Received: December 12, 2022. Revised: February 22, 2023.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Pose Based Multi View Sign Language Recognition through Deep Feature Embedding

Ashraf Ali SK[1] **Prasad M.V.D[1]** **Hima Bindu G[2]** **P. Praveen Kumar P[3]**
Anil Kumar D[4]* **Kishore P.V.V[1]**

[1]*Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, 522302, India*
[2]*Department of Artificial Intelligence and Data Science (AIDS), Vignan's Institute of Information Technology, Duvvada, 530049, Visakhapatnam, India*
[3]*Department of Information Technology, Vignan's Institute of Information Technology, Duvvada, 530049, Visakhapatnam, India*
[4]*Department of Electronics and Communication Engineering, PACE Institute of Technology and Sciences, Ongole, Andhra Pradesh, 523272, India*
* Corresponding author's Email: danilmurali@gmail.com

Abstract: Sign language recognition in real time has been leveraged by the continuously varying hand movements in both shape and orientation across Spatio-Temporal dimensions. This is accomplished by either independent view or shared view feature learning. However, information movement between views is neither total nor restricted to achieve view insensitivity during testing where all views are needed. The objective is to perform pose based multi view sign language recognition by applying triplet loss on a pair of specific and shared view features. Shared view features are obtained using view compatibility matrix which maps within class between view features and between class within view features. This mapping helps in increasing information flow between views from the same class and restricting it between classes thereby making highly discriminative feature representation for all views. Subsequently, metric learning enables to build a view invariant feature embedding by stacking view specific and shared features from different layers for training deep models. In the end, a blended view feature representation is obtained per class. Experiments were designed on our multi view skeletal sign language video dataset and three benchmark action datasets. The results of the experimentation have shown that the performance of the classifier has improved by 8% over the linear view combiners such as Laplacian eigenmaps. Further, the proposed model is useful in constructing a view invariant feature for recognition of multi view sign language.

Keywords: 3D video analysis, Triplet loss, Multi view spatio temporal features, Sign language recognition.

1. Introduction

Sign language recognition (SLR) is an automated machine learning system for the classification of visual information of the human signer into text or voice commands. The visual inputs are in either 2 or 3-dimensional spatiotemporal information [1]. Generally, sign language is a visual form of communication between the hearing impaired or hard hearing people. The language corpus is made from finger and hand movements with respect to the face, head and upper torose of the signer. Particularly, the SLR models used in machine translation used RGB video frames as input. Specifically, some cases considered depth and skeletal sequences for recognition [2]. Appearance features were extracted from RGB and depth sequences whereas pose-based features were operated upon for classification [3]. Unquestionably, the RGB is considered the primary choice as input to the machine interpreters due to availability and cost. However, the results show that the machine learning models find it difficult to extract

Received: December 12, 2022. Revised: February 22, 2023.

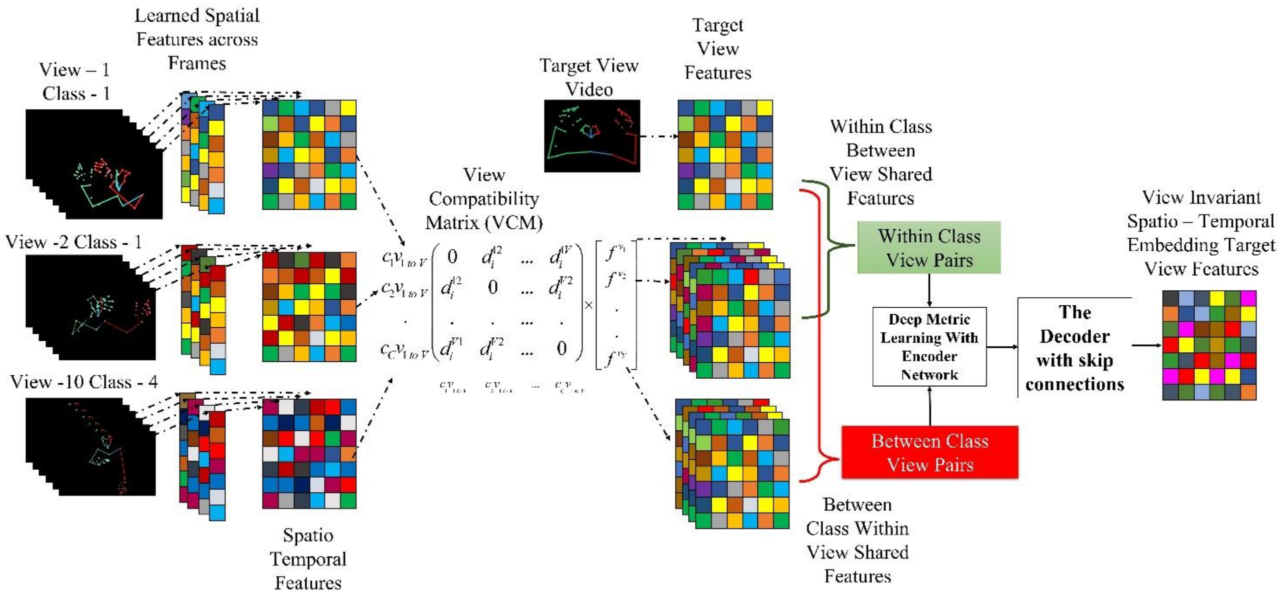
Learned Spatial Features across Frames
View – 1 Class - 1
View -2 Class - 1
View -10 Class - 4
Spatio Temporal Features
View Compatibility Matrix (VCM)
$c_1 v_{1\ to\ V}$ $c_2 v_{1\ to\ V}$ $c_C v_{1\ to\ V}$
0 d_i^{12} ... d_i^{1V}
d_i^{12} 0 ... d_i^{V2}
d_i^{V1} d_i^{V2} ... 0
×
f^{v_1} f^{v_2} f^{v_V}
Target View Video
Target View Features
Within Class Between View Shared Features
Within Class View Pairs
Deep Metric Learning With Encoder Network
The Decoder with skip connections
View Invariant Spatio – Temporal Embedding Target View Features
Between Class View Pairs
Between Class Within View Shared Features

Figure. 1 Illustration of the proposed view invariant feature generation process

features from low-resolution finger data and their nonlinear movement in space and time [4]. Whereas these problems can be easily mitigated in the 3D model-based approaches where the signs are represented as skeletal data. The 3D skeletal information can be characterized as multi-dimensional vectors, images, and RGB video data. Any of these formats can be used for developing a pose-based automated SLR [5]. Consequently, applying skeletal 3D video data in the development of a real time sign language interpreter comes with many synchronization challenges.

To conciliate the above research challenges, we propose to learn viewpoint features as well as shared features simultaneously resulting in blended Spatio-Temporal feature. Here viewpoint features are specific features obtained from each of the views that carry information representing one view in a class. The shared features are constructed from within class between views and between class within views with the help of a proposed mapping function called as view compatibility matrix (VCM). The within class between view features define the similarities in multiple views from the same class. The VCM mapping function is a diagonal matrix that gives the mean distance coefficients between views. Specifically, VCM gives the percentage of similarly metric between views of a class. Correspondingly, the between class within view features define the similarities between classes which must be restricted for effective discrimination of class labels.

Hence to generate a maximally discriminating spatio temporal representation for a specified target view, there should be efficient transfer of information between specific and shared view features from across the dataset. Consequently, this is accomplished through deep metric learning (DML) in this work. The DML model operates on pairs of specific and shared views using triplet loss embedding. Admittedly, the output of DML is a feature representation for the target class that is closely associated with all the within view between class features and loosely associated between class within views. Finally, the resulting feature vector is decoded using the skip connections from the target stream in DML model to reconstruct blended features. The obtained blended features are highly discriminating and extremely expressive. These Spatio-Temporal features are constructed from multiple layers in the CNN architecture which are bonded together into a training dataset. We call our proposed method as deep triplet encoder decoder (DTED). Any deep learning model trained on the generated singular view blended Spatio-Temporal feature set per class is enough to test the previously unseen view within the class. The primary advantage of our work is to achieve high accuracy with single view testing which was highly unlikely on the previous models. Fig. 1 shows the illustration of the proposed view invariant feature generation process.

The proposed method has been investigated on our 3D skeletal video datasets of sign language (KLEF3DSL_2Dskeletal) [6] and four other multiview action datasets NTU RGB-D [7], SBU Kinect Interaction [8], KLYoga3D [9] and KL3D_MVaction [10]. Subsequently, we validated the proposed method against the state – of – the – art on these multi view datasets. The rest of the paper is organized into four sections. The literature related to multi view methods is highlighted in section two.

Section three consists of methodology and the experimentation is provided in section four. Finally, the last section draws conclusion on the results obtained in section four.

2. Literature review

This section gives an extensive analysis of the multi view deep learning models for sign language and human action recognition. Furthermore, the advantages and disadvantages of each of the previous methods has been discussed.

The deep learning revolution has seen an exponential growth of 2D video based SLR models, which has transformed from feature extraction to feature learning paradigms [11]. On one hand, these models reported the highest possible accuracies and on the other hand it has become difficult to get generalized on the inputs with multiple signers and viewing angles. This problem found the solution in the form of higher dimensional datasets such as depth and 3D skeletal representations. Admittedly these models have shown exceptional performance with multi stream convolutional neural networks for action recognition [12]. In comparison to single modal datasets, the multi modal data has been shown to establish higher recognition accuracies. Usually, the computational power required for training on multi modal datasets is on the higher side when compared to single-modal datasets.

Correspondingly, to develop a real time deployable sign language machine translator it is necessary to train the model with multi view datasets. Following this has seen an incremental surge in the use of multi view human action datasets for training and testing the deep models [13]. The developments in this direction generated research related to dictionary learning [14, 15], artificial neural networks [16], convolutional neural networks [17] and deep attention networks [18]. Obviously, the attention mechanisms with deep networks have shown to learn specific features across views when compared to other models [19]. The attention models were able to produce good view-specific features but have failed to generate cross view features for classification. Moreover, the fusion of view specific and cross view features into a single view invariant feature has failed to capture most of the view variations in the multi view data [20].

The past works on multi-view can be classified as learning based and view invariant models. In multiview learning approaches, the machine learned time series representation of actions or signs in different views independently [21, 22, 23]. The methods generated combined view features based on low level observations of the spatial frames in the videos sequence [24]. Consequently, different training algorithms and network architectures have been proposed to learn a set of equivalent features between views [25, 26]. The canonical correlation coefficient (CCA) [27] and view projection matrices were used to extract relationships across views of a class [28]. These methods are further improved through matrix factorization [29] and low rank constrained matrix factorization [30] for finding similarities between views. The above works have shown to provide good recognition accuracies trained with limited number of views.

Alternatively, the above limitations were subjugated through mapping descriptors which transfer information between views. These mapping functions that have shown maximum robustness for action recognition applications are self-similarity matrix (SSM) [31] and sample affinity matrix (SAM) [32]. The SSM models summarize views across all class and transfer those similarities to all views during learning. However, the biggest disadvantage of SSM comes from the assumption that all views contribute equally to the shared features. This assumption has shown to have negative implications on the overall performance of the classifier as each view contributes differently to the viewpoint. Additionally, the intra class variations across views were ignored in SSM which define the dissimilarities between classes.

The above two problems were handled by sample affinity matrix (SAM) [32]. The SAM is a transformation matrix to generate the weighted similarities between views within class and dissimilarities between class within views. The mapping function was learned on autoencoder features at each level which are bonded together to generate a view invariant feature representation. The only drawback of this method is the use of regularizers for prevention of data transfer between target and shared views. The regularizer parameters have to randomly selected through experimentation to make the model efficient.

In this paper, we propose to learn the regularizer parameters through metric learning. Instead of performing regularization through random selection, we propose to learn the view invariant features by simultaneously attracting the within class between view features and pushing out the between class within view shared features. This process leverages highly discriminative Spatio-Temporal feature embedding space for skeletal video data. There is a threefold difference of the proposed method from works in literature. 1) The design of a view compatibility matrix which discovers the dependencies between shared features. 2) The proposed method has constructed a highly discriminative Spatio-Temporal features by metric learning on triplet pairs of target views and shared views. 3) Our model learns the target vectors using the decoder network with skip connections. The following objectives are formulated: 1. To map a view compatibility matrix for finding relationships between shared view features. 2. To learn mapping function for making a view invariant Spatio-Temporal feature matrix. 3. To test the trained model with only one view. We call our model deep triplet encoder decoder (DTED).

3. Multi view feature learning (MVFL): the deep triplet encoder decoder (DTED)

The objective is to construct a blended view feature by learning the relations between target and shared views across classes on skeletal sign language videos. This will enable inferencing the trained model with any one view as against all the views required in the previous models.

3.1 View compatibility matrix (VCM)

Based on the previous models such as self-similarity matrix (SSM) [31] and sample - affinity matrix (SAM) [32] for defining view transformations among classes, we propose view compatibility matrix (VCM). VCM measures the similarity between pairs of multiple views in skeletal sign videos. Given V skeletal sign views for training: $\{X^v, y^v\}_{v=1}^{V}$, consisting of C sign language classes. The c^{th}class in v^{th} view consists of S videos: $X^{vc} = [X^{v1}, X^{v2}, \ldots\ldots, X^{vC}] \in R^{f\times C}$, where f is the feature dimension per video sample. The corresponding class labels are $y^{vc} = [y^{v1}, y^{v2}, \ldots\ldots, y^{vC}]$. Similar to SAM, we construct VCM using the parametrized distance function between the features $F^{vc} = [f^{v1}, f^{v2}, \ldots\ldots\ldots, f^{vC}] \in R^{g\times N}$, where$N$is the number of frames in the skeletal video and g is the length of the feature vector. The block diagonal matrix $D = diag(D_1, D_2, \ldots\ldots, D_C)$ is defined as

$$D_i = \begin{array}{cc} \begin{array}{c} c_1v_{1_{to\,V}} \\ c_2v_{1_{to\,V}} \\ \vdots \\ c_Cv_{1_{to\,V}} \end{array} & \begin{pmatrix} 0 & d_i^{12} & \cdots & d_i^{1V} \\ d_i^{12} & 0 & & d_i^{2V} \\ & \vdots & \ddots & \vdots \\ d_i^{V1} & d_i^{V2} & \cdots & 0 \end{pmatrix} \\ & \begin{array}{cccc} c_1v_{1_{to\,V}} & c_2v_{1_{to\,V}} & \cdots & c_Cv_{1_{to\,V}} \end{array} \end{array} \tag{1}$$

The $D \in R^{CV\times CV}$ is a matrix giving similarities between all views within a class and also between class views. The d_i^{cv} is a distance of the i^{th} sample computed as

$$d_i^{cv} = \frac{exp\left(\|f_i^v - f_i^u\|^2\right)}{2C} \tag{2}$$

Where, (u, v) are view pointers in all classes. The features f exhibit spatio temporal characteristics of the skeletal sign in the video sequence.

The resulting block diagonal matrix $D_i$represent distances between features $f$of video samples across multiple views within the class and across the class. This gives similarity measure of within class between view features and between class within view features. Consequently, the within class between view similarity measure tells the appearance distinction between views and also the distance provides the amount of closeness between views. Similarly, the between class view feature distances account for the information that is invariably shared across signs for incorrect classification. To this extent the between class shared information has to be eliminated to generate a highly discriminative feature vector. Specifically, the VCM defines the combinations of shared features across views and classes. The objective is to effectively learn the right combination of shared features to generate a view invariant feature for a target class.

3.2 Spatio temporal feature extraction

Let $X^{vc} = (x_v = \{S_v\}\forall v = 1\ to\ V, c = 1\ to\ C)$ be Vviews of the video frames with $V \in R^3$. The deep learning model learns f^v features from $x_v$on labels y_v at specific views with θ_{fen} trainable parameters for Lloss function optimization on the dataset as

$$\theta_{fen} = arg \min_{\theta_{fen}} L\left(\theta_{fen}; x_v, y_v\right) \tag{3}$$

The trained model θ_{fen} has view specific features f_v at the output of the dense layers as

$$\{f_v\}_{v=\{1,N\}} = \sum_{i=1}^{I}\sum_{j=1}^{J} x_v(i,j) \\ * K(k-i, k-j)\forall k \in \text{kernel size} \tag{4}$$

Fig. 2 shows the features learning network. The network is built with four convolution layer pairs on top of the rectified linear activations and a 2 × 2 maximum pooling layer. The convolution layers use strides of one and the maximum pooling uses two. Subsequently, a batch normalization layer standardizes inputs of deeper layers. Lastly, fully

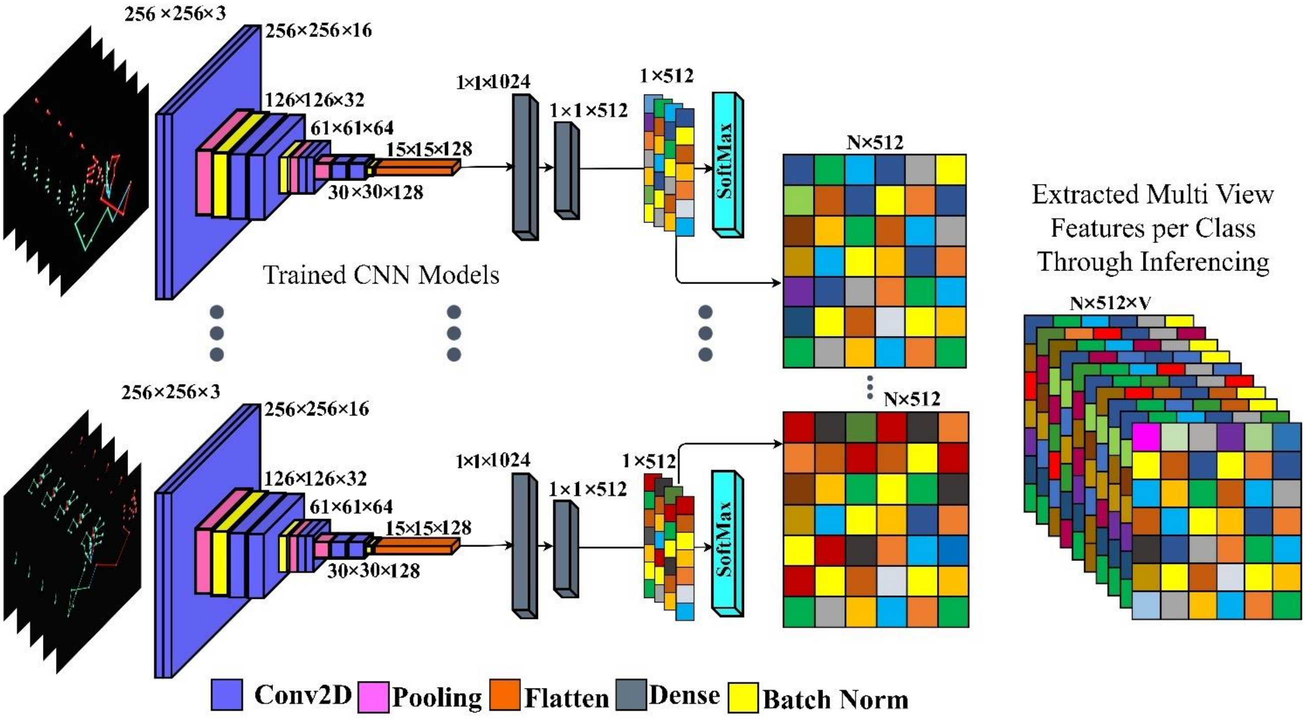

Figure. 2 Spatio temporal view feature extraction network

connected layers learn on the features generated by the convolutional layers. Then, a complete spatio temporal feature matrix representing the 2D skeletal video sequence is generated by concatenating the frame wise spatial features at the output of dense layers. Altogether, Vnetworks operate separately for producing view specific class features $F^{cv} = \{f_{ic}\}\ \forall i = 1 \text{ to } V \in R^{g \times N}$. Categorical cross entropy loss with stochastic gradient descent optimizer is applied during training on the entire dataset. The trained model $\widetilde{\theta_{fen}}$ is employed for all classes to obtain the features as

$$F^{cv} = \widetilde{\theta_{fen}}(w, b) \times X^{cv} \forall V \& C \in R^{g \times N} \qquad (5)$$

The spatio temporal feature matrix for all views in all classes are inputted to calculate the weighted dependencies of shared views against the target views.

Firstly, the input is a skeletal 2D video data obtained from mapping of 3D motion captured sign language data. The 2D skeletal video data is considered to achieve our long-term goal of finding a reliable relationship between 2D real time sign video data and 2D skeletal information for better recovery of spatial and motion information respectively. Secondly, the input training data for the network in Fig. 2 is 2D skeletal video frames in multiple views. This multi stream network is trained using supervised learning approach on individual views per stream. Inferencing is conducted with any view randomly from the set of trainable views and select the stream that generates maximum recognition rate. Subsequently, extract Spatio-Temporal features from this stream from multiple layers for further processing.

3.3 Shared feature factorization

All previous works had considered this as a linear combination of views and calculated the weights for individual views that re contributing to the target view using laplacian eigenmaps [31, 32]. The effectiveness of laplacian eigenmaps has been proved to provide excellent results when the number of views available for training is small (V=4). This is because of the computational complexities for classification of eigenmaps. Since, we have more than 15 views available for processing, we propose to learn these weights using triplet loss embedding based autoencoder. The proposed novel idea can help in training a database with large number of views. For experimentation, we selected mocap videos to reciprocate the large number of views which can be constructed with ease. However, the principal question worth answering is why video data of skeletal joints Instead of 3D joint information directly. The 3D joint data has minimalistic variations within views which cannot be rationalized into actual view representation in real time. Hence our project contains 3D skeletal representations of 2D videos where the view variations have considerable impact on the performance of the multi view training.

Given a set of learned spatio temporal features $F^{cv}\ \forall\ c = 1\ to\ C, v = 1\ to\ V$, the aim of this module is to generate a weighted combination of shared feature using VCM, the distance matrix D_i. The VCM consists of two types of shared features with one representing similarities between within class between view features $d_i^{cv} \forall c, v = 1\ to\ V$ and the other between class within view features $d_i^{cv} \forall v, c = 1\ to\ C$. The subscript i indexes the location of weights in D_i. Specifically, we calculate the relationships between all views from within a class and across classes using D_i. Given a class c with view v^{th} target features, the within class between views relationship R_{WCBV} is a weighted feature combination of all $(v-1)^{th}$ views defined by the $d_i^{cv} \forall c, v = 1\ to\ V$. The $R_{WCBV}$is calculated as

$$R_{WCBV} = D_i \times F^{cv}\ \forall\ i = index(diag(D)) \quad (6)$$

Explicitly the $R_{WVBC}$for a single class is shown as

$$R_{WCBV} = \begin{bmatrix} 0 & d_1^{12} & d_1^{13} & . & d_1^{1V} \\ d_1^{21} & 0 & d_1^{23} & . & d_1^{2V} \\ . & . & 0 & . & d_1^{3V} \\ . & . & . & 0 & . \\ d_1^{V1} & . & . & d_1^{V(V-1)} & 0 \end{bmatrix} \times \begin{bmatrix} f^{v_1} \\ f^{v_2} \\ . \\ . \\ f^{v_V} \end{bmatrix} \forall c = 1 \quad (7)$$

Consequently, the $R_{WCBV} \in R^{V\times 1}$ is a column vector representing the weight combination of a particular view against all other views within a class. The challenge is to select a particular weighted combination fromR_{WVBC}which closely matches the target view in a particular class. This was achieved by learning a set of shared and unshared features using a mapping matrix with two regularizes on the feature set [32]. Despite success, the method suffers from dependence on hyper parameters of regularizes. To overcome this disadvantage, we extract the second relationship R_{WVBC}, which defines the weighted combination of between class views as

$$R_{WVBC} = D_i \times F^{cv}\ \forall\ i = index(non\ diag(D)) \quad (8)$$

The $R_{WVBC} \in R^{1\times VC}$ is a feature space with common features between classes that has the ability to interfere during classification. Markedly, the view invariant feature for a particular class should be similar to R_{WCBV} and dissimilar to R_{WVBC}. As discussed previously, this was achieved through an objective function with two regularization terms [39]. In this work, we apply deep Triplet learning along with a set of encoder decoder network to learn a highly discriminative and robust view invariant feature per class.

3.4 Deep triplet view invariant feature learning

The proposed model is built on Triplet learning with single encoder decoder network. The architecture of the proposed model is illustrated in Fig. 3. Given the v^{th} target view features f^{vc} from a particular class c and the shared features (R_{WCBV}, R_{WVBC}), the objective is to learn a view invariant feature embedding. Firstly, the features are pre-processed by selecting the maximum operator and only 100 features per video frame are selected. Secondly, feature pairs are constructed as positive and negative sets. The positive set consisting of view specific target features and within class between view features. Subsequently, the negative features are paired as view specific targets and between class within view features. Thirdly, we apply the triplet loss embeddings on the paired features specified by a set of multi view training data $S = \{F_T^{iv}, y_i\} \forall i = 1\ to\ C, v = 1\ to\ V$with V views and C classes, deep Triplet encoder decoder (DTED) classifier focuses on learning a mapping function relating the target view features F_T^{iv} to y_i such that the predicted label $\hat{y}_i \rightarrow y_i$.

Lastly, a decoder is built alongside the encoder network with view specific feature inputs. The decoder reconstructs the encoded target features which are closely related to within class view features and distant to between class view features. This mapping is achieved by reducing the view specific triplet loss. The trained model D_{TL} extracts the maximally discriminant target features $f^{cv} \in R^d$in d dimensions being represented as

$$f^{cv} = D_{ML}(F_T^v, \theta_{ML}) \forall v = 1\ to\ V, c = 1\ to\ C \quad (9)$$

Here θ_{ML} consists of trained parameters of the model D_{ML} that extracts the d dimensional view invariant features per class. The D_{ML} model is trained in each iteration with a single triplet pair $t_z = (f_T^{cv}, R_{WCBV}, R_{WVBC})$ which is constructed by applying the following the condition $y_T = y_{WCBV} \neq y_{WVBC}$. Fig. 3 shows the deep network used for learning from t_z. The deep triplet encoder decoder

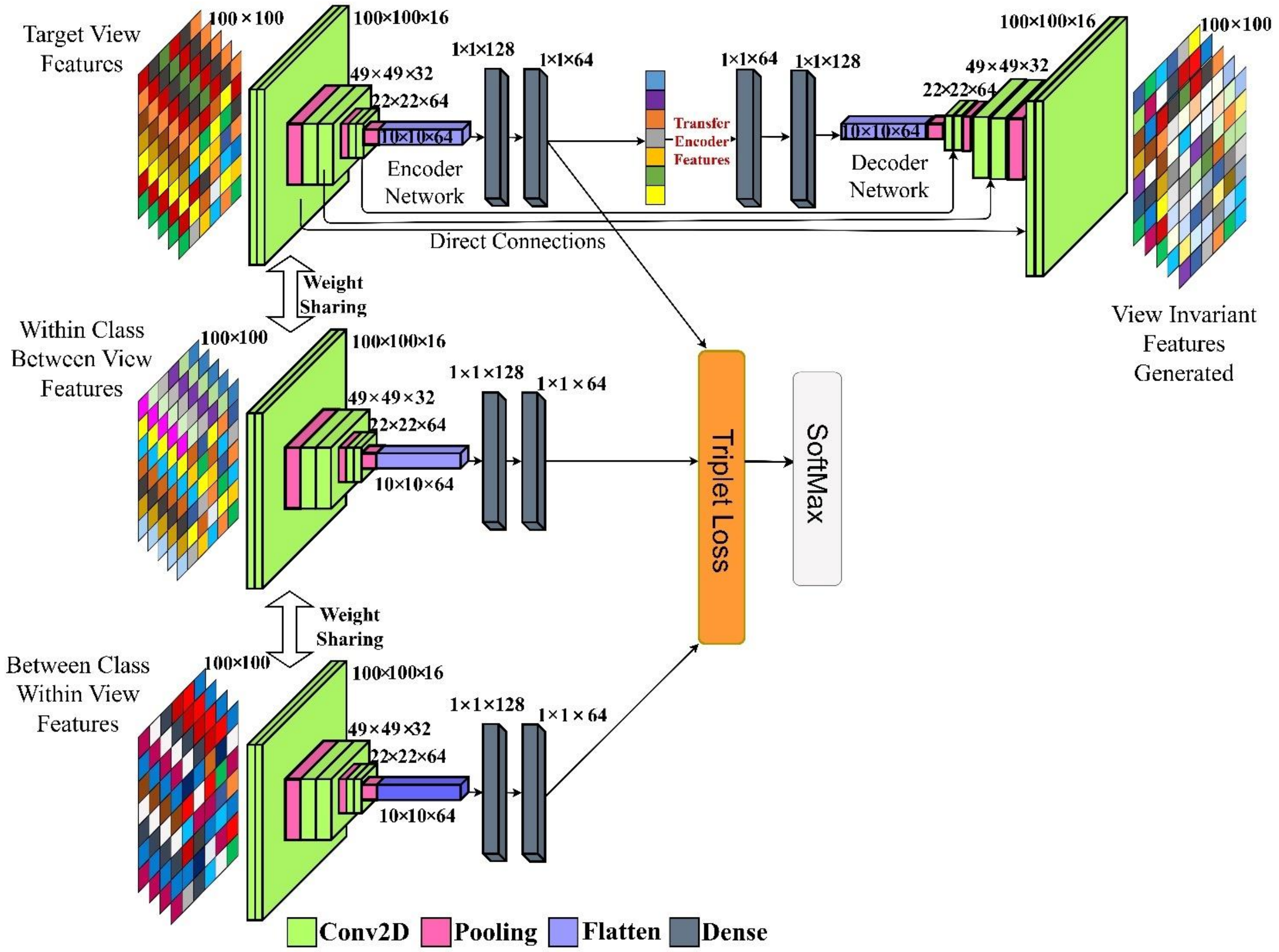

Figure. 3 Deep triplet encoder decoder (DTED) architecture for view invariant feature learning

learning (DTEDL) network learns the view mapping function through view specific loss computed on the feature embedding spacet_z.

The triplet loss functional $l_{triplet}$ is

$$l_{triplet}(t_z) = \sum\nolimits_{\forall V} h(\delta - \|f_T^z - R_{WCBV}^z\|^2 + \|f_T^z - R_{WVBC}^z\|) \quad (10)$$

Where δ is the allowable margin that marks the boundary to discriminate positive and negative pairs. Here $h(\quad) = max(,0)$ is the hinge loss. The triplet loss aims to rationalize the weight vectors in the direction dictated at maximizing the Triplets between (f_T^{cv}, R_{WVBC}) features and minimizing Triplets between (f_T^{cv}, R_{WCBV}) , respectively.

Consequently, the trained network on target view features is extracted for pairing with a decoder network to form an encoder decoder network D_{ED}. The encoder weights are transferred directly to the decoder side which enables for reconstruction of complete set of view invariant features for a particular target view within a specified class. Notably, view invariant features can be inferenced onD_{ED} for any set of target views from any class. Despite multiple networks and excessive trainings, view invariant features can be generated instantaneously by inferencing on the auto encoder for any number of untrained views within a class. Finally, these features can be learned by any deep neural network architecture for view invariant skeletal sign or action video classification.

3.5 The classification network

The classification process is designed in Fig. 4. Any CNN architecture can be trained on the view invariant features in a class label for classification. The CNN model is trained with categorical cross entropy loss function and stochastic gradient descent optimizer. The proposed methodology has dual advantages in the form of marginal computational complexity and single view inferencing. Moreover, the generated features are packed with both spatial and temporal information.

The following procedure is instigated to train the proposed networks. First, view specific features for all classes are extracted through the network in Fig. 2. There are 100 frames in each video sample. From

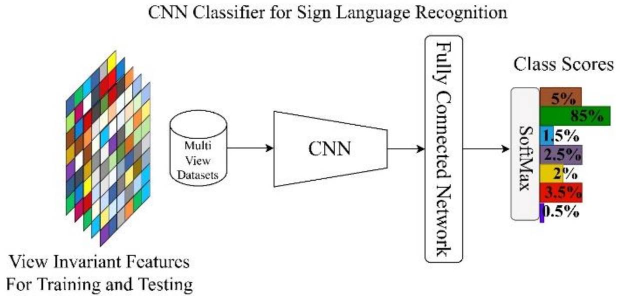
CNN Classifier for Sign Language Recognition
Multi View Datasets
CNN
Fully Connected Network
SoftMax
Class Scores
5%
85%
1.5%
2.5%
2%
3.5%
0.5%
View Invariant Features For Training and Testing

Figure. 4 The classification process

each frame 100 features are selected from the output of the dense with 512 values per frame. The top 100 feature are the values greater than the mean of the 512 values. The reason behind selecting 100 features out of 512 is experimental in nature. Since the skeletal representation in video sequences occupied minimum pixel density, it facilitates highly concentrated features with minimalistic representation. After multiple attempts to find the right feature length for maximum performance, we arrived at 100 features per frame. Importantly, the position of the selected features was unaltered in the final representations. The learning rate for this network was fixed at 0.001. The extracted spatial features are concatenated to form a spatio temporal feature matrix characterizing each video sample. Secondly, VCM is calculated on the view specific spatio temporal features to determine the weighted combination of shared features across views and classes. Thirdly, Triplet auto encoder learning model in Fig. 3 is trained to extract view invariant features across views in each class. Here, the network is trained with triplet loss first and then, decoder is added to extract the features. A learning rate of 0.00001 is selected initially, which was the progressively regularized with a decay of 0.1 whenever the error became constant. Finally, these view invariant features are used for classification using as shown in Fig. 4. The learning rate of the network in Fig. 4 is selected as 0.01. All the models were trained with Adam optimizer on 8GB NVDIA RTX 1070x GPU with 16GB memory using TensorFlow 2.5 APIs. Subsequent sections provide a detailed description of the results obtained through rigorous experimentation on various skeletal video datasets to evaluate the performance of the proposed method against similar frameworks.

4. Experimentation

The proposed deep triplet encoder decoder (DTED) has been trained and tested on multi view skeletal sign (action) video datasets with multiple train test splits. The experimentation was conducted with a one – to – one, one – to – many, many – to – one and many – to – many cross view training and testing approaches on DTED. Additionally, the findings of DTED were validated against the other state – of – the – art multi view methods. Ultimately, to check the robustness of the proposed feature extraction process multiple CNNs architectures were tested for generating view invariant features.

4.1 Skeletal video datasets and evaluation metrics

To experiment with the proposed methodology, we start with our multi view sign language dataset KLEF3DSL_2Dskeletal with $V = 15$ views, 200 classes. The dataset is produced at KL biomechanics and vision computing research centre utilizing 3D motion capture technology [6]. Additional to our multi view sign language data, we evaluated the methods proposed on multi view benchmark skeletal action datasets such as NTU RGB-D [7], SBU Kinect Interaction [8], KLYoga3D [9] and KL3D_MVaction [10]. Fig. 5 shows data sample subset from KLEF3DSL_2Dskeletal for a sign basketball. The train test ratios are maintained stable through the entire experimentation phase. The preferred ratios are one – to – one and one – to – many. Because there are not any multi view sign language datasets, we estimated our model on multi view benchmark action datasets. Even though there are enormous action dataset classes, we preferred just 40 action classes for experimentation in 15 views per class for preserving homogeneity throughout comparison. Random views were generated through rotation of original skeletal data to create more views in some databases used in this paper. For example, NTU RGB-D dataset has only 8 views. Specifically, our sign language data has 15 views. To compensate for the remaining views in NTU RGB-D, we rotated the skeleton by angles that in-between the available views. All the rotations were performed on the front view skeleton and then it was transformed into a continuous video sequence. The predominant rotation angles were +-10,+-15,+-20,+-25,+-30,+-40 and +-45 degrees. Here, the assessment is accomplished unbiased of the nature of view in which the action is filmed. Fig. 6 (a). shows examples from NTU RGB-D dataset. Fig. 6 (b) shows samples from KL3D_MVaction and Fig. 6 (c) shows multi view samples from KLYoga3D dataset. Mean recognition accuracy (mRA) has been used as performance evaluator throughout this work.

The Fig. 2 network has been trained on all the views of sign (action) skeletal videos with analogous hyper parameters with the exception of learning rate and epochs. The learning rate for KLEF3DSL_2Dskeletal sign language video dataset is 0.001 and it was 0.005

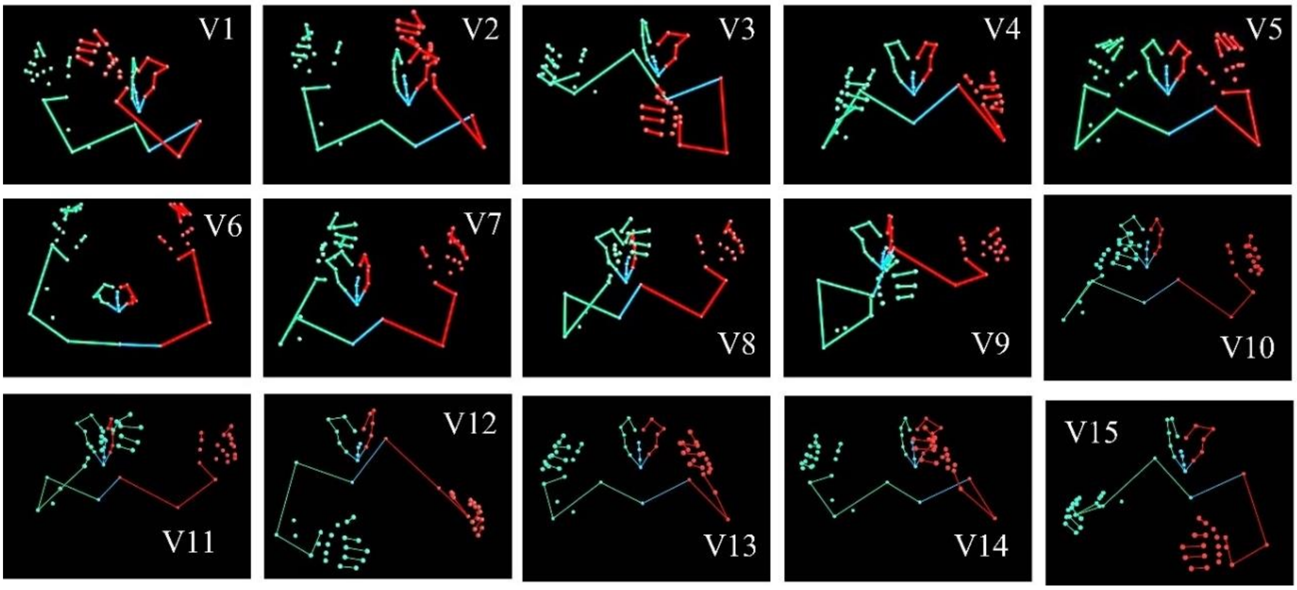

Figure. 5 An example frame in 15 different views for the skeletal video sign "Basketball" from KLEF3DSL_2Dskeletal sign language video dataset

for all other action datasets.

Nevertheless, the KLYoga3D have being trained on a learning rate of 0.0001 for 300 epochs owing to substantial number of skeletal joints. The residual datasets were trained for 200 epochs. The trained models are inferenced by the training sets for feature extraction. Consequently, the dense layer features were concatenated to construct a spatio temporal feature representation for each of the views in a class. Interestingly, the significance of features across other layers is also evaluated from the model in Fig. 2. The features obtained on the trained CNN model in Fig. 2 are termed as view specific features. To generate shared view features across multiple classes, VCM is applied. The VCM generates a set of within view between class R_{WVBC} and within class between view features R_{WCBV} . The connection between view specific and shared features is critical in creating view invariant features, which are learned using the proposed deep triplet encoder decoder (DTED) architecture in Fig. 3. The entire model was trained using triplet loss embeddings with different δvalues without the decoder network. The hyperparameter (δ)for DTED on KLEF3DSL_2Dskeletal($\delta = 0.28$), NTU RGB-D($\delta = 0.35$), SBU Kinect Interaction ($\delta = 0.37$) , KLYoga3D ($\delta = 0.38$) and KL3D_MVaction ($\delta = 0.29$) is selected iteratively. After successful training of the deep Triplet network, the view specific trained parameters were transferred directly to the decoder network. The deconvolution on the encoder features in decoder network in successive layers has generated view invariant features that have high correlativity within class views and disassociation with between class views. Overfitting on datasets is avoided by setting the training stop loss at 0.001 across all datasets. Finally, these created view invariant features are applied for classification. Exclusively, the obtained skeletal feature robustness for classification is tested by regulated training and inferencing on standard CNN models. However, these models are diminished in layers and depth to source a size of 100×100 to avoid vanishing gradients. To corroborate the actual effectiveness the view invariant features, multiple evaluation processes on the classifier are presented.

4.2 One – to – one classifier evaluation

The one – to – one cross view identification test is organised by training the classifier in Fig. 4. in conjunction with one view invariant feature on behalf of all views and testing on separate views. Exclusively, the crucial standpoint of this is to test the strength of the produced view invariant features in predicting a particular class based on its fundamental views on which it is created. To establish this, we constructed the deep network virtuoso with VGG-16 consisting of six convolutional layers, three maximum pooling, one flatten and two dense layers. The produced view invariant features in each class are used for training the deep network and subsequently it is tested on view specific features. Accordingly, we preferred the learning rate of 0.0001 for the network and is trained using categorical cross entropy loss and Adam optimizer. Later, the overhead process is repeated for all datasets with the identical hyper parameters. Moreover, three baseline architectures such as Inception – V4, GoogleNet and ResNet – 50 have been used for training and testing.

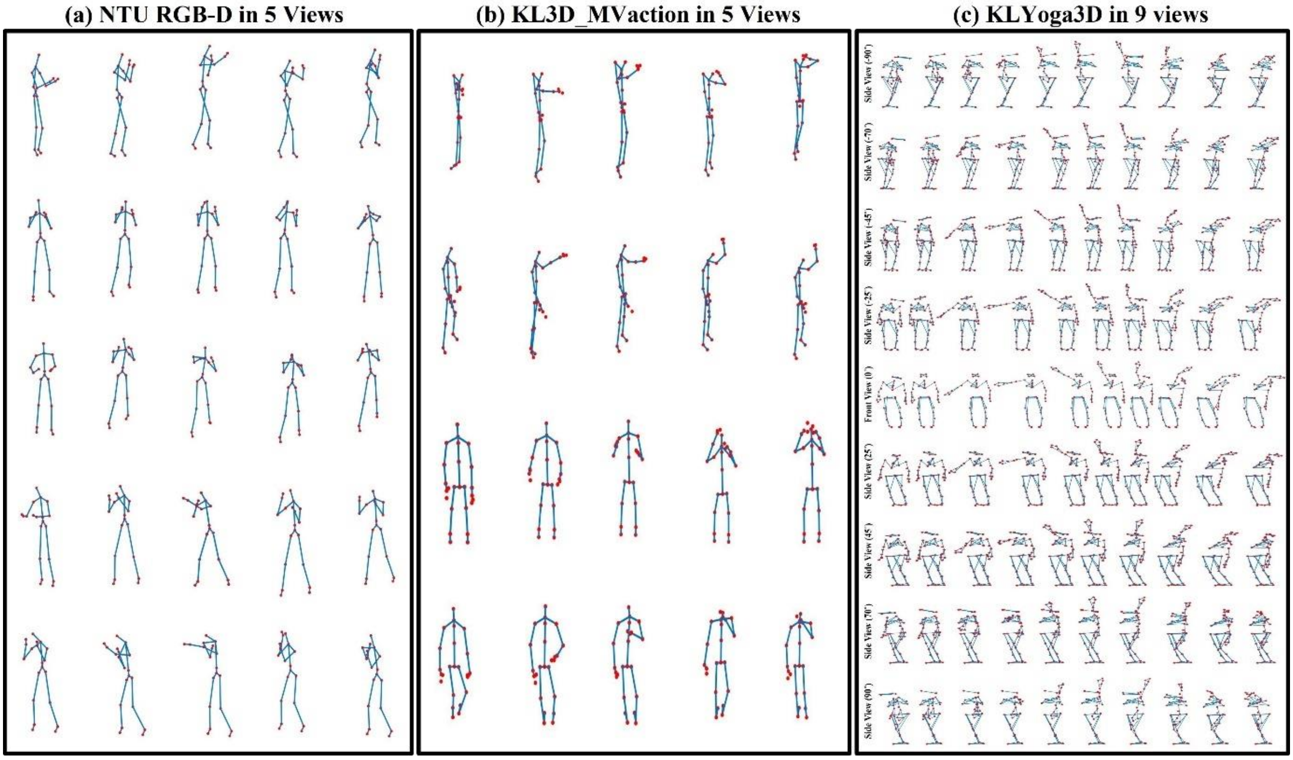

Figure. 6 Action datasets in multiple views used as benchmarks for evaluating the models: (a) NTU RGB-D in 5 views, (b) KL3D_MVaction in 5 views, and (c) KLYoga3D in 9 views

Then again vanishing gradients and overfitting difficulty were reduced by re-constructing the architectures with only one-half the layers than the initial models. The composition of the initial models was well-preserved to accomplish peak performance. Ultimately, mRA is calculated throughout inferencing and the 10-fold highest value is given in table 1 for all the datasets.

After analysing the mRA in Table 1, it is obvious that the models operate perfectly on test views that have additional visible knowledge when compared to views consisting of intersecting joints. The results from Table 1 too implies that the view invariant features have appeared to lessen false positives in all classes. Remarkably, the DTED has produced extremely discriminatory features from view specific features to enhance the functioning of the classifier. Additionally, the proposed design emphasizes the use of one individual view for testing as against the earlier models, where all views are mandatory as input. The advantage of DTED over prior works is to learn discriminations within view between class features and merging within class between view features. Subsequently, it will be noteworthy to test the many – to – one cross view evaluation, where the models are trained with view specific features and tested with only one view invariant feature.

4.3 Many – to – one classifier evaluation

At this juncture, we train the classifiers with all available views and test with just one view invariant feature. The mRA values for several sets of training views is presented in Table 2. The findings in Table 2 shows that an increase in the number of view specific training view features increases the test performance of the models. This assessment highlights the efficacy of the produced view invariant features by DTED model. The Inception – V4 has demonstrated to outshine other baseline classifiers employed for investigation owing to the point that it comprises numerous attention layers for picking maximally impacting vectors.

4.4 Performance of view invariant feature at multiple convolutional layers

The evidence from the previous works [24] suggests that the view invariant features in multiple layers across the CNN in Fig. 2 will affect the overall performance of the classifier. In this regard, we experimented with features from different convolutional layers from Fig. 2. Given the trained model of Fig. 2, we inferenced using all views from all classes and extracted features after the maximum pooling layers. For convenience, we name these

Table 1. One – to – one evaluation of the designated classifiers trained with the one view invariant feature and examined with specific view features using the performance measure mRA

Classifier	Views Datasets	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
Tiny VGG – 16	KLEF3DSL_2Dskeletal	0.74	0.76	0.78	0.76	0.81	0.75	0.76	0.70	0.69	0.70
	NTU RGB-D	0.72	0.78	0.78	0.76	0.82	0.77	0.74	0.75	0.74	0.77
	SBU Kinect Interaction	0.72	0.74	0.73	0.75	0.73	0.74	0.74	0.68	0.67	0.64
	KLYoga3D	0.77	0.77	0.81	0.80	0.84	0.80	0.79	0.74	0.75	0.77
	KL3D_MVaction	0.76	0.75	0.75	0.77	0.75	0.73	0.76	0.72	0.71	0.71
Inception - V4	KLEF3DSL_2Dskeletal	0.79	0.81	0.82	0.80	0.85	0.79	0.80	0.75	0.73	0.74
	NTU RGB-D	0.77	0.82	0.83	0.81	0.86	0.81	0.79	0.79	0.79	0.81
	SBU Kinect Interaction	0.77	0.78	0.78	0.79	0.77	0.78	0.78	0.72	0.72	0.69
	KLYoga3D	0.81	0.82	0.85	0.84	0.89	0.84	0.83	0.78	0.79	0.82
	KL3D_MVaction	0.81	0.79	0.79	0.82	0.80	0.78	0.80	0.76	0.75	0.75
GoogleNet	KLEF3DSL_2Dskeletal	0.69	0.71	0.73	0.71	0.76	0.70	0.71	0.65	0.64	0.65
	NTU RGB-D	0.67	0.73	0.74	0.72	0.77	0.72	0.70	0.70	0.69	0.72
	SBU Kinect Interaction	0.68	0.69	0.68	0.70	0.68	0.69	0.69	0.63	0.62	0.60
	KLYoga3D	0.72	0.72	0.76	0.75	0.79	0.75	0.74	0.69	0.70	0.72
	KL3D_MVaction	0.71	0.70	0.70	0.72	0.70	0.69	0.71	0.67	0.66	0.66
ResNet - 50	KLEF3DSL_2Dskeletal	0.70	0.75	0.76	0.74	0.80	0.74	0.72	0.72	0.72	0.75
	NTU RGB-D	0.70	0.72	0.71	0.72	0.71	0.71	0.72	0.65	0.65	0.62
	SBU Kinect Interaction	0.74	0.75	0.79	0.77	0.82	0.77	0.76	0.72	0.72	0.75
	KLYoga3D	0.74	0.73	0.72	0.75	0.73	0.71	0.74	0.70	0.69	0.69
	KL3D_MVaction	0.77	0.79	0.81	0.79	0.84	0.78	0.79	0.73	0.72	0.72

Classifiers	Views Datasets	V11	V12	V13	V14	V15	Average mRA
Tiny VGG – 16	KLEF3DSL_2Dskeletal	0.70	0.78	0.81	0.74	0.77	0.75
	NTU RGB-D	0.74	0.76	0.81	0.75	0.81	0.77
	SBU Kinect Interaction	0.66	0.71	0.76	0.75	0.73	0.72
	KLYoga3D	0.76	0.83	0.85	0.75	0.77	0.78
	KL3D_MVaction	0.70	0.77	0.79	0.75	0.76	0.75

Inception - V4	KLEF3DSL_2Dskeletal	0.74	0.82	0.85	0.78	0.81	0.79
	NTU RGB-D	0.78	0.80	0.85	0.79	0.85	0.81
	SBU Kinect Interaction	0.71	0.75	0.81	0.79	0.78	0.76
	KLYoga3D	0.81	0.88	0.89	0.79	0.81	0.83
	KL3D_MVaction	0.75	0.81	0.84	0.79	0.81	0.79
GoogleNet	KLEF3DSL_2Dskeletal	0.65	0.73	0.76	0.69	0.72	0.70
	NTU RGB-D	0.69	0.71	0.76	0.70	0.76	0.72
	SBU Kinect Interaction	0.62	0.66	0.71	0.70	0.69	0.67
	KLYoga3D	0.71	0.78	0.80	0.70	0.72	0.74
	KL3D_MVaction	0.65	0.72	0.74	0.70	0.72	0.70
ResNet - 50	KLEF3DSL_2Dskeletal	0.72	0.73	0.78	0.72	0.78	0.74
	NTU RGB-D	0.64	0.69	0.74	0.72	0.71	0.69
	SBU Kinect Interaction	0.74	0.81	0.82	0.72	0.74	0.76
	KLYoga3D	0.68	0.75	0.77	0.72	0.74	0.72
	KL3D_MVaction	0.73	0.81	0.84	0.77	0.79	0.78

Table 2. Many – to – One evaluation of the classifiers trained with several sets of training views and tested with one view invariant feature produced using DTED. The mixture of training views was introduced arbitrarily

Classifiers	Training Views	1	2	3	4	5	6	7	8	9	10
	Datasets										
Tiny VGG – 16	KLEF3DSL_2Dskeletal	0.62	0.63	0.64	0.64	0.69	0.70	0.70	0.72	0.74	0.76
	NTU RGB-D	0.61	0.63	0.65	0.68	0.70	0.72	0.73	0.75	0.77	0.80
	SBU Kinect Interaction	0.61	0.62	0.63	0.63	0.64	0.65	0.69	0.71	0.73	0.75
	KLYoga3D	0.61	0.62	0.64	0.66	0.68	0.69	0.70	0.72	0.74	0.76
	KL3D_MVaction	0.61	0.63	0.64	0.64	0.68	0.68	0.69	0.73	0.75	0.76
Inception - V4	KLEF3DSL_2Dskeletal	0.68	0.68	0.69	0.69	0.74	0.75	0.75	0.76	0.77	0.79
	NTU RGB-D	0.67	0.69	0.70	0.73	0.75	0.78	0.79	0.79	0.81	0.83
	SBU Kinect Interaction	0.66	0.68	0.69	0.69	0.70	0.71	0.74	0.75	0.76	0.78
	KLYoga3D	0.67	0.68	0.69	0.71	0.73	0.74	0.75	0.76	0.78	0.80
	KL3D_MVaction	0.67	0.69	0.69	0.70	0.73	0.74	0.75	0.76	0.79	0.80
GoogleNet	KLEF3DSL_2Dskeletal	0.65	0.65	0.66	0.66	0.71	0.73	0.73	0.73	0.75	0.77
	NTU RGB-D	0.64	0.66	0.67	0.70	0.72	0.75	0.76	0.76	0.78	0.81

	SBU Kinect Interaction	0.63	0.64	0.66	0.65	0.67	0.68	0.72	0.72	0.74	0.76
	KLYoga3D	0.63	0.65	0.66	0.68	0.70	0.72	0.73	0.73	0.75	0.77
	KL3D_MVaction	0.63	0.66	0.66	0.67	0.70	0.71	0.72	0.74	0.76	0.77
ResNet - 50	KLEF3DSL_2Dskeletal	0.63	0.63	0.64	0.64	0.69	0.70	0.70	0.73	0.74	0.76
	NTU RGB-D	0.62	0.64	0.65	0.68	0.70	0.73	0.74	0.76	0.78	0.80
	SBU Kinect Interaction	0.61	0.63	0.64	0.63	0.65	0.66	0.69	0.72	0.73	0.75
	KLYoga3D	0.61	0.63	0.64	0.66	0.68	0.69	0.70	0.73	0.75	0.77
	KL3D_MVaction	0.62	0.64	0.64	0.65	0.68	0.69	0.70	0.73	0.76	0.77

Classifiers	Training Views Datasets	11	12	13	14	15	Average mRA
Tiny VGG – 16	KLEF3DSL_2Dskeletal	0.78	0.79	0.84	0.86	0.90	0.73
	NTU RGB-D	0.82	0.84	0.87	0.88	0.92	0.76
	SBU Kinect Interaction	0.78	0.81	0.84	0.86	0.89	0.72
	KLYoga3D	0.80	0.81	0.82	0.85	0.90	0.73
	KL3D_MVaction	0.79	0.80	0.83	0.86	0.89	0.73
Inception - V4	KLEF3DSL_2Dskeletal	0.82	0.83	0.88	0.90	0.93	0.78
	NTU RGB-D	0.85	0.87	0.90	0.91	0.95	0.80
	SBU Kinect Interaction	0.82	0.84	0.88	0.90	0.93	0.77
	KLYoga3D	0.83	0.85	0.86	0.89	0.93	0.78
	KL3D_MVaction	0.82	0.83	0.86	0.89	0.92	0.77
GoogleNet	KLEF3DSL_2Dskeletal	0.79	0.81	0.86	0.88	0.92	0.75
	NTU RGB-D	0.83	0.86	0.88	0.90	0.94	0.78
	SBU Kinect Interaction	0.79	0.83	0.86	0.88	0.91	0.74
	KLYoga3D	0.81	0.83	0.84	0.87	0.92	0.75
	KL3D_MVaction	0.80	0.81	0.84	0.88	0.91	0.75
ResNet - 50	KLEF3DSL_2Dskeletal	0.79	0.80	0.85	0.87	0.90	0.74
	NTU RGB-D	0.82	0.84	0.87	0.88	0.92	0.76
	SBU Kinect Interaction	0.79	0.81	0.85	0.87	0.90	0.73
	KLYoga3D	0.80	0.82	0.83	0.86	0.90	0.74
	KL3D_MVaction	0.79	0.80	0.83	0.86	0.89	0.73

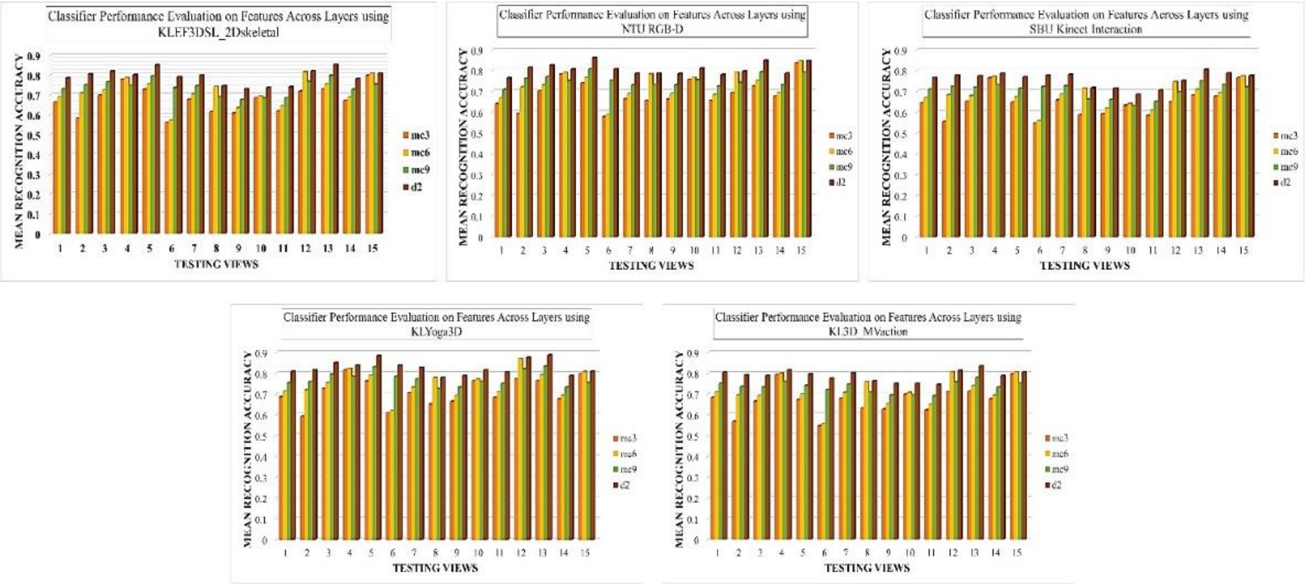

Figure. 7 One – to – one performance of view specific features extracted from different layers on view invariant trained Inception – V4 classifier for all skeletal video datasets

layers as mc3, mc6 and mc9. Markedly, the previous experiments considered the feature after the second dense layer, which will be called as d2. The features from mc3, mc6 and mc9 are multi-dimensional with, $126 \times 126 \times 32$, $61 \times 61 \times 64$ and $15 \times 15 \times 128$respectively. Eventually, the entire feature set is considered for training the DTED feature generators. The obtained view invariant features from these layers were used for classification. Here, we evaluate these intermediate layered features on Inception – V4 model classifier for its highest 10-fold mRA for the features in layer d2 on all datasets. The one – to – one evaluation is adopted where the inception – V4 is trained with view invariant features obtained from DTED and tested with view specific features in all views. The hyper parameters for training on each of these layered features are kept constant, except for the number of training epochs. Customarily, as the learning rate was kept constant at 0.0001 for all layered features, it becomes inevitable to train the network for more iterations. Therefore, the number of epochs were selected based on the input size and the maximum number of iterations was 650 for mc6 with maximum input size. The results of this experiment are shown in Fig. 7.

The results show that the mc3 features performed poorly over features from other layers. Because the features in mc3 have less overall information to characterize a particular object in the video frame when compared the deeper layers. Overall, we can observe that the d2 layer features have produced the highest recognition accuracies across all datasets. This is evidenced by the fact that the d2 features consists of comprehensive information about the structure of skeletal data in a video frame in comparison to features across other layers. Despite high performance of the proposed DTED in generating view invariant features, it would be interesting to find their usefulness against the previously proposed state – of – the – arts such as self-similarity matrix (SSM) [23] and sample affinity matrix (SAM) [24].

4.5 Evaluations compared to view invariant production techniques

The previous models SSM [23] and SAM [24] are designed with objective functions that require precise control on hyperparameters for maximizing the performance of the classifier. Though the proposed work has three independent deep learning models to be trained and inferenced for classification, it is free from those independently selectable hyperparameters. Table 3 presents the results of SSM and SAM along with our proposed DTED model on benchmark datasets.

The results in Table 3 were averaged across views for estimating the performance of these methods. Interestingly, the SSM has performed weakly against the SAM and DTED methods. The reason being that the SMM does not consider the relationships among views between classes to build view invariant features. Even though, SAM considers both within class between view and within view between class relationships, it is built on optimization platform with multiple regularization terms. Obviously, it has become difficult to predict the hyper parameters of regularizers, especially for skeletal sign language data. The concept of SAM was adopted in our work,

Table 3. Evaluation of the proposed view invariant feature generator DTED against two most efficient methods SSM, SAM and temporal self similarities

Multi View Algorithms	Classifiers	Tiny VGG – 16		Inception - V4		GoogleNet		ResNet – 50	
	Train Test Methods	One – to – one	Many – to – one	One – to – one	Many – to – one	One – to – one	Many – to – one	One – to – one	Many – to – one
	Datasets								
SSM [31]	KLEF3DSL _2Dskeletal	0.58	0.68	0.65	0.77	0.58	0.73	0.62	0.72
	NTU RGB-D	0.61	0.77	0.68	0.81	0.66	0.79	0.65	0.77
	SBU Kinect Interaction	0.56	0.71	0.63	0.73	0.60	0.70	0.61	0.70
	KLYoga3D	0.61	0.75	0.69	0.83	0.67	0.80	0.67	0.78
	KL3D_ MVaction	0.59	0.71	0.67	0.77	0.64	0.76	0.64	0.75
SAM [32]	KLEF3DSL _2Dskeletal	0.68	0.77	0.73	0.87	0.68	0.81	0.72	0.82
	NTU RGB-D	0.71	0.87	0.78	0.91	0.75	0.89	0.75	0.87
	SBU Kinect Interaction	0.66	0.81	0.73	0.82	0.70	0.80	0.70	0.80
	KLYoga3D	0.71	0.85	0.79	0.93	0.77	0.90	0.76	0.88
	KL3D_ MVaction	0.69	0.80	0.76	0.86	0.74	0.85	0.74	0.85
Temporal self-similarities [21]	KLEF3DSL _2Dskeletal	0.60	0.73	0.65	0.80	0.61	0.73	0.62	0.73
	NTU RGB-D	0.64	0.76	0.68	0.82	0.65	0.79	0.65	0.77
	SBU Kinect Interaction	0.58	0.71	0.63	0.73	0.61	0.71	0.60	0.70
	KLYoga3D	0.65	0.75	0.69	0.83	0.67	0.81	0.66	0.78
	KL3D_ MVaction	0.63	0.71	0.66	0.78	0.64	0.76	0.64	0.75
DTED Proposed	KLEF3DSL _2Dskeletal	0.72	0.85	0.77	0.92	0.73	0.85	0.74	0.85
	NTU RGB-D	0.76	0.88	0.79	0.94	0.77	0.91	0.77	0.89
	SBU Kinect Interaction	0.70	0.83	0.75	0.85	0.73	0.83	0.72	0.82
	KLYoga3D	0.77	0.87	0.81	0.95	0.79	0.93	0.78	0.90
	KL3D_ MVaction	0.75	0.82	0.78	0.90	0.76	0.87	0.76	0.87

except for the fact that the view invariant features were learned using triplet loss embedding between within class views and across class views by eliminating the dependencies. More importantly, the generated view invariant features have shown highest recognition accuracies on complicated skeletal video datasets. Further, we evaluated our spatio temporal representation of features against temporal self-similarities method [21] for view invariance. However, the self-similarities in temporal domain have lost the overall appearance information in the skeletal frames resulting in weakly modelled features.

4.6 Validation of DTED against state – of – the – arts

Chronological validation of the DTED is presented by judging it with state – of – the – art view invariant methods. The techniques preferred for evaluation have employed various

Table 4. Comparison between several view-based identification techniques

		V1	V2	V3	V4	V5	V6	V7	V8	V9	V10
NTU RGB+D	[16]	0.65	0.63	0.62	0.65	0.69	0.70	0.62	0.59	0.64	0.61
	[17]	0.66	0.64	0.63	0.66	0.70	0.71	0.63	0.60	0.65	0.62
	[18]	0.63	0.61	0.60	0.63	0.67	0.68	0.60	0.57	0.62	0.59
	[19]	0.66	0.64	0.62	0.66	0.69	0.71	0.62	0.60	0.65	0.62
	[21]	0.63	0.61	0.60	0.63	0.67	0.68	0.60	0.57	0.62	0.60
	[26]	0.67	0.65	0.63	0.67	0.70	0.72	0.63	0.61	0.66	0.63
	[33]	0.64	0.69	0.70	0.68	0.74	0.69	0.66	0.66	0.66	0.69
	ours	0.77	0.82	0.83	0.81	0.86	0.81	0.79	0.79	0.79	0.81
SBU Kinect Interaction	[16]	0.67	0.65	0.64	0.67	0.71	0.72	0.64	0.61	0.67	0.64
	[17]	0.67	0.65	0.64	0.67	0.71	0.72	0.64	0.61	0.66	0.63
	[18]	0.68	0.66	0.65	0.68	0.72	0.73	0.65	0.62	0.67	0.64
	[19]	0.65	0.63	0.62	0.65	0.69	0.70	0.62	0.59	0.64	0.61
	[21]	0.68	0.66	0.64	0.68	0.71	0.73	0.64	0.62	0.67	0.64
	[26]	0.65	0.63	0.62	0.65	0.69	0.70	0.62	0.59	0.64	0.62
	[33]	0.66	0.71	0.72	0.70	0.76	0.70	0.68	0.68	0.68	0.70
	Ours	0.77	0.78	0.78	0.79	0.77	0.78	0.78	0.72	0.72	0.69
KLYoga3D	[16]	0.69	0.67	0.65	0.69	0.72	0.74	0.65	0.63	0.68	0.65
	[17]	0.69	0.67	0.66	0.69	0.73	0.74	0.66	0.63	0.69	0.66
	[18]	0.68	0.66	0.65	0.68	0.72	0.73	0.65	0.62	0.67	0.64
	[19]	0.69	0.67	0.66	0.69	0.73	0.74	0.66	0.63	0.68	0.65
	[21]	0.66	0.64	0.63	0.66	0.70	0.71	0.63	0.60	0.65	0.62
	[26]	0.69	0.67	0.65	0.69	0.72	0.74	0.65	0.63	0.68	0.65
	[33]	0.68	0.73	0.74	0.72	0.78	0.72	0.70	0.70	0.70	0.72
	Ours	0.81	0.82	0.85	0.84	0.89	0.84	0.83	0.78	0.79	0.82
KL3D_MVaction	[16]	0.66	0.64	0.63	0.66	0.70	0.71	0.63	0.60	0.65	0.63
	[17]	0.70	0.68	0.66	0.70	0.73	0.75	0.66	0.64	0.69	0.66
	[18]	0.70	0.68	0.67	0.70	0.74	0.75	0.67	0.64	0.70	0.67
	[19]	0.64	0.62	0.61	0.64	0.68	0.69	0.61	0.58	0.63	0.60
	[21]	0.65	0.63	0.62	0.65	0.69	0.70	0.62	0.59	0.64	0.61
	[26]	0.62	0.60	0.59	0.62	0.66	0.67	0.59	0.56	0.61	0.58
	[33]	0.67	0.72	0.73	0.71	0.76	0.71	0.69	0.69	0.69	0.71
	Ours	0.81	0.79	0.79	0.82	0.80	0.78	0.80	0.76	0.75	0.75
KLEF3DSL_2Dskeletal	[16]	0.65	0.63	0.61	0.65	0.68	0.70	0.61	0.59	0.64	0.61
	[17]	0.62	0.60	0.59	0.62	0.66	0.67	0.59	0.56	0.61	0.59
	[18]	0.66	0.64	0.62	0.66	0.69	0.71	0.62	0.60	0.65	0.62
	[19]	0.66	0.64	0.63	0.66	0.70	0.71	0.63	0.60	0.66	0.63
	[21]	0.63	0.61	0.59	0.63	0.66	0.68	0.59	0.57	0.62	0.59
	[26]	0.60	0.58	0.56	0.60	0.64	0.65	0.56	0.54	0.59	0.56
	[33]	0.66	0.71	0.72	0.70	0.75	0.70	0.68	0.68	0.68	0.70
	Ours	0.79	0.81	0.82	0.80	0.85	0.79	0.80	0.75	0.73	0.74

		V11	V12	V13	V14	V15	Average mRA
NTU RGB+D	[16]	0.62	0.61	0.62	0.65	0.61	0.63
	[17]	0.63	0.62	0.61	0.65	0.68	0.65
	[18]	0.60	0.59	0.58	0.62	0.65	0.62
	[19]	0.63	0.62	0.62	0.65	0.69	0.65
	[21]	0.60	0.59	0.62	0.66	0.69	0.63
	[26]	0.64	0.63	0.61	0.65	0.68	0.65
	[33]	0.66	0.67	0.73	0.67	0.72	0.68
	ours	0.78	0.80	0.85	0.79	0.85	0.81
SBU Kinect Interaction	[16]	0.64	0.63	0.62	0.66	0.69	0.66
	[17]	0.64	0.63	0.59	0.63	0.66	0.65
	[18]	0.65	0.64	0.62	0.66	0.69	0.66
	[19]	0.62	0.61	0.59	0.63	0.66	0.63
	[21]	0.65	0.64	0.63	0.66	0.70	0.66
	[26]	0.62	0.61	0.63	0.67	0.70	0.64
	[33]	0.67	0.69	0.74	0.68	0.74	0.70
	Ours	0.71	0.75	0.81	0.79	0.78	0.76
KLYoga3D	[16]	0.66	0.65	0.57	0.61	0.64	0.66
	[17]	0.66	0.65	0.58	0.62	0.65	0.67
	[18]	0.65	0.64	0.55	0.59	0.62	0.65
	[19]	0.66	0.65	0.58	0.62	0.65	0.66
	[21]	0.63	0.62	0.55	0.59	0.62	0.63
	[26]	0.66	0.65	0.59	0.62	0.66	0.66
	[33]	0.69	0.71	0.76	0.70	0.76	0.72
	Ours	0.81	0.88	0.89	0.79	0.81	0.83
KL3D_MVaction	[16]	0.63	0.62	0.59	0.63	0.66	0.64
	[17]	0.67	0.66	0.63	0.66	0.70	0.68
	[18]	0.67	0.66	0.63	0.67	0.70	0.68
	[19]	0.61	0.60	0.57	0.61	0.64	0.62
	[21]	0.62	0.61	0.58	0.62	0.65	0.63
	[26]	0.59	0.58	0.55	0.59	0.62	0.60
	[33]	0.68	0.70	0.75	0.69	0.75	0.71
	Ours	0.75	0.81	0.84	0.79	0.81	0.79
KLEF3DSL_2Dskeletal	[16]	0.62	0.61	0.58	0.62	0.65	0.63
	[17]	0.59	0.58	0.55	0.59	0.62	0.60
	[18]	0.63	0.62	0.59	0.62	0.66	0.64
	[19]	0.63	0.62	0.59	0.63	0.66	0.64
	[21]	0.60	0.58	0.56	0.59	0.63	0.61
	[26]	0.57	0.56	0.53	0.57	0.60	0.58
	[33]	0.67	0.69	0.74	0.68	0.74	0.70
	Ours	0.74	0.82	0.85	0.78	0.81	0.79

kinds of learning algorithms for production and categorization of video views. Because the information employed in these techniques were distinct, we reconstructed these models from scratch as given in their corresponding scripts. All the experimentations were performed on the benchmark skeletal data utilized in this paper with one – to – one train – test design. We represented our finest outcome gained from inception V4 classifier in this comparison. Nevertheless, the hyper parameters for this comparison networks were implemented from our Inception V4.

Based on the results in Table 4, the success of our DTED blended view feature generator when compared to previous models is threefold. One, the selection of feature representation as spatio temporal matrix as against either spatial or temporal in previous works. Two, the computation VCM which calculates the shared features across views and classes. Finally, the application of deep Triplet encoder decoder network to generate highly Discrimant view invariant features.

5. Conclusions

Exalting multi view skeletal sign language recognition to leverage the effects of view sensitivity during classification has been achieved. Accordingly, the VCM provided weighted relationships between shared features across views and class from automated view specific spatio temporal features. These shared features are grouped with view specific ones using triple loss embedding on deep triplet encoder decoder learning model. Eventually, a highly discriminative view invariant features were generated which can be applied independently for classification as against other previous models where these are mixed with view specific features. Undoubtedly, the classifiers performance has registered an upward improvement over similar methods.

References

[1] R. Rastgoo, K. Kiani, and S. Escalera, "Sign language recognition: A deep survey", *Expert Systems with Applications*, Vol. 164, p. 113794, 2021.

[2] R. Poppe, "A survey on vision-based human action recognition", *Image and vision computing*, Vol. 28, No. 6, pp. 976-990, 2010.

[3] R. Rastgoo, K. Kiani, and S. Escalera, "Multi-Modal Deep Hand Sign Language Recognition in Still Images Using Restricted Boltzmann Machine", *Entropy*, Vol. 20, No. 11, p. 809, 2018.

[4] R. Elakkiya, "Machine learning based sign language recognition: A review and its research frontier", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 12, No. 7, pp. 7205-7224, 2021.

[5] O. M. Sincan and H. Y. Keles, "Autsl: A large scale multi-modal turkish sign language dataset and baseline methods", *IEEE Access*, Vol. 8, pp. 181340-181355, 2020.

[6] P. V. V. Kishore, D. A. Kumar, A. S. C. S. Sastry, and E. K. Kumar, "Motionlets matching with adaptive kernels for 3-d indian sign language recognition", *IEEE Sensors Journal*, Vol. 18, No. 8, pp. 3327-3337, 2018.

[7] A. Shahroudy, J. Liu, T. T. Ng, and G. Wang, "Ntu rgb+ d: A large scale dataset for 3d human activity analysis", In: *Proc. of the IEEE conference on computer vision and pattern recognition*, pp. 1010-1019. 2016.

[8] M. Li and H. Leung, "Multiview skeletal interaction recognition using active joint interaction graph", *IEEE Transactions on Multimedia*, Vol. 18, No. 11 pp. 2293-2302, 2016.

[9] T. K. K. Maddala, P. V. V. Kishore, K. K. Eepuri, and A. K. Dande, "YogaNet: 3-D Yoga Asana Recognition Using Joint Angular Displacement Maps With ConvNets", *IEEE Transactions on Multimedia*, Vol. 21, No. 10, pp. 2492-2503, 2019.

[10] D. Srihari, P. V. V. Kishore, K. K. Eepuri, A. K. Dande, T. K. K. Maddala, M. V. D. Prasad, and Ch. R. Prasad, "A four-stream ConvNet based on spatial and depth flow for human action classification using RGB-D data", *Multimedia Tools and Applications*, Vol. 79, No. 17, pp. 11723–11746, 2020.

[11] N. Aloysius and M. Geetha, "Understanding vision-based continuous sign language recognition", *Multimedia Tools and Applications*, Vol. 79, No. 31, pp. 22177-22209, 2020.

[12] Y. Xing and J. Zhu, "Deep learning‐based action recognition with 3D skeleton: A survey", *CAAI Transactions on Intelligence Technology*, Vol. 6, No. 1, pp. 80‑92, 2021.

[13] T. Hussain, K. Muhammad, W. Ding, J. Lloret, S. W. Baik, and V. H. C. D. Albuquerque, "A comprehensive survey of multi-view video summarization", *Pattern Recognition*, Vol. 109, pp. 107567, 2021.

[14] Gao, Zan, H. Zhang, G. P. Xu, Y. B. Xue, and A. G. Hauptmann, "Multi-view discriminative and structured dictionary learning with group

sparsity for human action recognition", *Signal Processing*, Vol. 112, pp. 83-97, 2015.

[15] Zheng, Jingjing, Z. Jiang, and R. Chellappa. "Cross-view action recognition via transferable dictionary learning", *IEEE Transactions on Image Processing*, Vol. 25, No. 6, pp. 2542-2556, 2016.

[16] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive recurrent neural networks for high-performance human action recognition from skeleton data", In: *Proc. of the IEEE International Conference on Computer Vision*, pp. 2117-2126, 2017.

[17] A. Ullah, K. Muhammad, T. Hussain, and S. W. Baik, "Conflux LSTMs network: A novel approach for multi-view action recognition", *Neurocomputing*, Vol. 435, pp. 321-329, 2021.

[18] C. Si, W. Chen, W. Wang, L. Wang, and T. Tan, "An attention enhanced graph convolutional lstm network for skeleton-based action recognition", In: *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1227-1236. 2019.

[19] D. Wang, W. Ouyang, W. Li, and D. Xu, "Dividing and aggregating network for multi-view action recognition", In: *Proc. of the European Conference on Computer Vision (ECCV)*, pp. 451-467, 2018.

[20] P. V. V. Kishore, M. V. D. Prasad, C. R. Prasad, and R. Rahul, "4-Camera model for sign language recognition using elliptical fourier descriptors and ANN", In: *Proc. of International Conference on Signal Processing and Communication Engineering Systems*, IEEE, pp. 34-38, 2015.

[21] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez, "Cross-view action recognition from temporal self-similarities", In: *Proc. of European Conference on Computer Vision*, Springer, Berlin, Heidelberg, pp. 293-306, 2008.

[22] H. Rahmani and A. Mian, "Learning a non-linear knowledge transfer model for cross-view action recognition", In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2458-2466, 2015.

[23] J. Zheng, Z. Jiang, and R. Chellappa, "Cross-view action recognition via transferable dictionary learning", *IEEE Transactions on Image Processing*, Vol. 25, No. 6, pp. 2542-2556, 2016.

[24] H. Rahmani, A. Mahmood, D. Huynh, and A. Mian, "Histogram of oriented principal components for cross-view action recognition", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 38, No. 12, pp. 2430-2443, 2016.

[25] J. Wang, X. Nie, Y. Xia, Y. Wu, and S. C. Zhu, "Cross-view action modeling, learning and recognition", In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2649-2656, 2014.

[26] W. Nie, A. Liu, W. Li, and Y. Su, "Cross-view action recognition by cross-domain learning", *Image and Vision Computing*, Vol. 55, pp. 109-118, 2016.

[27] N. E. D. Elmadany, Y. He, and L. Guan, "Information fusion for human action recognition via biset/multiset globality locality preserving canonical correlation analysis", *IEEE Transactions on Image Processing*, Vol. 27, No. 11, pp. 5275-5287, 2018.

[28] L. Shao, L. Liu, and M. Yu, "Kernelized multiview projection for robust action recognition", *International Journal of Computer Vision*, Vol. 118, No. 2, pp. 115-129, 2016.

[29] L. Wang, Z. Ding, Z. Tao, Y. Liu, and Y. Fu, "Generative multi-view human action recognition", In: *Proc. of the IEEE/CVF International Conference on Computer Vision*, pp. 6212-6221, 2019.

[30] Y. Wang, L. Wu, X. Lin, and J. Gao, "Multiview spectral clustering via structured low-rank matrix factorization", *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 29, No. 10, pp. 4833-4843, 2018.

[31] C. Sun, I. N. Junejo, M. Tappen, and H. Foroosh, "Exploring sparseness and self-similarity for action recognition", *IEEE Transactions on Image Processing*, Vol. 24, No. 8, pp. 2488-2501, 2015.

[32] Y. Kong, Z. Ding, J. Li, and Y. Fu, "Deeply learned view-invariant features for cross-view action recognition", *IEEE Transactions on Image Processing*, Vol. 26, No. 6, pp. 3028-3037, 2017.

[33] Y. Chen, Z. Zhang, C. Yuan, B. Li, Y. Deng, and W. Hu, "Channel-wise topology refinement graph convolution for skeleton-based action recognition", In: *Proc. of the IEEE/CVF International Conference on Computer Vision*, pp. 13359-13368, 2021.