield the ResBlock's output. This design strategy aims to ease the training process of deep learning models, optimize parameter utilization, and enhance feature representation, thereby bolstering the model's performance and generalization capabilities.

The algorithm is based on the TCN-2DCNN model, as shown in Fig. 7. Two input methods are used: image data and time series data. The image data undergoes processing via a series of convolutional and fully connected layers, while the temporal data is handled through a set of ResBlocks. The features from these two data streams are fused, and predictions are generated through fully connected layers with Softmax activation functions. The validation set is utilized during model training, and the best model is saved after each epoch. Subsequently, the best-saved model is loaded to make predictions on the test set, and the model performance is assessed using the confusion matrix and classification report. The experiment compared the TCN-ResNet model with the TCN-2DCNN model and found that the TCN-ResNet model showed lower prediction accuracy and slower processing speed than the TCN-2DCNN model. The experimental results are shown in Table 3.

Table 3 Comparison between TCN-ResNet model and TCN-2DCNN model

Model	Category	Precision	Recall	Model (F1-score)	Support	Macro avg	Weighted avg	Confusion matrix	Model accuracy	Prediction class	Repaid
TCN-Resnet	2	0.8019	1.0000	0.8901	247	0.9010	0.8780	0.8756	0.8773	497	4.952s
	5	1.0000	0.7560	0.8610	250	0.9016	0.8773	0.8755			
TCN-2DCNN	2	0.5491	0.9960	0.7079	247	247	0.7643	0.5940	0.5915	497	5.03s
	5	0.9796	0.1920	0.3211	250	250	0.7656	0.5915			

4.5. 1DCNN-2DCNN model

First, the data from the training and test sets are preprocessed and labeled. The 1DCNN-2DCNN model consists of two inputs: image data and textual time series data. Image data undergoes processing through convolutional layers and fully connected layers, while temporal data is processed through one-dimensional convolutional layers and fully connected layers. The features extracted from these two data streams are merged, and predictions are generated through fully connected layers and Softmax activation functions. Finally, confusion matrices and classification reports evaluate the model's performance.

During execution, the time history callback function is applied to record the time taken for each iteration. In terms of speed analysis, the 1DCNN-2DCNN model demonstrates superior performance. To verify whether the model exhibits overfitting, both the validation loss (loss on the validation set) and loss (loss on the training set) are monitored during training. The experimental running process is shown in Fig. 8 and Fig. 9.

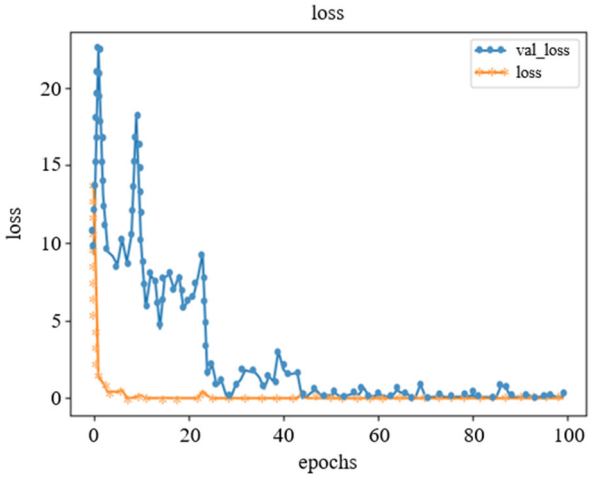

Fig. 8 Validation loss and loss

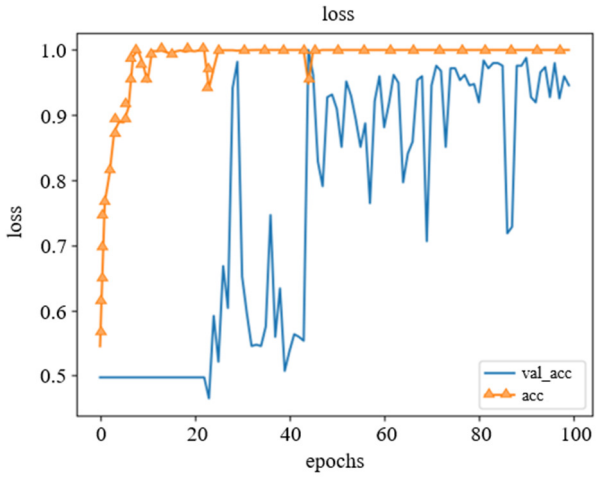

Fig. 9 Validation loss and accuracy

The experiment reveals that both the loss and validation loss consistently decrease, with a small gap between them, indicating well-trained and generalizable models. The model is shown in Fig. 10. Consequently, accuracy and validation accuracy are utilized to evaluate the model's classification performance. It is found that when epochs are 100 and batch size

is 32, the accuracy and validation accuracy of the model remain close and relatively stable, which proves that the model has good performance and generalization ability. The experiments are shown in Table 4. Comparing the TCN-ResNet model and the 1DCNN-2DCNN model, it is found that the 1DCNN-2DCNN model has higher prediction accuracy and faster speed than the TCN-ResNet model.

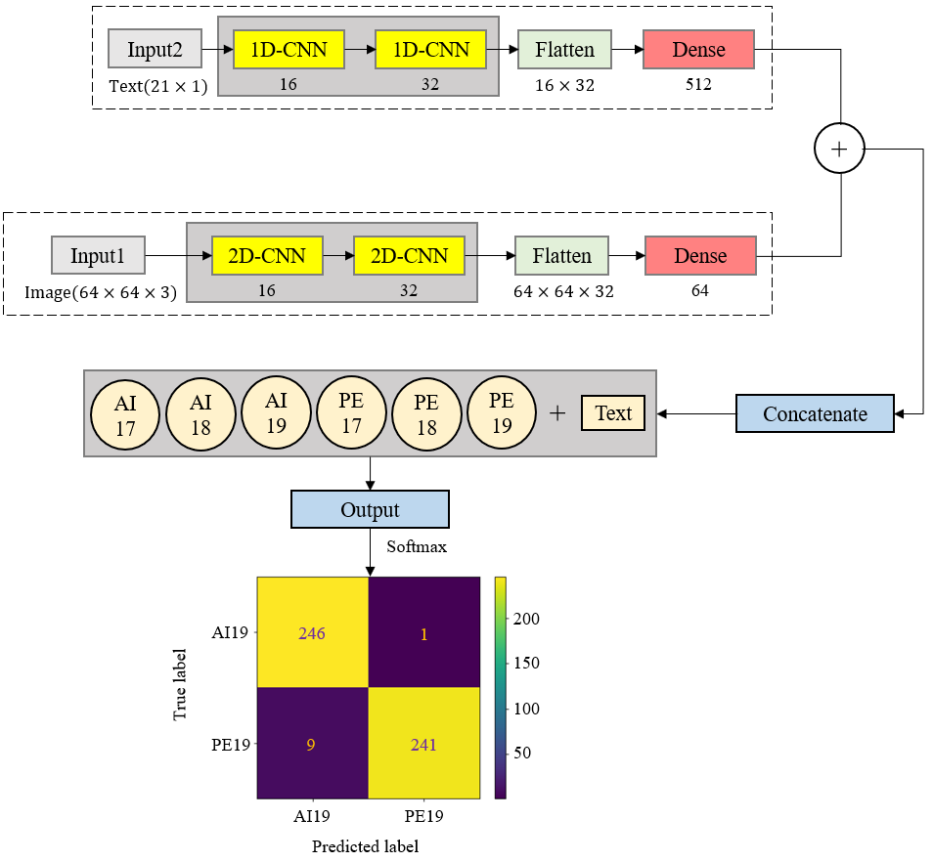

Fig. 10 1DCNN-2DCNN model

Table 4 Comparison between the TCN-ResNet model and 1DCNN-2DCNN model

Model	Category	Precision	Recall	Model (F1-score)	Support	Macro avg	Weighted avg	Confusion matrix	Model accuracy	Prediction class	Repaid
TCN-Resnet	2	0.8019	1.0000	0.8901	247	0.9010	0.8780	0.8756	0.8773	497	4.952s
	5	1.0000	0.7560	0.8610	250	0.9016	0.8773	0.8755			
1DCNN-2DCNN	2	0.9647	0.9960	0.9801	247	0.9803	0.9800	0.9801	0.9799	497	4.90s
	5	0.9959	0.9640	0.9797	250	0.9804	0.9799	0.9797			

4.6. Repeated K-fold cross-validation

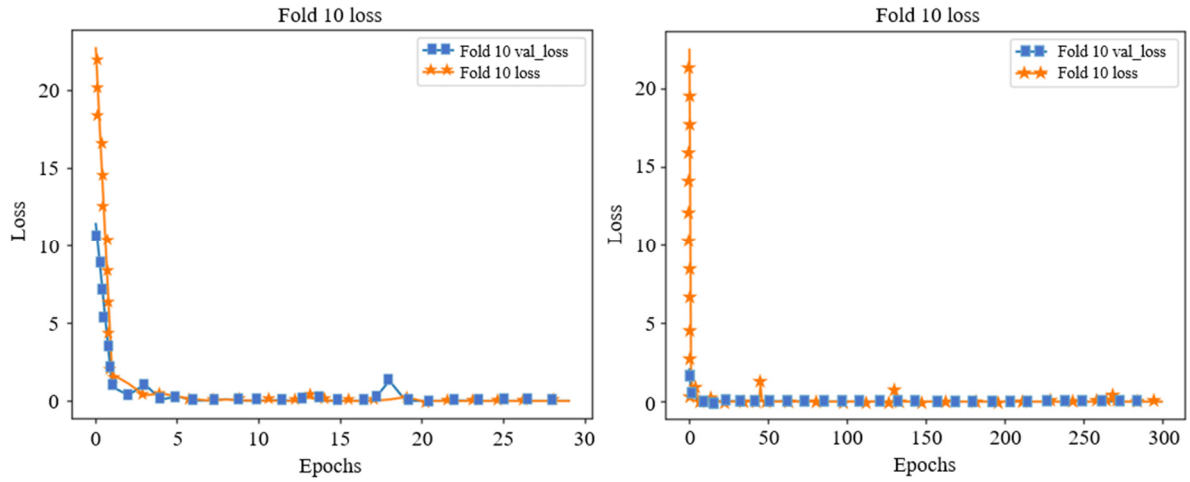

Fig. 11 Repeated cross-validation 30 and 300 epochs

In this study, duplicate 10-fold cross-validation was used, with each fold containing ten subsets and repeated twice. The model is trained and validated on current fold data. The training results for each fold are then saved, and a fold result chart is printed, as shown in Fig. 11. The randomness of data segmentation, the randomness of model initialization, and the imbalance of data samples should all be taken into account when improving model performance. These factors help to train the model with different initial parameters in each repetition, resulting in different feature representations. In addition, feature engineering, model selection, hyperparameter tuning, data preprocessing, and training strategies may significantly impact model performance. The model's accuracy increases to 98.59% when all possible factors are considered, as illustrated in Fig. 12. The results after adding repeated verification are shown in Table 5.

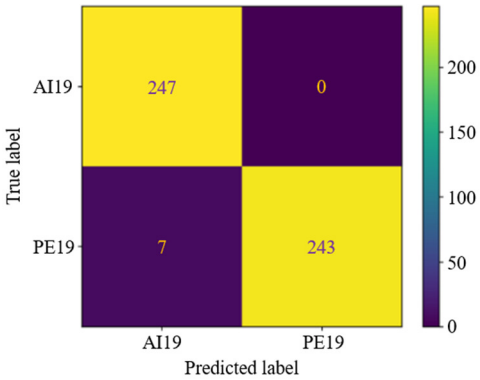

Fig. 12 Model improvement effect

Table 5 Changes after adding repeated cross-validation

Model	Category	Precision	Recall	Model (F1-score)	Support	Macro avg	Weighted avg	Confusion matrix	Model accuracy	Prediction class
1DCNN-2DCNN	2	0.9724	1	0.9860	247	0.9862	0.9863	0.8755	0.9859	497
	5	1	0.9720	0.9858	250	0.9860	0.9859	0.9797	0.9859	497

5. Conclusions

This study adopts a multi-modal fusion method to integrate different data sources (image features and user behavior data), enhancing clothing classification prediction accuracy and reliability. The innovation lies in clothing classification prediction for individual users, considering their needs and preferences, which is closer to users' actual requirements than traditional unified models.

This study draws several vital conclusions based on the experimental and model comparative analyses. First, after evaluating the performance of four different models, it was found that the 1DCNN-ResNet and the 1DCNN-2DCNN model showed excellent accuracy, precision, recall, and F1 score. Secondly, while the 1DCNN-ResNet model showed a slightly slower speed than the 1DCNN-2DCNN model, its superior accuracy renders it a viable alternative worth considering.

The experimental findings indicate the absence of significant overfitting during the model training process, with minimal disparity between verification and training losses, underscoring the model's robust generalization capability. Although methods such as confusion matrices have achieved good results for model evaluation, this study introduces cross-validation to further prove the model's accuracy. This evaluation technique improved the model's accuracy, reaching an accuracy of 98.59%, demonstrating the model's reliability and practicality. For personalized clothing prediction with multi-modal fusion, future research directions may explore how to adapt large-scale models to vertical fields better to improve further the effectiveness and practicality of the model in personalized clothing prediction.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

The authors would like to thank the Faculty of Engineering, Universiti Malaysia Sarawak (UNIMAS), and Qilu Institute of Technology for Supporting this research through the Research Program of Qilu Institute of Technology (No. QIT23NN016).

References

[1] X. Wu and L. Zhu, "Application of Product Form Recognition Combined with Deep Learning Algorithm," Computer-Aided Design & Applications, vol. 21, no. S15, pp. 54-68, 2024.
[2] S. Yuan, L. Zhong, and L. Li, "WhatFits- Deep Learning for Clothing Collocation," 7th International Conference on Behavioural and Social Computing, pp. 1-4, November 2020.
[3] Z. He, Y. Li, X. Shi, P. Li, and W. Huang, "Multi-Deep Features Fusion Algorithm for Clothing Image Recognition," 8th International Conference on Digital Home, pp. 104-109, September 2020.
[4] Z. W. Wang, Y. Y. Pu, X. Wang, Z. P. Zhao, D. Xu, and W. H. Qian, "Accurate Retrieval of Multi-Scale Clothing Images Based on Multi-Feature Fusion," Chinese Journal of Computers, vol. 43, no. 4, pp. 740-754, 2020. (In Chinese)
[5] S. S. Islam, E. K. Dey, M. N. A. Tawhid, and B. M. M. Hossain, "A CNN Based Approach for Garments Texture Design Classification," Advances in Technology Innovation, vol. 2, no. 4, pp. 119-125, October 2017.
[6] J. Zhao, "The Evolution of Chinese Traditional Ethnic Clothing Design Style Based on Interactive Dichroism Algorithm," Applied Mathematics and Nonlinear Sciences, vol. 9, no. 1, pp. 1-15, January 2024.
[7] X. Han, "Research on Clothing Personalized Recommendation Algorithm Based on Improved Collaborative Filtering Algorithm," 3rd International Conference on Internet of Things and Smart City (IoTSC), vol. 12708, article no. 127080C, June 2023.
[8] P. Jing, K. Cui, W. Guan, L. Nie, and Y. Su, "Category-Aware Multimodal Attention Network for Fashion Compatibility Modeling," IEEE Transactions on Multimedia, vol. 25, pp. 9120-9131, 2023.
[9] L. Liu, H. Zhang, Q. Li, J. Ma, and Z. Zhang, "Collocated Clothing Synthesis with GANs Aided by Textual Information: A Multi-Modal Framework," ACM Transactions on Multimedia Computing, Communications, and Applications, vol. 20, no. 1, article no. 26, January 2024.
[10] M. S. Amin, C. Wang, and S. Jabeen, "Fashion Sub-Categories and Attributes Prediction Model Using Deep Learning," The Visual Computer, vol. 39, no. 9, pp. 3851-3864, September 2023.
[11] H. Zhang, W. Huang, L. Liu, and T. W. S. Chow, "Learning to Match Clothing from Textual Feature-Based Compatible Relationships," IEEE Transactions on Industrial Informatics, vol. 16, no. 11, pp. 6750-6759, November 2020.
[12] Y. Chen, Z. Zhou, G. Lin, X. Chen, and Z. Su, "Personalized Outfit Compatibility Prediction Based on Regional Attention," 9th International Conference on Digital Home, pp. 75-80, October 2022.
[13] D. Kim, K. Saito, S. Mishra, S. Sclaroff, K. Saenko, and B. A. Plummer, "Self-Supervised Visual Attribute Learning for Fashion Compatibility," https://arxiv.org/pdf/2008.00348.pdf, August 12, 2021.
[14] S. Lu, X. Zhu, Y. Wu, X. Wan, and F. Gao, "Outfit Compatibility Prediction with Multi-Layered Feature FusionNetwork," Pattern Recognition Letters, vol. 147, pp. 150-156, July 2021.
[15] J. Shi, X. Song, Z. Liu, and L. Nie, "Fashion Graph-Enhanced Personalized Complementary Clothing Recommendation," Journal of Cyber Security, vol. 6, no. 5, pp. 181-198, 2021. (In Chinese)
[16] Y. Wang, L. Liu, X. Fu, and L. Liu, "MCCP: Multi-Modal Fashion Compatibility and Conditional Preference Model for Personalized Clothing Recommendation," Multimedia Tools and Applications, vol. 83, no. 4, pp. 9621-9645, January 2024.
[17] V. Ekambaram, K. Manglik, S. Mukherjee, S. S. K. Sajja, S. Dwivedi, and V. Raykar, "Attention Based Multi-Modal New Product Sales Time-Series Forecasting," Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3110-3118, August 2020.
[18] G. Skenderi, C. Joppi, M. Denitto, B. Scarpa, and M. Cristani, "The Multi-Modal Universe of Fast-Fashion: The Visuelle 2.0 Benchmark," https://arxiv.org/pdf/2204.06972.pdf, April 14, 2022.
[19] J. Yang, C. Li, P. Zhang, B. Xiao, C. Liu, L. Yuan, et al., "Unified Contrastive Learning in Image-Text-Label Space," https://arxiv.org/pdf/2204.03610.pdf, April 07, 2022.
[20] M. Wu, G. Zhang, and C. Jin, "Time Series Prediction Model Based on Multimodal Information Fusion," Journal of Computer Applications, vol. 42, no. 8, pp. 2326-2332, August 2022. (In Chinese)