Information Fusion 101 (2024) 101974

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

Multi-view Instance Attention Fusion Network for classification

Jinxing Li [a,b,1], Chuhao Zhou [a,1], Xiaoqiang Ji [c,d,*], Mu Li [a], Guangming Lu [a], Yong Xu [a,b], David Zhang [c]

[a] Harbin Institute of Technology, Shenzhen, PR China
[b] Shenzhen Key Laboratory of Visual Object Detection and Recognition, Shenzhen, PR China
[c] The Chinese University of Hong Kong, Shenzhen, PR China
[d] Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, PR China

ARTICLE INFO

Keywords:
Multi-view
Instance learning
Classification
Cross-fusion

ABSTRACT

Multi-view learning for classification has achieved a remarkable performance compared with the single-view based methods. Inspired by the instance based learning which directly regards the instance as the prior and well preserves the valuable information in different instances, a Multi-view Instance Attention Fusion Network (MvIAFN) is proposed to efficiently exploit the correlation across both instances and views. Specifically, a small number of instances from different views are first sampled as the set of templates. Given an additional instance and based on the similarities between it and the selected templates, it can be re-presented by following an attention strategy. Thanks for this strategy, the given instance is capable of preserving the additional information from the selected instances, achieving the purpose of extracting the instance-correlation. Additionally, for each sample, we not only perform the instance attention in each single view but also get the attention across multiple views, allowing us to further fuse them to obtain the fused attention for each view. Experimental results on datasets substantiate the effectiveness of our proposed method compared with state-of-the-arts.

1. Introduction

Due to the rapid development of multi-media techniques, a same object is usually represented with multiple modalities. For instance, a person can be identified by the face [1,2], fingerprint [3,4], and palmprint images [5,6];a single image can also be described with different types of features, e.g., HoG, SIFT, and LBP, etc. Generally, these multiple kinds of data are named multi-view or multi-modality data [7–10]. To comprehensively exploit multiple views, the multi-view learning has recently attracted much attention [8,11–15]. In contrast to conventional single-view based methods, multi-view learning enjoys the capability of extracting correlation and complementary information, being contributing to the performance improvement in many fields.

A typical branch of multi-view learning is to assume that there exists a common subspace, in which multiple views can be projected to it to follow some priors [16–18]. For example, the Canonical Correlation Analysis (CCA) [16] learns two projection matrices for two views, through which two mapped vectors in a common subspace are encouraged to enjoy the maximum of correlation. In recent year, due to the powerful data representation capability, some multi-view methods have also been extended to the deep version, like Deep Canonical Correlation Analysis (DCCA) [18].

Despite the fact that deep learning based multi-view approaches have achieved better performance in classification, they only focus on digging out the relationship among different views of a single instance [19–24]. As depicted in Fig. 1(a), by applying the view-specific networks to the source data, the multiple views on different spaces are first mapped to a common space. Then some priors, like that in CCA, are enforced on the mapped features to meet some specific tasks. However, in this strategy, the correlation among different instances in a view and multiple views are ignored. In fact, especially in the classification, the instances belonging to the same category usually do enjoy the similarity and they make it possible to implement the self-instances representation. A typical example is the instance based learning (IBL) methods [25], in which a sample can be well and adaptively represented by the training samples. Due to the free distribution assumption, these algorithms can more flexibly apply the relationship between the training samples and testing samples to classification compared with the traditional parametric models which are constraint to the determined or specific distribution assumption and objective function.

* Corresponding author.
E-mail address: jixiaoqiang@cuhk.edu.cn (X. Ji).
[1] Equal contribution.

https://doi.org/10.1016/j.inffus.2023.101974
Received 23 November 2022; Received in revised form 8 August 2023; Accepted 10 August 2023
Available online 22 August 2023

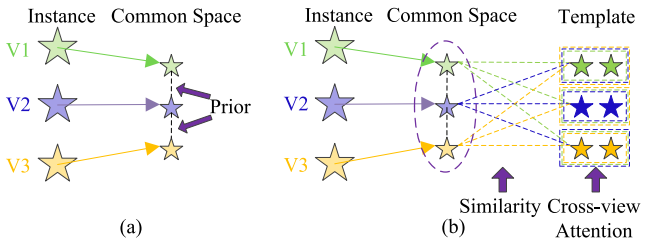

Fig. 1. The comparison between the existing multi-view method and our proposed method. (a) After projecting the observed multi-view data to a common space, different views from a same instance are jointly exploited by following some priors. (b) Instead of only exploiting the relationship among different views of an instance, the correlations between each instance and the selected keys from various views are further learned.

In spite of the advantages existing in the IBL methods, they still meet some specific limitations. (1) Although an instance can be represented by the training samples in the IBL methods, it still follows the shallow architecture, meeting the limitation of data-representation capability. In the classification, the extracted features play a key role for the performance enhancement. However, the most IBL methods simply adopt the features extracted via an off-line way, while the efficient end-to-end deep learning strategy is ignored. (2) With the increase of training samples, a large storage is inevitable for IBL based methods, e.g., KNN [26], being incapable in the real-world applications. In these existing approaches, the adequate number of training samples are necessary for the instance representation. However, in many fields with numerous samples, it is impossible to load whole training samples into the algorithm, increasing the difficulty of the IBL's real-world application. (3) Referring to the multi-view case, the cross-view instance correlation is not taken into account. Generally, multiple views are generated from a same object, so that it does exist the instance-relationship across various views, while the most existing IBL based methods ignore this valuable information.

To address the aforementioned problems, we propose a novel deep network to not only achieve the multi-view IBL through a deep structure, but also get the implementation with low storage. As shown in Fig. 1(b), after projecting the multiple views to a common space, these projected features are used to generate the similarities with a set of templates. A proposed instance attention strategy is then performed to represent the feature of a given sample as a weighted sum of the features of the selected instances. Additionally, for each sample, we not only perform the instance attention in a single view but also get the attention across multiple views, allowing us to further fuse them to obtain the fused attention for each view.

Different from the conventional deep learning methods which only forward one instance, our proposed method introduces a set of selected instances to re-represent a given instance by following the attention strategy, preventing from the distribution assumption and subsequently getting a novel feature which preserves class-related information among the selected instances. In contrast to the IBL methods, our presented approach embeds the IBL into the deep architecture, which not only allows us to simultaneously learn the deep feature mapping transformations to achieve the instance attention across various views, but also limits the storage to a low value with the small number of selected instances.

In this paper, Multi-view Instance Attention Fusion Network (MvIAFN) is proposed. The main contributions of the proposed method include:

- The proposed MvIAFN not only exploits the instance-correlation across various views, but also achieves the deep representation for each instance. To the best of our knowledge, MvIAFN is the very first effort of jointly taking the IBL and deep learning method into a unified model for classification.
- The similarity measurements between the training instances and selected instances across various views are introduced to guide the instance learning in a supervised way, encouraging a part of the selected instances to remarkably represent a given instance if they belong to the same category.
- By virtue of the proposed effective fusion method, MvIAFN shows clearly superior multi-view classification performance in comparison with state-of-the-art methods. More importantly, unlike existing IBL approaches which meet a large computational memory with the increase number of the training samples, MvIAFN requires significantly less memory overhead, which is critical for realistic large-scale applications with limited computing and memory resources.

The rest of this paper is organized as follows. In Section 2, we briefly introduce some related works about multi-view representation learning, instance-based learning and attention mechanism. Our proposed MvIAFN is then discussed in Section 3. We conduct experiments in Section 4, demonstrating the superiority of our proposed method on multi-view classification. The conclusion in Section 5 is finally concluded.

2. Related works

2.1. Multi-view representation learning

Multi-view learning aims to learn features that fully exploit the correlation and complementary information among views of an instance. It has been proved to contribute the performance improvement in many fields, including classification, object detection, recognition, etc. In general, multi-view learning algorithms can be divided into two categories: representation alignment and representation fusion [27].

Algorithms in the first category seek to project multiple views to a common subspace where projected features are under some constraints. For example, the Canonical Correlation Analysis (CCA) [16] models the correlation between two views explicitly and tries to learn two projection matrices that make the projected representation of the two views enjoy the maximum correlation. To introduce non-linearity into CCA, KCCA [17] embeds the data into a higher dimensional feature space through kernerlization. Extending CCA to a deep version, DCCA [18] learns deep nonlinear transformation of two views, through which two transformed features are highly and linearly correlated.

Algorithms in the second category aim to directly learn a shared and compact representation from multi-view data. For instance, MSAF [20] splits each view into channel-wise equal feature blocks where an optimized feature map can be generated for each view based on the corresponding per channel block-wise attention. Then, a joint representation for all views can be learned from the optimized feature maps. To fully explore the underlying interactive relations among multiple views, CMRN [21] proposes three subnetworks to respectively extract view-specific features, capture collaborative knowledge from multiple views, and characterize the shared-specific correlations. Besides, TMC [22] dynamically integrates different views according to the evidence from each view, which promotes the reliability and robustness of the model for multi-view classification.

The closest works to our proposed method are MvDA [28] and MvNNCor [19]. To find a single discriminant common space for multiple views, MvDA learns multiple linear transformations that are solved by optimizing a generalized Rayleigh quotient. MvNNCor projects view-specific features to a common space by a series of neural networks and captures complementary information among projected features through outer product and a shared mapping module. In contrast to all of the above, our work is the first to take the correlation among different instances in a view and multiple views into consideration. The proposed MvIAFN unifies the IBL and deep learning method into a single model and can fully exploit the valuable information from datasets through selected templates.

2.2. Instance-based learning and attention mechanism

Instance-based learning (IBL) holds the faith that a sample can be well and adaptively represented by the training samples. As a primary example, K-nearest neighbors (KNN) [29], serving as a classifier, assigns an instance to the category of the nearest set in the training samples. Similarly, a growing number of literatures adopt IBL methods as a classifier for higher interpretability and performance. Ouchi et al. [30] learned representations for spans through an IBL method and quantified the contribution of each training instance based on similarities among testing and training spans. Haddad et al. [31] obtained 3D representations of human action through GF-OF and K-means which are then represented by a Gaussian mixture model, and finally KL-divergence and the IBL method are applied to complete the human action recognition task. In the scenario of multi-view classification, Sun et al. [32] utilized the Fenchel–Legendre conjugates to rewrite an insensitive loss which involves unlabeled data as a regularization. By this way, they achieved to represent the target function by a few labeled data and only a small amount of unlabeled data, effectively accelerating the function evaluations of a series of semi-supervised methods. Besides, MvDGPs [33] extends the powerful deep Gaussian processes (DGPs) [34] into the multi-view representation learning and realizes adaptive modeling depths of DGPs for different views. Consequently, a more reasonable joint representation that models the discrepancies of different views can be achieved. In addition to taking the IBL method as a classifier, the idea of IBL is also adopted in the scenario of clustering. For example, Chen [35] applied it for feature selection. He tried to identify the salient features that are most helpful in dividing the nearest and farthest neighbors of an instance. Zhang et al. [36] utilized the idea of IBL and proposed a joint framework called MUSLA for multivariate time series clustering. The MUSLA first determines a group of salient subsequences with different lengths to form the multi-view shapelets. Then, the shapelet-based multi-view representation can be obtained for the multivariate time series, based on which the task of clustering can be done with the help of an adaptive neighbor model.

In addition, the attention mechanism has been increasingly applied both in the field of computer vision and natural language processing recently [37–42]. Volodymyr et al. [37] introduced attention to the image classification task through recurrent neural networks and extracted information from adaptively selected regions. Bahdanau et al. [38] proposed a model to automatically search for words in a source sentence that are relevant to predict a target word and obtain a promising performance in the machine translation task. Besides, self-attention has been adopted in [39] to compute a representation of the sequence, which is widely utilized in many works.

In this paper, we propose the instance attention strategy in MvIAFN to fully exploit the valuable information between the feature of a given sample and selected template features in both single-view and cross-view scenarios. As far as we know, our MvIAFN is the very first work that applies the IBL method to representation learning. Besides, since we only sample a small ratio of instances from training set as templates, our method can effectively alleviate the "huge data size" problem of the IBL method that mentioned in [43].

3. Multi-view Instance Attention Fusion Network

The pipeline of the proposed MvIAFN is displayed in Fig. 2. As we can observe, by designing a view-specific network for each view, the source data is transformed to a subspace with the same dimensionality. Guided by the similarity measurement, the instance attention strategy is introduced to represent a given instance across all views in a weighted way. By applying the maximization operation, the final instance re-represented feature is obtained for each view. Jointly taking the re-represented features and original projected features into account, the prediction of a sample is achieved by using an additional network followed by the softmax.

Assume that the V views of multi-view input data are denoted as $\mathbf{F} = \left\{\mathbf{F}^1, \mathbf{F}^2, \ldots, \mathbf{F}^v, \ldots, \mathbf{F}^V\right\}$ and $\mathbf{F}^v = \left[\mathbf{f}_1^v, \ldots, \mathbf{f}_N^v\right] \in \Re^{d^v \times N}$, where N is the number of training samples and d^v is the dimensionality of the vth view. To project multiple views to a common space for further processing, view-specific networks are constructed and the outputs of them enjoy the same dimensionality. Denote these networks as $\boldsymbol{g} = \left\{\boldsymbol{g}^1, \ldots, \boldsymbol{g}^v, \ldots, \boldsymbol{g}^V\right\}$. Then the transformed features are represented as:

$$\mathbf{x}_i^v = \boldsymbol{g}^v(\mathbf{f}_i^v), \quad s.t., \ v = 1, \ldots, V \tag{1}$$

where $\mathbf{x}_i^v \in \Re^{d\times 1}$ and d is the dimensionality of the common space. Specifically, in this paper, each neural network $\boldsymbol{g}^v$ is defined with L layers:

$$\mathbf{h}_{g_v}^l = \delta\left(\mathbf{W}_{g_v}^l \mathbf{h}_{g_v}^{l-1} + \mathbf{b}_{g_v}^l\right), \quad s.t., \ l = 1, \ldots, L \tag{2}$$

where δ is the activation function; $\mathbf{h}_{g_v}^l$, $\mathbf{W}_{g_v}^l$ and $\mathbf{b}_{g_v}^l$ are the output, weight matrix, and bias vector associated with the lth layer, respectively. Particularly, $\mathbf{f}_i^v = \mathbf{h}_{g_v}^0$ and $\mathbf{x}_i^v = \mathbf{h}_{g_v}^L$.

3.1. Instance attention

Instance-based learning (IBL) algorithms assume that similar instances have similar assignment, leading to their local bias for classifying a novel instance according to their most similar neighbor's assignment [25]. Given the training set, most learning algorithms derive the generalizations by following some hand-crafted priors or distribution assumptions. By contrast, being free from these assumptions, IBL algorithms calculate the similarities of their training instances with the newly presented instance, being more smooth in data representation. However, despite that there do exist many advantages in IBL algorithms, they are sensitive to the quality of inputting features and meet the large storage requirements. Fortunately, thanks to the development of deep learning, here we efficiently embed the IBL into the deep neural network (DNN) to tackle the problems mentioned above.

Instead of taking whole training samples to represent an instance, here we select a small number of samples as the set of templates. Mathematically, denote the set of templates belonging to the vth view as $\mathbf{U}^v = \left[\mathbf{u}_1^v, \ldots, \mathbf{u}_{N_t}^v\right] \in \Re^{d^v \times N_t}$, where N_t is the number of templates (Templates are selected and updated by a clustering-based method from the projected features. We will discuss this selection in the following subsection.). Then their corresponding features can be represented as $\mathbf{T}^v = \left[\mathbf{t}_1^v, \ldots, \mathbf{t}_{N_t}^v\right] \in \Re^{d\times N_t}$. Inspired by the attention mechanism [39], we prefer the project features to be well represented by a part of templates if the projected features are similar to these templates. Here each projected feature can be regarded as the query, while all of the templates can be regarded as the keys. According to the pairwise similarity between each projected feature and template, the weight between them is obtained, allowing us to attain the attention $\mathbf{a}_i^{v,v}$ by weighted summarizing the template features. Mathematically, their relationship can be represented as:

$$\mathbf{a}_i^{v,v} = \sum_j^{N_t} w_{i,j}^{v,v} A(\mathbf{t}_j^v), \ s.t., \sum_j^{N_t} w_{i,j}^{v,v} = 1, \tag{3}$$

where $w_{i,j}^{v,v}$ is the estimated weight between $\mathbf{x}_i^v$ and $\mathbf{t}_j^v$, $A(\cdot)$ is a convolution layer with kernel size 1 and weight matrix W_A, and $\mathbf{a}_i^{v,v}$ is the generated attention vector through summarizing up all of templates in a weighted way. Since the template set is selected from the training set and its distribution or domain is approximate to the original one, the attention feature in Eq. (3) combined by all of the templates is capable of efficiently preserving more valuable information.

Additionally, as displayed in Fig. 3, when an instance is near the classification hyperplane, the templates are capable of pulling its attention feature away from other classes. It is true that there may exist templates which are far from their own class center but close to other class centers (denoted by red color in Fig. 3), making an inferior influence on the attention feature generation. However, thanks to our

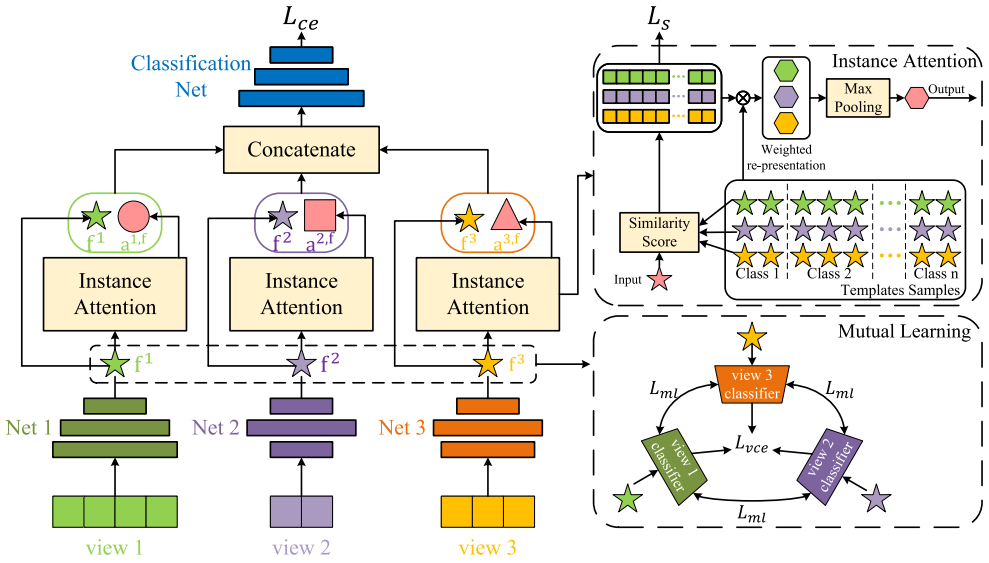

Fig. 2. The pipeline of the proposed MvIAFN. View-specific networks from different views are first learned to project each view to a common space. The projected features belonging to different views are then represented by the template features by following their similarity. Applying the cross fusion strategy and concatenation, the alignment of an instance is obtained using the softmax based network. Additionally, the mutual learning module is applied to further increase the discriminability of each view and enable them to learn classification-related knowledge from each other.

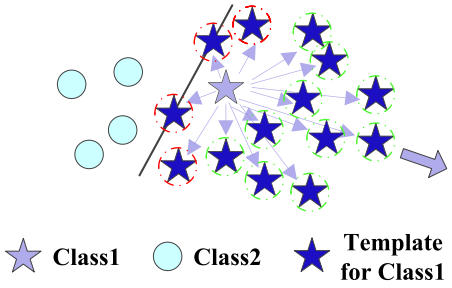

Fig. 3. The working insight of the proposed attention strategy.

template selection method, we can select templates which are relatively close to the class center. These templates (denoted by green color in Fig. 3) are far from other class centers. Subsequently, according to Eq. (3), the generated attention feature would be much farther from the hyperplane compared with the given instance, contributing to the classification.

Straightforwardly, the weight $w_{i,j}^{v}$ is significant for getting $\mathbf{a}_i^{v,v}$. Here we achieve it through its associated similarity. Particularly, the cosine similarity is used to measure the similarity between the instance and the each template.

$$s_{i,j}^{v,v} = \frac{\mathbf{x}_i^v \cdot \mathbf{t}_j^v}{\left\|\mathbf{x}_i^v\right\| \left\|\mathbf{t}_j^v\right\|} \tag{4}$$

From Eq. (4) we can see that if $\mathbf{x}_i^v$ and $\mathbf{t}_j^v$ are correlative, then their similarity $s_{i,j}^{v,v}$ would be close to 1. Otherwise, $s_{i,j}^{v,v}$ is close to 0.

In the classification, we prefer to enforce the samples belonging to the same class to be close while that belonging to the distinctive classes to be far. In other words, it is reasonable to pay more attention on the templates under the same category. Mathematically, as displayed by Eq. (4), if $\mathbf{x}_i^v$ and $\mathbf{t}_j^v$ enjoy the same category, $s_{i,j}^{v,v}$ is encouraged to be 1. Otherwise, $s_{i,j}^{v,v} = 0$. However, in Eq. (4), there is no any supervised constraint on it. Therefore, in this paper, a similarity loss is proposed to tackle this issue.

$$L_s^{v,v} = \frac{1}{N \times N_t} \sum_{i}^{N} \sum_{j}^{N_t} (s_{i,j}^{v,v} - gt_{i,j})^2 \tag{5}$$

where $L_s^{v,v}$ is the similarity loss between the projected feature and templates belonging to the vth view, and $gt_{i,j}$ is the predefined ground-truth. $gt_{i,j} = 1$ if $\mathbf{x}_i^v$ and $\mathbf{t}_j^v$ belong to the same category. Otherwise, $gt_{i,j} = 0$. From Eq. (5), it is easy to observe that under the guidance of $gt_{i,j}$, $\mathbf{x}_i^v$ is encouraged to pay the attention on the templates with the same category, enjoying the valuable supervised information.

Finally, by applying the softmax to $\{s_{i,j}^{v,v}\}_{j=1}^{N_t}$, the weights $\{w_{i,j}^{v,v}\}_{j=1}^{N_t}$ corresponding to different templates are then computed.

$$\left[w_{i,1}^{v,v}, w_{i,2}^{v,v}, \ldots, w_{i,N_t}^{v,v}\right] = softmax\left(\left[s_{i,1}^{v,v}, s_{i,2}^{v,v}, \ldots, s_{i,N_t}^{v,v}\right]\right) \tag{6}$$

3.2. Cross fusion

In the aforementioned subsection, only the instance attention in a view is extracted, while the correlation across different views is ignored. Since all of views are generated from a same object, it is reasonable to assume that the projected features in the common subspace from different views enjoy the similarity or dissimilarity by following their belongings. Thus, the similarity matrices across various views are obtained through

$$\mathbf{s}_{i,j}^{v,\bar{v}} = \frac{\mathbf{x}_i^v \cdot \mathbf{t}_j^{\bar{v}}}{\left\|\mathbf{x}_i^v\right\| \left\|\mathbf{t}_j^{\bar{v}}\right\|}, \tag{7}$$

where $\bar{v} \in \{1, 2, \ldots, v-1, v+1, \ldots, V\}$ means the $\bar{v}$-th view is different from the vth view. Similarly, the similarity loss and weights on these cross views are

$$L_s^{v,\bar{v}} = \frac{1}{N \times N_t} \sum_{i}^{N} \sum_{j}^{N_t} (s_{i,j}^{v,\bar{v}} - gt_{i,j})^2 \tag{8}$$

$$\left[w_{i,1}^{v,\bar{v}}, w_{i,2}^{v,\bar{v}}, \ldots, w_{i,N_t}^{v,\bar{v}}\right] = softmax\left(\left[s_{i,1}^{v,\bar{v}}, s_{i,2}^{v,\bar{v}}, \ldots, s_{i,N_t}^{v,\bar{v}}\right]\right). \tag{9}$$

Then the attention representation of $\mathbf{x}_i^v$ based on $\left\{\mathbf{t}_j^{\bar{v}}\right\}_{j=1}^{N_t}$ is

$$\mathbf{a}_i^{v,\bar{v}} = \sum_{j}^{N_t} w_{i,j}^{v,\bar{v}} A(\mathbf{t}_j^{\bar{v}}), \ s.t., \sum_{j}^{N_t} w_{i,j}^{v,\bar{v}} = 1. \tag{10}$$

From the aforementioned analysis, a sample $\mathbf{x}_i^v$ from a view would generate V novel attentions by doing the instance attention processing with the templates of V views, which are $\{\mathbf{a}_i^{v,k}\}_{k=1}^{V}$. To preserve the

most valuable information, in this paper, we further apply the maximization processing to them by following the view direction to get the cross-fused vector.

$$\mathbf{a}_i^{v,f} = \max_k[\mathbf{a}_i^{v,1}, \ldots, \mathbf{a}_i^{v,k}, \ldots, \mathbf{a}_i^{v,V}] \tag{11}$$

Taking all views $\{\mathbf{x}_i^v\}_{v=1}^V$ into account, there are finally V fused attentions $\{\mathbf{a}_i^{v,f}\}_{v=1}^V$ for the ith instance.

3.3. Classification

To prevent from losing the information in the instance attention processing, we additionally concatenate $\{\mathbf{x}_i^v\}_{v=1}^V$ and $\{\mathbf{a}_i^{v,f}\}_{v=1}^V$ for each view. By inputting them into an another shared network with softmax, the final predicted probabilities for different classes are achieved, which is denoted as $\mathbf{z}_i$.

$$\mathbf{z}_i = \boldsymbol{g}^s(\{[\mathbf{x}_i^v; \mathbf{a}_i^{v,f}]\}_{v=1}^V) \tag{12}$$

where $\boldsymbol{g}^s$ is the network constructed by fully-connected layers (being similar to Eq. (2)) and softmax.

Here we use the cross-entropy loss to handle this supervised learning.

$$L_{ce} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(z_{i,c}) \tag{13}$$

where $\mathbf{y}_i = [y_{i,1}; y_{i,2}; \cdots; y_{i,C}]$ is the one-hot label vector of the ith sample, C is the total number of categories, and $z_{i,c}$ denotes the cth element in $\mathbf{z}_i$.

To further increase the discriminability of each view and enable them to learn classification-related knowledge from each other, we introduce the view-specific classification loss L_{vce} and mutual learning loss L_{ml}. As shown in Fig. 2, each view is passed to a view-specific classifier $\boldsymbol{g}_c^v$ to obtain its corresponding predicted probabilities $\mathbf{z}_i^v = \boldsymbol{g}_c^v(\mathbf{x}_i^v)$. Then, we utilize the cross-entropy loss and the Kullback–Leibler divergence loss to handle the view-specific supervised learning and cross-view mutual learning, respectively.

$$L_{vce} = -\frac{1}{N\times V}\sum_{v=1}^{V}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c}\log(z_{i,c}^v) \tag{14}$$

$$L_{ml} = \frac{1}{N\times V\times(V-1)}\sum_{i=1}^{N}\sum_{a,b=1,a\neq b}^{V} D_{KL}(z_i^a \| z_i^b) \tag{15}$$

Jointly taking Eqs. (5), (8), (13), (14) and (15) into account, the networks can be learning by following

$$L = L_{ce} + L_{vce} + L_{ml} + \frac{\eta}{V\times V}\sum_{v=1}^{V}\sum_{k=1}^{V} L_s^{v,k} \tag{16}$$

where η is the non-zero parameter to control the importance of the similarity losses.

3.4. Template selection

Templates play a rather important role in our proposed method since the projected features of each view from a newly presented instance will be re-represented by the templates to obtain the instance attention. Therefore, an effective and rational template selection approach is instrumental to the performance of MvIAFN.

As mentioned in Fig. 3, templates are capable of pulling the attention features of an instance away from other classes when it is near the classification hyperplane. Intuitively, we should take instances that are near the center of each class as the templates, which brings two advantages. The one is that templates of each class are gathered around the class center, so that the distance among templates within one class is minimized while the distance among templates between two different classes is maximized. Consequently, such templates can provide instances near the classification hyperplane with more significant 'pull' effect. That is to say, if an instance is similar to templates in a certain class, it will be pulled to the center of that class more easily so as to be further away from classification hyperplane. The other advantage is that, we can avoid taking outliers as templates, subsequently reducing the inferior influence and fluctuation on the performance.

Algorithm 1: Template Selection

Input: projected features of all training instances $\{\mathbf{X}_c^v\}$
Output: $V \times N_t$ templates features

for v-th view in V views **do**
  for class c in C classes **do**
    $\epsilon \leftarrow \epsilon_0$
    $min_{samples} \leftarrow K_c$
    Utilize DBSCAN(ϵ, $min_{samples}$) to cluster over all instances $\mathbf{X}_c^v$.
    while n_cluster $\neq 1$ **do**
      $\epsilon \leftarrow \epsilon + 0.1$
      Utilize DBSCAN(ϵ, $min_{samples}$) to cluster over all instances $\mathbf{X}_c^v$ again.
    end while
    $ClusterCentroid \leftarrow$ the mean of instances in the cluster
    Compute distances from all instances $\mathbf{X}_c^v$ to the $ClusterCentroid$
    Take top-K_c nearest instances to the $ClusterCentroid$ as templates for v-th view of class c.
  end for
end for
return $V \times N_t$ selected template features

In this paper, a template selection method based on DBSCAN [44] is introduced to obtain optimized templates, as displayed in Algorithm 1. Assume that a dataset has C classes and each instance has V different views, $\mathbf{X}_c^v$ means projected features for the vth view of instances with class label c. The main idea of the algorithm can be summarized as follows:

(1) For the vth view of all training instances with class label c, we will select a class center based on projected features. In order to get rid of the impact caused by outliers, we only focus on relatively concentrated instances within one class. This is achieved by executing the DBSCAN algorithm until instances are aggregated into one cluster, and the rest instances within the same class are regarded as outliers. Specifically, we firstly initialize the radius ϵ and minimum samples $min_{samples}$ of the DBSCAN algorithm to ϵ_0 and K_c respectively, and add 0.1 to ϵ for each time. In the experiments, ϵ_0 is set to a small value 0.1 and K_c is the number of selected templates determined by the number of instances in class c and a pre-defined sample rate ξ.

(2) The centroid of the cluster is then determined by the mean value of all instances in the cluster.

(3) We select top-K_c instances nearest to the centroid as templates for the vth view of class c.

(4) We repeat the aforementioned process for V views and C classes. As a result, $V \times N_t$ templates are selected. Here N_t is the number of templates for the vth view which can be computed as:

$$N_t = \sum_{c}^{C} \xi \times N_c \tag{17}$$

where ξ is the sample rate and N_c is the number of instances in class c.

It is noticeable that the class center may deviate from its original place with the update of projected features. Therefore, we re-execute the template selection algorithm every epoch and re-collect templates to match such deviation.

Table 1
The architectures of $\{\boldsymbol{g}^v\}_{v=1}^V$, $\boldsymbol{g}^s$, and $\{\boldsymbol{g}_c^v\}_{v=1}^V$.

Layers	$\boldsymbol{g}^v$	$\boldsymbol{g}^s$	$\boldsymbol{g}_c^v$
Layer1	FC - $d^v \times 400$ BatchNorm ReLU Dropout	FC - $(V \times 2 \times 200) \times 600$ BatchNorm ReLU Dropout	FC - $200 \times C$ BatchNorm
Layer2	FC-400 × 200 BatchNorm ReLU	FC-600 × C BatchNorm	/

3.5. Implementation

To implement MvIAFN, the view-specific networks $\{\boldsymbol{g}^v\}_{v=1}^V$, shared network $\boldsymbol{g}^s$ and view-specific classifiers $\{\boldsymbol{g}_c^v\}_{v=1}^V$ should be designed. In this paper, being similar to MvNNcor, we achieve them using the fully connected networks. Specifically, each $\boldsymbol{g}^v$ consists of two layers by following $[d^v \times 400, 400 \times 200]$. After the attention block, we can get the concatenated feature whose dimensionality is $V \times 2 \times 200$. Thus the fully connected architecture of $\boldsymbol{g}^s$ is $[(V \times 2 \times 200) \times 600, 600 \times C]$. Besides, each $\boldsymbol{g}_c^v$ has a layer with architecture $[200, C]$ to conduct the view-specific classification. Note that, except for the last layer in $\boldsymbol{g}^s$ and the layers in $\{\boldsymbol{g}_c^v\}_{v=1}^V$, each fully connected layer is followed by the 'BatchNorm' and 'ReLU'. Also, there is an additional 'Dropout' in the first layer of $\boldsymbol{g}^v$ and $\boldsymbol{g}^s$. For the last layer in $\boldsymbol{g}^s$ and every layer in $\{\boldsymbol{g}_c^v\}_{v=1}^V$, only 'BatchNorm' is followed. Table 1 shows the details of networks.

We implement our model by Pytorch and train it on a GPU TITAN Xp. Here we set the batchsize, epochs and learning rate to 128, 120, and 0.001 initially. After each 30 epochs, the learning rate is decreased by setting the weight decay factor as 0.1. The weights of the networks are optimized by the Adam optimizer. For the template selection, we select templates through the aforementioned template selection method and re-collect templates for every epoch. We then forward the selected templates $\{\mathbf{U}^v\}_{v=1}^V$ into $\{\boldsymbol{g}^v\}_{v=1}^V$ to obtain their corresponding features $\{\mathbf{T}^v\}_{v=1}^V$. The training process of MvIAFN is shown in Algorithm 2.

Algorithm 2: Training Process of MvIAFN

for epoch in range $(0, 120)$ **do**
  $\{\mathbf{U}^v\} \leftarrow$ Template Selection$(\{\mathbf{X}_c^v\})$
  $\{\mathbf{T}^v\} \leftarrow \boldsymbol{g}^v(\{\mathbf{U}^v\})$
  $\{\mathbf{a}_i^{v,k}\}_{k=1}^V \leftarrow \sum_j^{N_t} w_{i,j}^{v,k} A(\mathbf{t}_j^k)$
  $\{\mathbf{a}_i^{v,f}\}_{v=1}^V \leftarrow \max_k[\mathbf{a}_i^{v,1}, \cdots, \mathbf{a}_i^{v,k}, \cdots, \mathbf{a}_i^{v,V}]$
  $\mathbf{z}_i \leftarrow \boldsymbol{g}^s(\{[\mathbf{x}_i^v; \mathbf{a}_i^{v,f}]\}_{v=1}^V)$
  $\{\mathbf{z}_i^v\}_{v=1}^V \leftarrow \{\boldsymbol{g}_c^v(\mathbf{x}_i^v)\}_{v=1}^V$
  Calculate L_{ce}, L_{vce}, L_{ml} and L_s based on Eq.(5), Eq.(8), Eq.(13), Eq.(14) and Eq.(15)
  Update $\{\mathbf{W}_{g_v}^l\}$, $\{\mathbf{b}_{g_v}^l\}$, $\mathbf{W}_A$, $\{\mathbf{W}_{g_s}^l\}$ and $\{\mathbf{b}_{g_s}^l\}$
end for
return

4. Experiment

In this section, the comparison experiments as well as the ablation studies are conducted to demonstrate the superiority of our proposed method.

4.1. Dataset

Being similar to MvNNcor, we conduct experiments on the AWA [45,46], NUSOBJ [47], Caltech101/20 [48], Hand [49] and Reuters [50] datasets. Furthermore, we further conduct experiments on another challenging dataset CMU-MOSEI [51].

The AWA dataset [45,46] consists of 30,475 images, in which there are 50 animals classes. Here six pre-extracted features including 2688-D color histogram feature, 2000-D local self-similarity feature, 252-D pyramid HOG feature, 2000-D color SIFT feature, 2000-D SIFT feature, and 2000-D SURF feature are regarded as the multi-view data.

The NUSOBJ dataset is a subset of the NUS-WIDE dataset [47], in which 31 object categories are composed. It contains 30,000 images and each image is represented as 5 types of features: 64-D color histogram, 225-D blockwise color moments, 144-D color correlogram, 73-D edge direction histogram, and 128-D wavelet texture.

The Caltech101/20 dataset is a widely used subset of the dataset [48] that consists of 102 categories (101 object categories and an additional background class). Following [19], we select 2386 images of 20 classes and 9144 images of 102 classes for Caltech20 and Caltech101, respectively. Each image is pre-extracted 6 kinds of features: 48-D Gabor, 40-D Wavelet moments, 254-D CEN· TRIST, 1984-D HOG, 512-D GIST, and 928-D LBP.

The Hand dataset [49] collects handwritten numerals from '0' to '9'. Each digit contains 200 patterns and there are 2,000 patterns in total. The feature set of each digit has 6 types of features: 240-D Pix, 76-D Fou, 216-D Fac, 47-D Zer, 64-D Kar, and 6-D Mor.

The Reuters dataset [50] contains 18,758 documents with 6 classes, in which each sample is represented with five languages: 21531-D English, 24892-D French, 34251-D German, 15506-D Italian, and 11547-D Spanish. Here, different languages are treated as different views of a document.

Furthermore, we further conduct experiments on another challenging dataset CMU-MOSEI [51]. The CMU-MOSEI dataset consists of 23,453 annotated video segments from 1000 distinct speakers and 250 topics acquired from social media channels. There are 7 classes in the dataset, and each video segment is represented by a 50 × 35 visual feature, a 50 × 74 textual feature and a 50 × 768 text feature (extracted by BERT). For the sake of simplicity, in our experiments, we do average over each pre-extracted feature and get 35-D visual feature, 74-D textual feature and 768-D text feature respectively.

4.2. Experimental setting

Several state-of-the-art methods for multi-view classification are compared with our MvIAFN method, including MSAF [20], CMRN [21], TMC [22], MvNNcor [19], MEIB [23] and MvESR [24]. Specifically, we extend MSAF to six views by keeping the view-specific LSTM network architecture the same as the 'Bert Text LSTM network' in the official code. Besides, since the official implementation of MEIB is for two views, we re-implement it to support for more views by utilizing extra view-specific encoders and fusing all of them to obtain the joint representation. Since CMRN is limited to three views or less, we can only demonstrate classification performance with no more than three views. For TMC, MvNNcor and MvESR, we follow the default settings in their corresponding official codes. For all experiments, we conduct four times and take their average.

To demonstrate the discriminant and generalization, except of CMU-MOSEI, each dataset is separated into three parts: 50% samples for training, 30% samples for validation, and rest samples for testing. For CMU-MOSEI, we directly follow its source layout. Here the top-1 classification accuracy is exploited to quantitatively evaluate the effectiveness of different methods. The experiments results may fluctuate with the change of random seeds. To keep the experimental settings on all datasets consistent, we **fix the random seed** on all datasets. Besides, for all experiments, we conduct four times and take their average as the results. Note that, in MvNNcor, training, validation and testing sets are set to 7:2:1 for each dataset. In fact, we also conduct experiments on the setting of 7:2:1 and results compared with MvNNcor are shown in Table 2. It can be observed that MvIAFN is also superior to MvNNcor in all cases. Furthermore, in the scenario of 7:2:1, it may cause relatively large fluctuations due to the small number of testing samples. Therefore, we follow aforementioned 5:3:2 partition with the fixed random seed in all experiments.

Table 2
Results when the training, validation and testing sets are 7:2:1. The 'Fixed Random Seed' means we fix the random seed on all datasets.

Settings	Methods	AWA	Caltech101	Caltech20	NUSOBJ	Hand	Reuters
Fixed Random Seed	MvNNcor	46.826	76.172	95.117	51.087	99.022	88.058
	MvIAFN	**47.819**	**77.372**	**97.266**	**51.732**	**99.219**	**89.174**

Table 3
Comparison results of MvIAFN and other state-of-the-art methods on all the datasets with different views.

Datasets	Methods	Views				
		(1,2)	(1,2,3)	(1,2,3,4)	(1,2,3,4,5)	(1,2,3,4,5,6)
	MSAF [20]	25.198	25.396	25.712	26.879	28.006
	CMRN [21]	19.304	22.607	/	/	/
	TMC [22]	22.033	24.446	25.000	25.415	26.246
AWA	MvNNcor [19]	31.951	35.221	40.139	40.625	43.849
	MEIB [23]	32.053	35.878	40.685	41.717	44.542
	MvESR [24]	31.901	35.737	40.195	41.076	43.250
	MvIAFN(ours)	**33.489**	**36.068**	**41.722**	**42.238**	**45.878**
		(1,2)	(1,2,3)	(1,2,3,4)	(1,2,3,4,5)	(1,2,3,4,5,6)
	MSAF [20]	38.477	39.648	47.982	49.414	47.005
	CMRN [21]	32.812	42.318	/	/	/
	TMC [22]	32.487	33.203	64.909	64.518	63.411
Caltech101	MvNNcor [19]	47.201	55.127	69.059	71.435	73.519
	MEIB [23]	49.349	57.683	70.866	74.935	75.586
	MvESR [24]	48.405	55.990	70.248	73.210	74.382
	MvIAFN(ours)	**50.130**	**58.463**	**72.168**	**75.830**	**76.970**
		(1,2)	(1,2,3)	(1,2,3,4)	(1,2,3,4,5)	(1,2,3,4,5,6)
	MSAF [20]	71.615	71.875	82.292	79.948	82.031
	CMRN [21]	59.375	80.208	/	/	/
	TMC [22]	64.855	68.229	81.510	80.990	81.250
Caltech20	MvNNcor [19]	82.617	87.630	94.466	95.247	96.354
	MEIB [23]	84.635	89.063	93.750	95.313	96.485
	MvESR [24]	83.334	88.802	93.490	94.531	95.573
	MvIAFN(ours)	**85.482**	**89.779**	**94.922**	**95.703**	**97.070**
		(1,2)	(1,2,3)	(1,2,3,4)	(1,2,3,4,5)	/
	MSAF [20]	33.293	36.338	41.927	43.630	/
	CMRN [21]	30.529	35.917	/	/	/
	TMC [22]	31.070	36.178	38.161	39.964	/
NUSOBJ	MvNNcor [19]	35.907	40.881	47.261	49.439	/
	MEIB [23]	37.460	41.437	48.678	50.581	/
	MvESR [24]	35.327	41.206	47.586	50.241	/
	MvIAFN(ours)	**37.916**	**42.083**	**49.124**	**51.923**	/
		(1,2)	(1,2,3)	(1,2,3,4)	(1,2,3,4,5)	(1,2,3,4,5,6)
	MSAF [20]	94.063	94.844	95.938	96.563	96.875
	CMRN [21]	93.125	96.250	/	/	/
	TMC [22]	93.282	94.063	94.532	93.438	93.751
Hand	MvNNcor [19]	97.656	97.852	98.047	98.438	98.829
	MEIB [23]	98.438	98.047	98.438	98.829	99.219
	MvESR [24]	97.656	98.047	98.243	98.438	98.633
	MvIAFN(ours)	**98.438**	**99.023**	**99.219**	**99.414**	**99.609**
		(1,2)	(1,2,3)	(1,2,3,4)	(1,2,3,4,5)	/
	MSAF [20]	86.491	86.637	86.931	87.126	/
	CMRN [21]	80.713	82.113	/	/	/
	TMC [22]	87.793	87.877	88.119	88.379	/
Reuters	MvNNcor [19]	86.947	87.614	87.826	88.200	/
	MEIB [23]	88.021	87.940	88.363	88.005	/
	MvESR [24]	87.338	87.630	87.923	87.956	/
	MvIAFN(ours)	**88.086**	**88.200**	**88.607**	**89.225**	/
		(A,V)	(A,T)	(V,T)	(A,V,T)	/
	MSAF [20]	**41.726**	50.402	51.091	50.977	/
	CMRN [21]	41.319	49.436	50.856	50.564	/
	TMC [22]	41.558	49.132	49.653	48.741	/
CMU-MOSEI	MvNNcor [19]	41.064	50.521	50.217	51.031	/
	MEIB [23]	41.363	50.673	50.499	51.237	/
	MvESR [24]	41.161	50.966	50.576	51.476	/
	MvIAFN(ours)	41.466	**51.313**	**51.427**	**52.018**	/

Table 4
Comparison results of 'MvIAFN(single)' and 'Baseline' on all the datasets with single view.

Datasets	Methods	View					
		(1)	(2)	(3)	(4)	(5)	(6)
AWA	Baseline	20.032	25.262	17.568	29.547	21.474	30.348
	MvIAFN(single)	20.974	25.861	18.429	29.657	22.196	31.070
		(1)	(2)	(3)	(4)	(5)	(6)
Caltech101	Baseline	37.500	37.630	37.174	37.891	38.151	37.500
	MvIAFN(single)	40.169	40.625	41.081	40.755	40.039	39.388
		(1)	(2)	(3)	(4)	(5)	(6)
Caltech20	Baseline	73.438	72.917	73.438	73.438	72.917	72.656
	MvIAFN(single)	75.781	75.260	74.479	74.599	74.479	73.698
		(1)	(2)	(3)	(4)	(5)	/
NUSOBJ	Baseline	30.849	30.809	31.069	30.689	30.869	/
	MvIAFN(single)	31.711	31.711	32.312	31.971	32.171	/
		(1)	(2)	(3)	(4)	(5)	(6)
Hand	Baseline	96.680	96.484	96.875	97.071	96.680	96.875
	MvIAFN(single)	97.071	97.656	98.047	97.852	97.071	97.266
		(1)	(2)	(3)	(4)	(5)	/
Reuters	Baseline	87.081	87.584	86.999	87.333	87.361	/
	MvIAFN(single)	87.875	88.226	87.890	87.947	87.891	/
		A	V	T	/	/	/
CMU-MOSEI	Baseline	41.211	40.994	48.220	/	/	/
	MvIAFN(single)	41.385	41.243	50.694	/	/	/

Table 5
Comparison results of MvIAFN and its variations on several datasets. 'concat' denotes that all source views are first concatenated and then forwarded into the fully-connected layers. 'self' denotes that the instance attention strategy is only applied to one view. By contrast, 'cross' means the cross-fusion strategy is further introduced. 'random' means the templates are selected randomly, while 'cluster' means the templates are selected through clustering. 'cluster at begin' means clustering is only done before the training stage. 'w/o L_s, L_{ml}' means the similarity and mutual learning losses are removed from the objective function. 'w/o L_s' means the similarity losses are removed from the objective function. 'K-means' means the clustering method 'DBSCAN' is replaced by 'K-means'.

	AWA	Caltech101	Caltech20	NUSOBJ	CMU-MOSEI
Concat	35.784	65.739	92.076	47.571	50.099
Cross + random	45.373	76.468	96.485	51.577	51.725
Cross + cluster at begin	45.418	76.627	96.354	51.648	51.579
Self + cluster	45.447	76.563	96.209	51.673	51.736
Cross + cluster (w/o L_s, L_{ml})	45.528	76.074	96.354	51.257	50.917
Cross + cluster (w/o L_s)	45.703	76.351	96.680	51.788	51.655
Cross + cluster(K-means)	45.210	76.340	96.485	51.763	51.650
Cross + cluster	45.878	76.970	97.070	51.923	52.018

4.3. Experimental comparison

The experimental results on all the datasets obtained by state-of-the-art methods and our MvIAFN method are displayed in Table 3. Here $(1, 2)$ means only the first and second views in the dataset are used. Then the views are gradually added one by one, until all views are totally exploited. Note that the order of views for AWA, Caltech20/101, NUSOBJ, Hand and Reuters datasets are listed in 4.1. Besides, (A,T) means only the audio and text modalities are used in CMU-MOSEI. We takes turns in utilizing two of three modalities in the dataset and all of them in the end. As we can see, except the case of (A, V) in the CMU-MOSEI, our proposed method is superior to comparison methods in all cases. Specifically, when utilizing all views (the last columns of each dataset), our method boosts the performance compared with all other state-of-the-art methods. It demonstrates that the selected templates do offer extra discriminative information, and our model can effectively capture such information through the instance attention strategy, successfully promoting the performance. Also, in the scenario of only partial views being available (the rest columns of each dataset), our method still surpasses other methods in almost all cases, further showing its effectiveness. In addition, we also conduct experiments based on single view on all datasets, and the results are summarized in Table 4. Here, the 'Baseline' means that only the view-specific fully connected networks (i.e., $\{\boldsymbol{g}^v\}_{v=1}^V$) are utilized, and for our 'MvIAFN(single)' only the instance attention within a single view is available. Compared with 'Baseline', our 'MvIAFN(single)' still boosts the performance even in the single-view scenario, which further proves that the instance attention does capture the additionally valuable information for classification.

4.4. Ablation study

To demonstrate the significance of our proposed strategy, the ablation studies are conducted, as listed in Table 5. Note that, here we exploit all views in our experiments.

Obviously, directly concatenation does not meet our requirement. Making a comparison between 'MvIAFN(self)' and 'MvIAFN(cross)', it is easy to observe that by applying the instance attention across multiple views, there is a performance improvement, indicating the importance of the instance based information interaction among various views. Furthermore, compared with the cases when instances are randomly selected for templates, making clustering for the template collection does contribute to our classification. Obviously, some outliers which make an inferior influence on our instance attention strategy can be well removed by grouping instances from the same category and taking instances near the corresponding centroids as templates. Additionally, we also remove the alternative strategy for updating templates in our training phase (cluster at begin) and the classification performance meets the degradation. Thus, it is quite significant to re-collect the templates according to the learned features. As for objective functions, it is necessary to enable samples to pay more attention to the

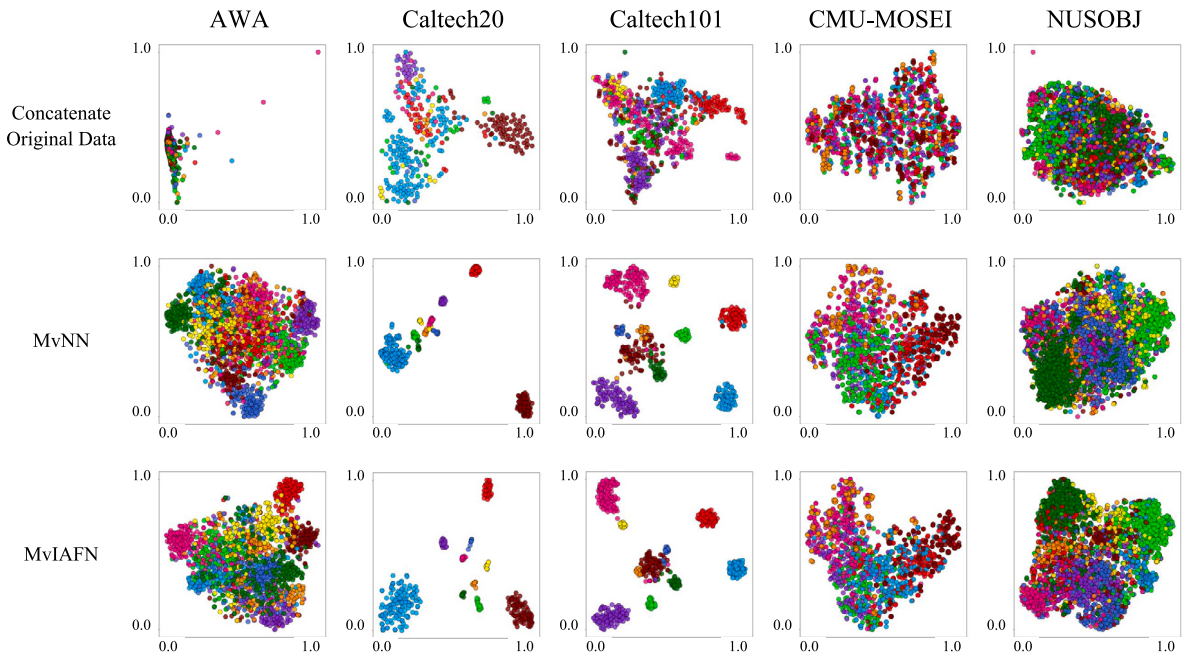

Fig. 4. Visualization of embedding feature spaces using t-SNE on five datasets. The first row is simply concatenating original multi-view features. The second and the third rows are obtained by MvNNcor [19] and our MvIAFN, respectively.

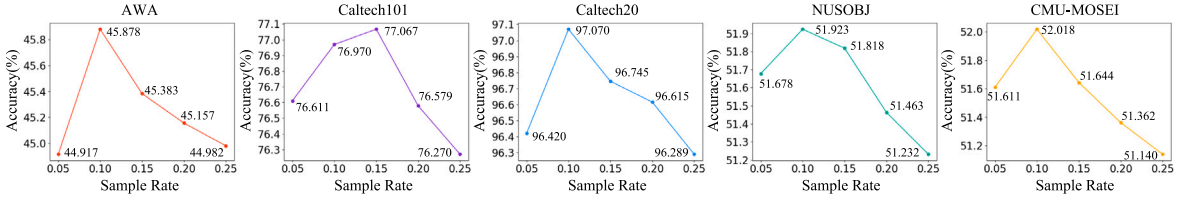

Fig. 5. The performance of MvIAFN with respect of different sample rates ξ on all the datasets.

templates under the same category. Otherwise, the templates from other categories may pull a sample away from its corresponding class centroid. Moreover, the mutual learning strategy does help the view-specific representations learn from each other and achieve to enjoy view-consistency. Once we successively remove the L_s and L_{ml}, the classification performance deteriorates continuously. Thus, it is significant to introduce the similarity loss and mutual learning loss into our proposed method. Eventually, we replace the clustering method DBSCAN in our template selection algorithm with K-means. In detail, for all instances from the vth view and the cth class, the K-means will divide them into 4 clusters, then we select several instances from each cluster that are near the centroid to form the templates set. Compared with K-means, the template selection algorithm with DBSCAN preserves the superiority. In other words, the K-means method is incapable of filtering the outliers among all instances, suffering from the inferior influence that is similar to the random selection strategy of templates.

Moreover, Fig. 4 visualizes the embedding feature spaces learned by MvNNCor and our MvIAFN using t-SNE. To show the feature spaces more clearly, for datasets with more than 10 categories (e.g., AWA), we only visualize the features from top-10 categories with the highest number of samples. Intuitively, simply concatenating the original features cannot separate different classes effectively. Besides, compared with MvNNCor, classes with our MvIAFN are more compact and separable, indicating that the learned features are more discriminative.

4.5. Parameter analysis

Finally, we explore the performance of MvIAFN with respect of different sample rate ξ, as shown in Fig. 5. It is noticeable that MvIAFN obtains the best top-1 classification accuracy on all the datasets when the sample rate ξ lied in $[0.1, 0.15]$. Intuitively, when ξ is too small, the amount of valuable information that carried by templates is limited, so that the attention vector $\mathbf{a}_i^{v,f}$ may lose some information which contributes to the performance improvement. Besides, when ξ is too large, it is inevitable to sample some outliers, which may cause an inferior influence on the attention vector generation. Particularly, for the categories with a small number of samples, the number of selected templates is also relatively small. In this case, a few of outliers can cause significant noise in the generated attention vectors and lead classification performance to meet the degradation.

5. Conclusion

In this paper, a novel multi-view method MvIAFN is proposed for classification. Different from existing methods which only exploit the correlation among multiple views from a same instance, our proposed method introduces the instance learning, so that the discriminative information between the input instance and template samples from a view of different categories is extracted by following an attention strategy. To further link one view to the remaining views, we also extend the attention to a multi-view version, allowing us to obtain the fused feature for each view. Experimental results demonstrate the superiority of our presented approach in comparison to state-of-the-arts.

CRediT authorship contribution statement

Jinxing Li: Ideas, Model design, Experimental design, Writing – original draft. **Chuhao Zhou:** Code implementation, Validation, Data

curation, Analysis. **Xiaoqiang Ji:** Model design, Data curation, Analysis, Reviewing and editing. **Mu Li:** Ideas, Experimental design, Data curation, Analysis. **Guangming Lu:** Experimental design, Reviewing and editing. **Yong Xu:** Experimental design, Reviewing and editing. **David Zhang:** Conceptualization, Oversight and leadership responsibility for the research activity planning and execution.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was partially supported by the NSFC fund (62272133, 62176077, 62102339), Shenzhen Colleges and Universities Stable Support Program (GXWD20220811170100001), Shenzhen Science and Technology Program (RCBS20200714114910193, RCBS20210706092219050), Guangdong Basic and Applied Basic Research Foundation (2022A1515110411, 2023A1515012883), Shenzhen Institute of Artificial Intelligence and Robotics for Society (AC01202201001), Shenzhen Key Technical Project (2020N046), and Shenzhen Fundamental Research Fund (JCYJ20210324132210025).

References

[1] X. He, S. Yan, Y. Hu, P. Niyogi, H.-J. Zhang, Face recognition using laplacianfaces, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 328–340.
[2] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 815–823.
[3] R. Cappelli, M. Ferrara, D. Maltoni, Minutia cylinder-code: A new representation and matching technique for fingerprint recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (12) (2010) 2128–2141.
[4] D. Zhang, F. Liu, Q. Zhao, G. Lu, N. Luo, Selecting a reference high resolution for fingerprint recognition using minutiae and pores, IEEE Trans. Instrum. Meas. 60 (3) (2010) 863–871.
[5] H. Shao, D. Zhong, X. Du, Efficient deep palmprint recognition via distilled hashing coding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.
[6] S. Zhao, J. Wu, L. Fei, B. Zhang, P. Zhao, Double-cohesion learning based multiview and discriminant palmprint recognition, Inf. Fusion 83 (2022) 96–109.
[7] J. Li, H. Yong, B. Zhang, M. Li, L. Zhang, D. Zhang, A probabilistic hierarchical model for multi-view and multi-feature classification, in: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32. No. 1, 2018.
[8] J. Li, B. Zhang, D. Zhang, Shared autoencoder Gaussian process latent variable model for visual classification, IEEE Trans. Neural Netw. Learn. Syst. 29 (9) (2017) 4272–4286.
[9] J. Li, B. Zhang, G. Lu, D. Zhang, Generative multi-view and multi-feature learning for classification, Inf. Fusion 45 (2019) 215–226.
[10] Q. Zheng, J. Zhu, Z. Li, Z. Tian, C. Li, Comprehensive multi-view representation learning, Inf. Fusion 89 (2023) 198–209.
[11] P. Li, S. Chen, Shared Gaussian process latent variable model for incomplete multiview clustering, IEEE Trans. Cybern. (99) (2018) 1–13.
[12] J. Li, G. Lu, B. Zhang, J. You, D. Zhang, Shared linear encoder-based multikernel Gaussian process latent variable model for visual classification, IEEE Trans. Cybern. 51 (2) (2019) 534–547.
[13] J. Ma, L. Tang, F. Fan, J. Huang, X. Mei, Y. Ma, SwinFusion: Cross-domain long-range learning for general image fusion via swin transformer, IEEE/CAA J. Autom. Sin. 9 (7) (2022) 1200–1217.
[14] L. Tang, Y. Deng, Y. Ma, J. Huang, J. Ma, SuperFusion: A versatile image registration and fusion network with semantic awareness, IEEE/CAA J. Autom. Sin. 9 (12) (2022) 2121–2137.
[15] Z. Chen, L. Fu, J. Yao, W. Guo, C. Plant, S. Wang, Learnable graph convolutional network and feature fusion for multi-view learning, Inf. Fusion 95 (2023) 109–119.
[16] D.R. Hardoon, S. Szedmak, J. Shawe-Taylor, Canonical correlation analysis: An overview with application to learning methods, Neural Comput. 16 (12) (2004) 2639–2664.
[17] P.L. Lai, C. Fyfe, Kernel and nonlinear canonical correlation analysis, Int. J. Neural Syst. 10 (05) (2000) 365–377.
[18] G. Andrew, R. Arora, J. Bilmes, K. Livescu, Deep canonical correlation analysis, in: International Conference on Machine Learning, 2013, pp. 1247–1255.
[19] J. Xu, W. Li, X. Liu, D. Zhang, J. Liu, J. Han, Deep embedded complementary and interactive information for multi-view classification, in: AAAI, 2020, pp. 6494–6501.
[20] L. Su, C. Hu, G. Li, D. Cao, MSAF: Multimodal split attention fusion, 2020, arXiv preprint arXiv:2012.07175.
[21] M. Hou, Z. Zhang, Q. Cao, D. Zhang, G. Lu, Multi-view speech emotion recognition via collective relation construction, IEEE/ACM Trans. Audio Speech Lang. Process. 30 (2021) 218–229.
[22] Z. Han, C. Zhang, H. Fu, J.T. Zhou, Trusted multi-view classification with dynamic evidential fusion, IEEE Trans. Pattern Anal. Mach. Intell. (2022).
[23] Q. Zhang, S. Yu, J. Xin, B. Chen, Multi-view information bottleneck without variational approximation, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2022, pp. 4318–4322.
[24] Y. Hao, X.-Y. Jing, R. Chen, W. Liu, Learning enhanced specific representations for multi-view feature learning, Knowl.-Based Syst. (2023) 110590.
[25] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 6 (1) (1991) 37–66.
[26] L.E. Peterson, K-nearest neighbor, Scholarpedia 4 (2) (2009) 1883.
[27] Y. Li, M. Yang, Z. Zhang, A survey of multi-view representation learning, IEEE Trans. Knowl. Data Eng. 31 (10) (2018) 1863–1883.
[28] M. Kan, S. Shan, H. Zhang, S. Lao, X. Chen, Multi-view discriminant analysis, IEEE Trans. Pattern Anal. Mach. Intell. 38 (1) (2015) 188–194.
[29] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory 13 (1) (1967) 21–27.
[30] H. Ouchi, J. Suzuki, S. Kobayashi, S. Yokoi, T. Kuribayashi, R. Konno, K. Inui, Instance-based learning of span representations: A case study through named entity recognition, 2020, arXiv preprint arXiv:2004.14514.
[31] M. Haddad, V.K. Ghassab, F. Najar, N. Bouguila, Instance-based learning for human action recognition, in: 2020 IEEE International Conference on Systems, Man, and Cybernetics, SMC, IEEE, 2020, pp. 147–153.
[32] S. Sun, J. Shawe-Taylor, Sparse semi-supervised learning using conjugate functions, J. Mach. Learn. Res. 11 (2010) 2423–2455.
[33] S. Sun, W. Dong, Q. Liu, Multi-view representation learning with deep gaussian processes, IEEE Trans. Pattern Anal. Mach. Intell. 43 (12) (2020) 4453–4468.
[34] M.M. Dunlop, M.A. Girolami, A.M. Stuart, A.L. Teckentrup, How deep are deep Gaussian processes? J. Mach. Learn. Res. 19 (54) (2018) 1–46.
[35] C.-H. Chen, Feature selection for clustering using instance-based learning by exploring the nearest and farthest neighbors, Inform. Sci. 318 (2015) 14–27.
[36] N. Zhang, S. Sun, Multiview unsupervised shapelet learning for multivariate time series clustering, IEEE Trans. Pattern Anal. Mach. Intell. 45 (4) (2022) 4981–4996.
[37] V. Mnih, N. Heess, A. Graves, et al., Recurrent models of visual attention, in: Advances in Neural Information Processing Systems. Vol. 27, 2014.
[38] D. Bahdanau, K.H. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: 3rd International Conference on Learning Representations, ICLR 2015, 2015.
[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems. Vol. 30, 2017.
[40] A. Gandhi, K. Adhvaryu, S. Poria, E. Cambria, A. Hussain, Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions, Inf. Fusion 91 (2023) 424–444.
[41] S. Zhang, M. Chen, J. Chen, F. Zou, Y.-F. Li, P. Lu, Multimodal feature-wise co-attention method for visual question answering, Inf. Fusion 73 (2021) 1–10.
[42] X. Wang, Y. Feng, R. Song, Z. Mu, C. Song, Multi-attentive hierarchical dense fusion net for fusion classification of hyperspectral and LiDAR data, Inf. Fusion 82 (2022) 1–18.
[43] A. de Haro-García, G. Cerruela-García, N. García-Pedrajas, Instance selection based on boosting for instance-based learners, Pattern Recognit. 96 (2019) 106959.
[44] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al., A density-based algorithm for discovering clusters in large spatial databases with noise, in: KDD. Vol. 96. No. 34, 1996, pp. 226–231.
[45] C.H. Lampert, H. Nickisch, S. Harmeling, Learning to detect unseen object classes by between-class attribute transfer, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 951–958.
[46] C. Kemp, J.B. Tenenbaum, T.L. Griffiths, T. Yamada, N. Ueda, Learning systems of concepts with an infinite relational model, in: AAAI. Vol. 3, 2006, p. 5.
[47] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y.-T. Zheng, NUS-WIDE: A real-world web image database from national university of Singapore, in: Proc. ACM Conf. Image Video Retrieval, CIVR'09, Santorini, Greece, July 8-10, 2009.
[48] L. Fei-Fei, R. Fergus, P. Perona, Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories, in: 2004 Conference on Computer Vision and Pattern Recognition Workshop, IEEE, 2004, p. 178.

[49] D. Dheeru, E.K. Taniskidou, UCI Machine Learning Repository, Irvine, CA, USA, 2017.
[50] M.R. Amini, N. Usunier, C. Goutte, Learning from multiple partially observed views-an application to multilingual text categorization, in: Advances in Neural Information Processing Systems, 2009, pp. 28–36.
[51] A. Zadeh, P. Pu, Multimodal language analysis in the wild: Cmu-mosei dataset and interpretable dynamic fusion graph, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Long Papers, 2018.