Deep Multi View Spatio Temporal Spectral Feature Embedding on Skeletal Sign Language Videos for Recognition

SK. Ashraf Ali[1], M. V. D. Prasad[2], P.Praveen Kumar[3], P. V. V. Kishore[4]
Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, India.[1,2,4]
Department of Information Technology, Vignan's Institute of Information Technology, Duvvada, Visakhapatnam 530049, India.[3]

***Abstract*—To build a competitive global view from multiple views which will represent all the views within a class label is the primary objective of this work. The first phase involves the extraction of spatio temporal features from videos of skeletal sign language using a 3D convolutional neural network. In phase two, the extracted spatio temporal features are ensembled into a latent low dimensional subspace for embedding in the global view. This is achieved by learning the weights of the linear combination of Laplacian eigenmaps of multiple views. Subsequently, the constructed global view is applied as training data for sign language recognition.**

***Keywords*—Laplacian eigenmaps; 3D convolutional networks; sign language recognition; multi view; skeletal data**

I. Introduction

Sign Language Recognition (SLR) is extremely coordinated movements of hands captured through sensors as 1/2/3D data and translated into text or voice by a machine learning interface [1]. Sign language is a communication medium for hearing impaired people which consists of hand movements and finger shapes that operate independently or collaboratively with respect to upper body parts. SLR is considered an extension of human action recognition (HAR) [2]. Automated HAR or SLR is accomplished through machine learning approaches on multi modal datasets such as RGB, Depth and skeletal information in image, video and data formats. The RGB and depth formats provide appearance information whereas the skeletal joint data exclusively models pose details. Although SL knowledge representation is largely modelled in RGB video formats, it is bottlenecked by motion blurring and spatial resolution of fingers with respect to the frame size. Therefore, the skeletal data has obtained wide acceptance for human action or sign language recognition problems. The 3D skeletal data has been used as vectorized, image and RGB video formats for recognition.

However, the pattern identification process on skeletal 3D video data for building a real time application is a supremely challenging task. Traditional models employed vectorized 3D data for recognition with deep neural networks(DNN) [3]. Above all the DNN models on 3D skeletal action data, long short-term memory (LSTM) [4] networks have shown greater reliability and robustness for HAR tasks. Similarly, 3D skeletal SLR on vectorized data was successfully designed and experimented with color coded Spatio-Temporal features [5]. Singularly, most of these methods presented results related to cross view testing with poor performance as these models received only single view training. As a result, the above methods failed to generalize on building a real time engine for HAR or SLR.

Meanwhile, the above problem is finding solutions in the form of multi view training on Deep Learning Models. Though multi view processing of video data is having 2 decades of research history, it has gained extensive attention in the last few years due to the progress in deep learning approaches. Earlier DNN proposed were constructed with multiple streams feeding into individual views independently whose Softmax scores are fused for getting a final recognition score. Later, learning approaches have trained multiple CNNs for each view and then learned the concatenated features in the dense layers. This approach has allowed for multiple views to share features across classes. Specifically, this process does not restrict the features that were not significant in the decision making. Additionally, the view specific features that play a major role in articulating the desired outcome are ignored.

To overcome the above challenges, we propose to learn a global synthesized target view by linearly combining the independent multiple views as suggested in [6]. However, these intra class independent views have shown to exhibit unequal similarities with other views which biases the result towards the false positives. Hence, to overcome this uniformity across views that influence the target class, we propose higher order Laplacian eigenmaps from [7]. This enables the target feature reconstruction to have a complete non uniform distribution across the multiple independent views. Consequently, we learn a nonuniform linear combination of weights on independent views which can be generalized for any target view. Finally, the synthesized target view features of all classes are classified using standard deep learning architectures. The proposed methodology called multi view spatio temporal feature embedding (MVSTFE) is illustrated in the following Fig. 1.

The proposed MVSTFE is investigated on our 3D skeletal video datasets of sign language (KLEF3DSL_2Dskeletal) [8] and four other multiview action datasets NTU RGB-D [9], SBU Kinect Interaction [10], KLYoga3D [11] and KL3D_MVaction [12]. The performance of the proposed deep networks was tested for the proposed method against the state-of-the-art on datasets. The remaining paper is clustered into four sections. The second section highlights the key historical aspects associated with multi view learning, sign language

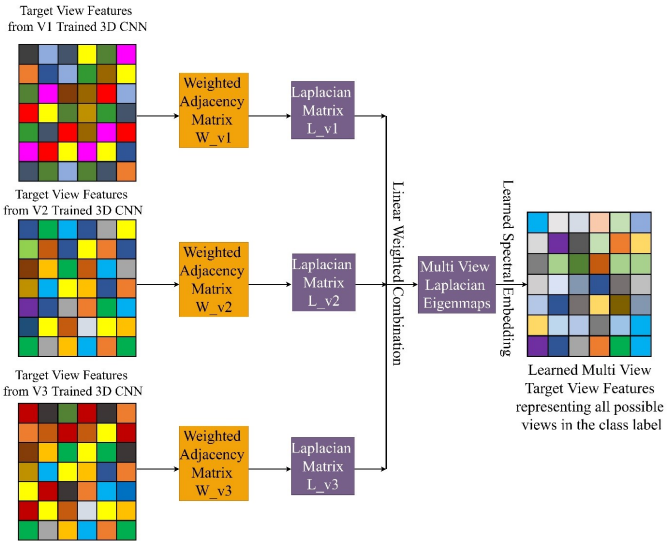

Fig. 1. Illustration of the Proposed Multi View Spatio Temporal Feature Embedding on Skeletal Sign Language Video Data.

recognition and deep networks. The methodology is packaged in the third section and the obtained results for experimentation with analysis are presented in section four. Finally, conclusions were drawn from the analytical insights gained on the overall performance of the proposed models.

II. Literature Review

This section of the paper dwells on the advantages and disadvantages of the previous methods of sign language and action recognition in multiple views. Additionally, it also discusses the current models in deep metric learning.

With the advent of deep learning frameworks, the 2D video based SLR has become powerful with the option of feature learning rather than feature extraction. A large contingent of them is available for perusal [13]. The accuracies reported by these methods are not reproducible or they simply fail to generalize on the video quality or the signer. This has motivated researchers towards higher dimensional data such as RGB D or 3D skeletal representations. Multi modal video sequences that are fed into multiple streams of a CNN are predominantly researched which have shown evidence of exceptional performances in real time for sign (action) recognition applications [14]. The recognition accuracies were better than the single modal datasets. However, the training requires higher computing powers, and the datasets are captured with special devices making it an unfeasible deployable solution.

Eventually, to develop a real time SLR or HAR system, it is intuitive to learn multi views across datasets. This has initiated action recognition research to move in the direction of developing view-based learning algorithms [15], [16]. Multi view HAR has evolved through research using dictionary learning [17], neural networks with adaptable views [18], convolutional neural networks [19] and deep attention models [20], to name a few. However, the most widely researched and acknowledged models are from deep learning networks. Moreover, visual attention models with deep CNNs have established themselves as a formidable solution to multi view learning [21]. Despite their success, attention models are specific to a particular view and the view specific features are to be fused accordingly for classification by the dense layers. The fusion mechanisms ensemble the view specific features into a multi view feature vector that has failed to capture the variations in multi view data [22].

Primarily multiview approaches were classified as multi-view learning and view invariant models. In multiview learning, the video input is considered as a time series of data frames in different views which are learned independently by the classifier [23], [24]. Most of the methods used low level observable features for generating discriminative features [17]. Subsequently, multiple training methods were employed for each of the views to find a set of consistent features between a pair of views [25], [26]. The algorithms are used for finding relationships between views canonical correlation analysis(CCA) [27] and projection matrices [28]. Extending to the above methods are matrix factorization [29] and low rank constrained matrix factorization [30] for capturing view similarities. All these models have shown good performance on instances where the number of views were limited and require extensive computational power for deployment.

Alternatively, view invariant models developed linear descriptors to transfer information between views. Accordingly, these models consider target views as a linear combination of views within a class label [6], [7]. Subsequently, the weight vectors are computed by applying optimization in Laplacian space. Moreover, these works assume that all views contribute equally to the target view features. However, in sign language recognition with video data from multiple source views it is difficult to impose the above assumption in real time. To overcome the disadvantage of equal contribution by all views to the target view, we propose to learn these contributions in the Laplacian space using deep learning.

The following points make the proposed method unique from the existing ones:

1) To design an unequal linear view combiner to extract target view features.
2) To construct highly discriminative Spatio-Temporal features in the Laplacian space.
3) To reconstruct learned target vectors into a Spatio-Temporal feature representation with 3D CNNs.

In order to find an appropriate solution for multiview problems, the following objectives are being formulated:

1) To design an unequally contributing linear view combiner to identify the linear combinations.
2) To learn the mapping function for generating a singularly trainable view invariant Spatio-Temporal feature.
3) To initiate anyone view testing model. We call our proposed model multi view spatio temporal feature embedding (MVSTFE).

III. Multi View Spatio Temporal Feature Embedding (MVSTFE)

This section describes the proposed multi view spatio temporal feature embedding model for multi view sign language

recognition on skeletal video datasets. First, a cluster of 3D CNNs is trained independently on individual views for all classes in the dataset. Secondly, a target view is selected randomly which is referenced on the pre trained 3D CNNs for feature extraction. The extracted features from independent view streams are learned by compiling Laplacian eigenmaps to construct a combined target view. This combined target view features will represent a linear combination of Laplacian eigen maps from multiple views generating a highly discriminative feature for all views of the target view class. Finally, these learned target view features will be used for training any deep classifier for sign or action recognition.

A. Independent View 3D CNN Model

The primary step in the process of multi view sign language recognition is to design and train a 3D convolutional neural network (3D CNN). The 3D CNN takes input as the skeletal video sequences as input for supervised training. The number of 3D CNN streams are equal to the number of source views available for training. The 3D CNN architecture used in this work is shown in Fig. 2. The model has 4 pairs of 3D convolutional layers with one set of batch normalization and maximum pooling layers after each pair respectively. The input of the network is a 2D skeletal video sequence of size $256 \times 256 \times 3$ with 100 frames. The features at the end of the convolutional layers are flattened and inputted to two fully connected layers with the last layers being Softmax.

Let $X^{vc} = (x_v = \{S_v\}\, \forall v = 1\ to\ V, c\ =\ 1\ to\ C)$ be the RGB skeletal video sequences in V views with $V \in R^3$. The 3D CNN model will extract the features f^v from x_v with view specific labels y_v using the trainable parameters θ_{3D} by optimizing a loss function L on the overall multi view dataset as

$$\theta_{3D} = \arg\min_{\theta_{3D}} L\left(\theta_{3D}; x_v, y_v\right) \tag{1}$$

For classification tasks, we need a global loss function to

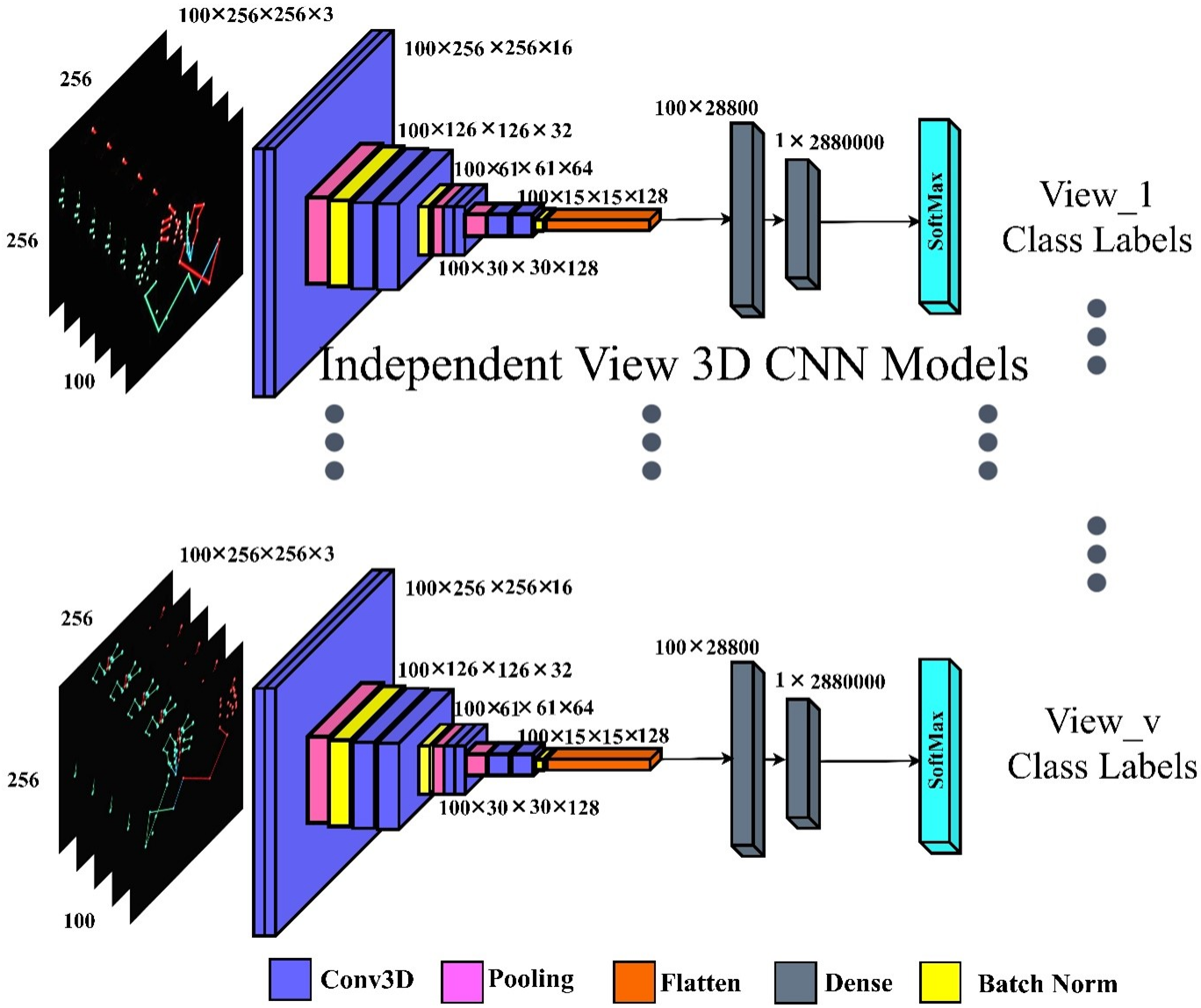

Fig. 2. 3D CNN Architecture for Training Multiple Views Independently across All Classes.

discriminate the classes with the help of SoftMax layers. The class label prediction is computed on the embedding space using the cross-entropy loss functional defined as

$$l_{CrossEnt} = -\sum_{i=1}^{C} \left(y_i \log\left(\widehat{y_i}\right) + (1 - y_i) \log\left(1 - \widehat{y_i}\right)\right) \quad (2)$$

The $l_{CrossEnt}$ is the loss function for training the network. The $(\widehat{y_i})$ is the predicted label and y_i is the actual. The C defines the total number of classes in the dataset. Each stream in the network is view independent with the specifications as shown in Fig. 2. Consequently, weight and biases are initialized using unit variance zero mean Gaussian random variable. The filter sizes in all 3D CNN layers is fixed at $3 \times 3 \times 3$. Moreover, the learning rate is dynamically controlled with 10% decrease rate from the previous valued whenever the loss became constant across 10 epochs. The initial learning rate was selected as 0.0001. Stochastic gradient descent optimizer is applied to update the wights and biases in the network. This trained network will be used to extract spatio temporal features from a target view which are further used to construct a combined view features. These constructed view features have the ability to represent all the views within a class label.

B. Combined View Feature Generation

Given a sign class in a specific target view x_{vt} as input the trained model θ_{3D}, the output features f_v at the end of dense layers are represented as

$$\{f_v\}_{v=\{1,V\}} = \sum_{i=1}^{I} \sum_{j=1}^{J} x_v\,(i,j) * K\,(k-i, k-j)\,\forall k \in \text{kernel size} \quad (3)$$

The features extraction network is shown in Fig. 3. The network consists of four pairs of convolutional layers with rectified linear activations followed by a 2×2 window maximum pooling layer. The strides of the kernels in convolution layers is one and that of maximum pooling is two. After maximum pooling a batch normalization layer is added to standardize the inputs to the deeper layers. Finally, two fully connected layers are added to learn on the feature extracted in the convolutional layers. Subsequently, the spatial features at the output of dense layers are concatenated along the frames to generate a complete spatio temporal feature matrix representing the 2D skeletal video sequence. Altogether, V streams operate independently in the network generating view specific class features $F^{cv} = \{f_{ic}\}\ \forall i = 1$ to $V \in R^{g \times N}$, Where g is the dimensionality of the features and N is the number of frames. The model is trained with categorical cross entropy loss with stochastic gradient descent optimizer on the entire dataset. The trained model $\widetilde{\theta_{3D}}$ is applied on all the input video frames to extract the feature samples as

$$F^{cv} = \widetilde{\theta_{3D}}\,(w, b) \times x_{vt}^{c} \forall V \ \ \& \ \ C \in R^{g \times N} \quad (4)$$

The spatio temporal feature matrix F^{cv} consists of the target view features inferenced from independently trained views across all classes. The objective is to generate a feature matrix that will represent all views in a class as a linear combination of the extracted features. Traditionally, this is achieved by considering the all the mixing coefficients are equally distributed across all views. However, equally distribution of information across all views has produced ambiguous recognition accuracies. To overcome this, non-uniform distribution is proposed [7] with Laplacian eigenmaps. In this work, we incorporate the process of spectral embedding using Laplacian eigenmaps to calculate the mixing coefficients of the linear combination.

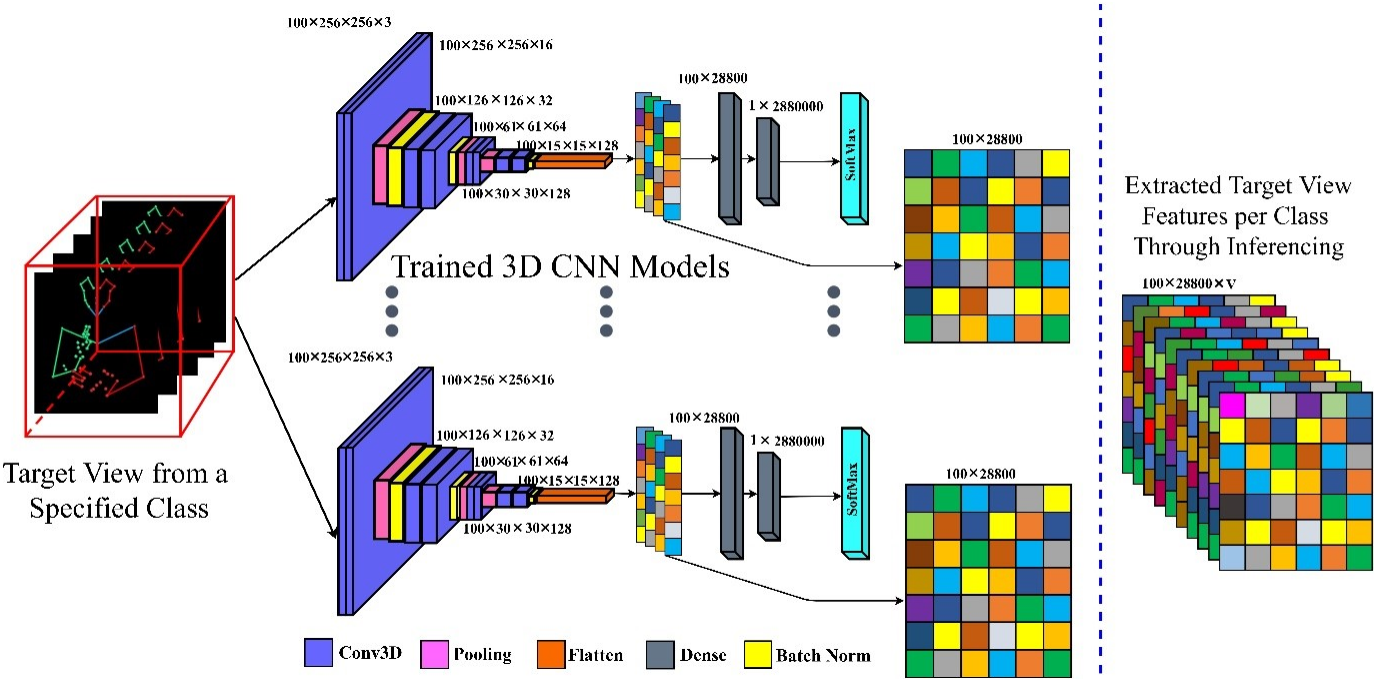

Fig. 3. The Inferencing Process on Trained 3D CNN Model with Features Extracted from Multiple Layers in the Network.

C. Constructing the Non-uniform View Linear Combiner

Given a set of spatio temporal target view features $F^{cv} \in R^{d \times V}$ from a particular class label $y_{i \in C}$ with V views, these views can be linearly combined with coefficients as

$$F^{cv}_{Comb} = \sum_{i=1}^{V} \lambda^i F^{cv(i)} \tag{5}$$

Where, V is the total number of source views and d is the feature matrix dimensionality. F^{cv}_{Comb} is the combined feature representation of the target feature. The mixing coefficients $\lambda^i \; \forall \; i = 1 \; to \; V$ is the weighted combination. The constraint on the mixing coefficient is

$$\sum_{i=1}^{V} \lambda^i = 1, \;\; \lambda^i > 0 \tag{6}$$

The intent in the above representation is to generate a global view that is compatible with all the views in the class. Mostly, the coefficient λ^i is considered as the average $1/V$ across all the views. However, in reality, the views that are in close proximity with the target view contribute more than $^1/_V$. Consequently, the obtained linearly combined global view features are least compatible for representing all the views in a class. This problem is solved by evaluating the mixing coefficients of individual views with the help of cost function derived using Laplacian eigenmaps [7].

First, the target features are arranged a V data matrices $F = \left\{F^{cv} \in R^d\right\}_{v=1}^{V} \;\; \forall d = R^{g \times N \times V}$ as shown in the output of Fig. 3. The objective is to calculate a set of mixing coefficients $\lambda = \left\{\lambda^i\right\}_{i=1}^{V}$. We start by initializing $\lambda = \left[\frac{1}{V}, ..., \frac{1}{V}\right]$. Subsequently, set the $g \times N$ feature points obtained from trained network in the t^{th} target view.

To compute the combined target view embedding features, we subsequently compute the weighted adjacency matrix A^t on the target features and the Laplacian matrix L^i of the individual views with $i \in (1, 2, ..., V)$. Consequently, the global Laplacian L^G of the entire target view class is computed as a linear combination of initial weights. The spectral encoding, Y^G can be computed from eigen value decomposition of L^G as a Laplacian eigen map. Accordingly, select the smallest eigen values other than the zeroth one, reconstruct the spectral encoding Y^{G*}. Using the reconstructed spectral encoding Y^{G*} and Y^G, update the mixing coefficients of the linear combination λ^i. Optimize till the distance between the reconstructed and the original spectral encoding are less than a set experimental threshold.

D. Construction of Laplacian Eigenmaps and Spectral Embedding

Given the feature data points in the t^{th} target view $\left\{F^{cv} \in R^d\right\}_{v=1}^{V}$ with $g \times N$ data points, we first compute the adjacency matrix A^t as

$$\left[A^t\right]_{i,j} = e^{\left(-\frac{\left\|F^{ci} - F^{cj}\right\|_2^2}{\sigma^2}\right)} \;\; \forall \; F^{ci} \;\; \& \;\; F^{cj} \; are \; associated \tag{7}$$

Where, A^t is a symmetric matrix of size $gN \times gN$. The value of σ is selected as 2. The adjacency matrix establishes a link between the target features extracted from trained CNN in Fig. 2 in all views. If the distance between the features is small, the value in the $(i,j)^{th}$ position tends towards 1 and vice versa. Consequently, A^testablishes a relationship between the features points formed by a set of d data points in multiple views.

Subsequently, to compute a single view feature combination from multiple target view features Laplacian eigenmaps were used from [30]. Laplacian eigenmaps reduces the data by projecting data on a different spectral view without compromising on the relationships between the feature points. Accordingly, the spectral encoding Y^{G*} can be computed by minimizing the cost function defined as

$$f\left(Y^G\right) = \sum_{i,j \in \{\forall gN\}} \left\|y_i^G - y_j^G\right\|^2 \left[A^t\right]_{i,j} \tag{8}$$

The above representation gives the difference between two embedding features in multiple views modulated by their association values in adjacency matrix. If the feature points in multiple views are in close proximity, the adjacency matrix value is large, thus contributing more to cost function. As a result of this, similar data points are preserved in the spectral embedding from different views. Eventually, the solution to the optimization is transformed into a minimization problem as described in [30] as

$$Y^{G*} = \arg \min_{Y^{GT} D Y^G = 1, Y^{GT} D 1 = 0} tr\left(\left(Y^G\right)^T L^G Y^G\right) \tag{9}$$

The global laplacian matrix L^G is computed as $L^G = D - A^t$, where D gives the degree of connectivity in the data as $[D]_{i,i} = \sum_{j=1}^{gN} [A^t]_{i,j}$. Computing Y^{G*} in (9) is equivalent to fining eigen vectors of Y^{G*} as $L^G Y^{G*} = \alpha D Y^{G*}$. The spectral embedding Y^{G*} can also be calculated by simply computing the eigen values of L^G. Finally, the laplacian eigen maps L^G and spectral embedding Y^{G*} are used to compute the cost function to find the mixing coefficients as

$$\lambda^i = \frac{tr\left(\left(Y^G\right)^T L^i Y^{G*}\right)}{\sum_{i=1}^{V} tr\left(\left(Y^G\right)^T L^i Y^{G*}\right)} \tag{10}$$

Overall, the convergence of (10) can be decided based on the $l2$ norm between iterations as

$$\sqrt{\sum_{i=1}^{V} \left(\lambda_k^i - \lambda_{k-1}^i\right)^2} < \delta \tag{11}$$

Here, λ_k^i is the value of mixing coefficients at k^{th} iteration and λ_{k-1}^i is the value at $(k-1)^{th}$ iteration. The constant δ is a user defined parameter less than 1. Eventually, the value of λ^i will be different from $\frac{1}{V}$where multiple views are contributing differently to the target view. Finally, by multiplying the obtained mixing coefficients with target features from different views, we obtain a global view feature that closely relates to the target view features. Furthermore, the resulting single view target view feature is highly discriminative across classes and has found have close proximity with all the views from within a class label. The following section describes the datasets and

experiments conducted to ascertain the performance of the proposed method.

IV. Experimentation

The proposed view invariant method, Deep Metric Encoder Decoder (DMED) was trained and tested on multi view skeletal sign (action) video datasets in multiple ratios. We present a one – to – one, one – to – many, many – to – one and many – to – many cross view training and testing approaches on DMED. Further, we compare the results of our approach with other state – of – the – art multi view methods. Finally, multiple CNNs architecture's for classification were tested to check the robustness of the proposed feature extraction process in generating view invariant features.

A. Skeletal Video Datasets and Evaluation Metrics

The multi view sign language dataset KLEF3DSL_2Dskeletal with $V = 15$ views, 200 classes is generated at KL Biomechanics and Vision Computing Research Centre using 3D motion capture technology [8]. Further, the proposed model is evaluated on multi view benchmark skeletal action datasets such as NTU RGB-D [9], SBU Kinect Interaction [10], KLYoga3D [11] and KL3D_MVaction [12]. A small subset of data sample from KLEF3DSL_2Dskeletal is presented in Fig. 4 for a sign basketball. In this work we are limiting our views to 15 due to computational constraints. The training testing ratios are kept constant across all networks and datasets. The selected train test ratios are one – to – one and one – to – many. The remaining views were also evaluated but are not presented here as they have not produced any noticeable performance changes when compared to the selected ones. Since there are no multi view sign language datasets, we evaluated our model on multi view benchmark action datasets. Despite the availability of huge classes in action datasets, we selected only 40 action classes for training with 15 views from each class for maintaining uniformity during comparison. In some cases, unavailability of views has prompted us to generate random views by altering the viewing angles of joints. Here, the evaluation is performed independent of the type of view in which the action is recorded. Fig. 5(a), (b) and (c) shows samples from NTU RGB-D, KL3D_MVaction and KLYoga3D dataset respectively. We used mean recognition accuracy (mRA) for performance evaluations.

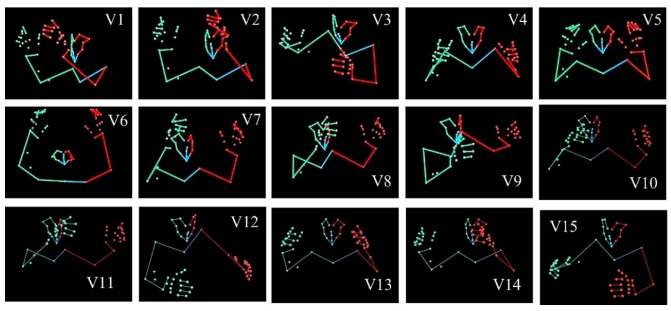

Fig. 4. KLEF3DSL_2Dskeletal Sign Language Video Dataset. A Sample Frame in 15 Different Views from the Skeletal Video Sign "Basketball".

The first 3D CNN network in Fig. 2 extracts the features from skeletal sign (action) video datasets. The network in Fig. 2 is trained on all the available views with similar hyper parameters except for the learning rate and number of epochs. The learning rate for KLEF3DSL_2Dskeletal sign language video dataset is 0.001 and it was 0.005 for all other action datasets. However, the KLYoga3D was trained on a learning rate of 0.0001 for 200 epochs due to large number of skeletal joints. The remaining datasets were trained for 150 epochs. The maximum recognition accuracy achieved during training was around 0.973 for KLEF3DSL_2Dskeletal sign language, 0.942 for NTU RGB-D, 0.845 for SBU Kinect Interaction, 0.902 for KLYoga3D and 0.985 for KL3D_MVaction datasets respectively. Consequently, these individual view trained 3D CNN streams will be inferenced for all dataset samples to generate global view features which represent all views within a class label.

To accomplish the proposed objectives of MVSTFE, we select a target view from each class for inferencing on the trained 3D CNN in Fig. 2 as shown in Fig. 3. The output of Fig. 3 are the features extracted from each of the individual views for the inputted target view. These target view features are combined using the non – uniform linear combiner by computing the value of linear combination value λ^i using spectral embedding of Laplacian eigenmaps. The hyperparameter $(\delta)$for MVSTFE on KLEF3DSL_2Dskeletal$(\delta = 0.54)$, NTU RGB-D$(\delta = 0.71)$, SBU Kinect Interaction $(\delta = 0.94)$, KLYoga3D $(\delta = 0.83)$ and KL3D_MVaction $(\delta = 0.57)$ is selected iteratively. Finally, the generated combined view target features are used for classification. Specifically, to test the robustness of the features in the classification process, we standardized it by training and inferencing on benchmark CNN architectures. However, these architectures are miniaturized in layers and depth to source the feature inputs of size 100×100. Moreover, the regular 2D Convolutional layers in these models were replaced with 3D layers. This has been done to directly extract spatio temporal features from the network. To demonstrate the actual usefulness of these view invariant features, which resulted in the formulation of multiple performance evaluation procedures on the classifier as presented in the following sections.

B. One – to – One Classifier Performance Evaluation

The one – to – one cross view recognition experiment is conducted by training the classifier in Fig. 6. with one view global target feature representing all views and inferencing on a different views. Specifically, the key aspect of this experiment is to test the robustness of the generated view invariant features in estimating a class label based on its constituent views on which it is formulated. To demonstrate this, we designed a CNN network inspired from VGG-16 with 6 convolutional layers, 3 maximum pooling, one flatten and 2 dense layers. The network is trained with the generated view invariant features in each class and tested with view specific features. Consequently, we selected the learning rate of 0.01 for this network with categorical cross entropy loss and Adam optimizer. Subsequently, the above procedure is repeated for all datasets with the same hyper parameters. Furthermore, three benchmark architectures such as Inception – V4, GoogleNet and ResNet – 50 were trained and tested. However, vanishing gradients and overfitting problem were eliminated by re-designing the

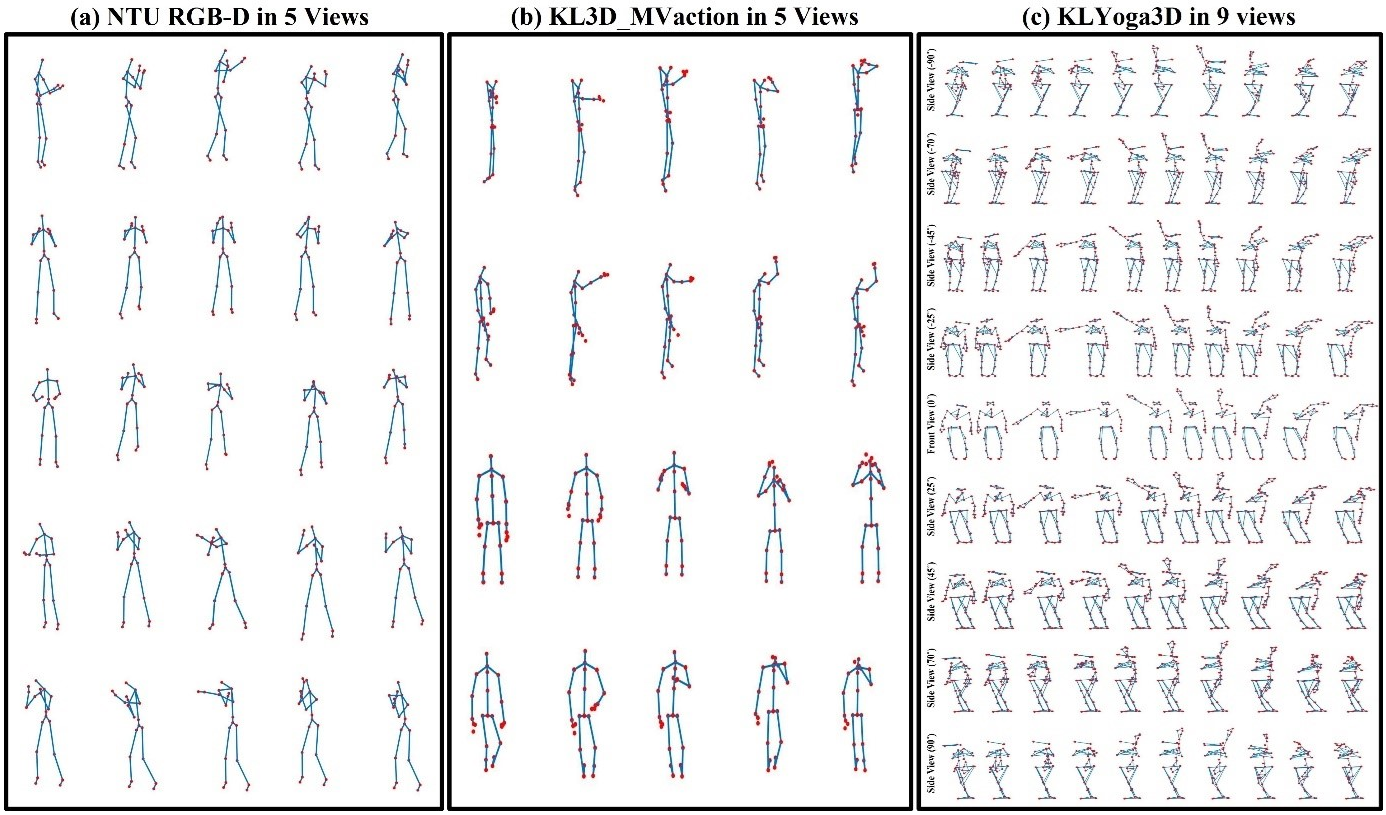

Fig. 5. Multi View Benchmark Action Datasets for Model Evaluation.

architectures with only half the layers than the original models. On the other hand, the structure of the original models were preserved to achieve highest performance. Eventually, mRA is computed during inferencing and the 10-fold maximum value is presented in Table I for all the datasets.

After examining the mRA in Table I, it is evident that all the models perform well on test views that have more visual information when compared to views with overlapping joints. The outcomes from Table I also suggests that the view target global features have shown to reduce false positives in all

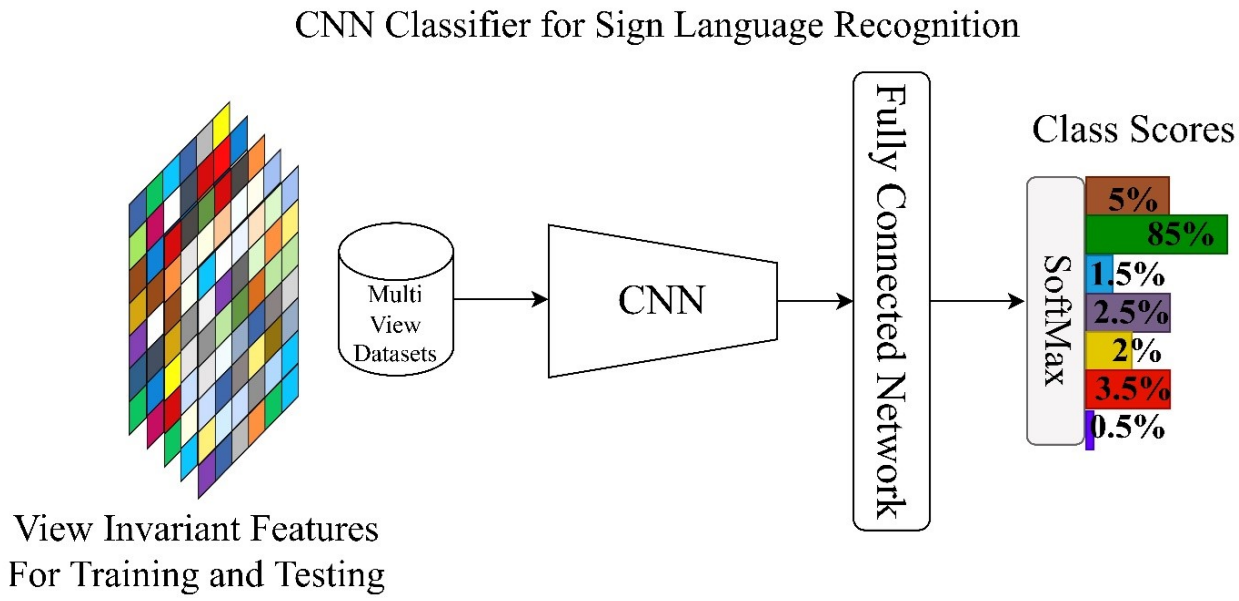

Fig. 6. CNN Architecture for Classification.

TABLE I. ONE – TO – ONE PERFORMANCE EVALUATION OF THE SELECTED CLASSIFIERS ON SKELETAL VIDEO DATASETS TRAINED WITH THE ONE VIEW OF TARGET VIEW FEATURE AND TESTED WITH ALL SPECIFIC VIEW FEATURES. THE PERFORMANCE OF THE CLASSIFIER IS MEASURED USING MRA

Classifiers	Views Datasets	V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15
Tiny VGG – 16	KLEF3DSL_2Dskeletal	0.599	0.619	0.634	0.615	0.664	0.604	0.614	0.559	0.544	0.552	0.554	0.635	0.666	0.596	0.622
	NTU RGB-D	0.578	0.631	0.64	0.62	0.676	0.622	0.6	0.601	0.599	0.624	0.594	0.611	0.662	0.602	0.661
	SBU Kinect Interaction	0.58	0.593	0.589	0.601	0.584	0.591	0.595	0.531	0.529	0.5	0.52	0.566	0.619	0.602	0.59
	KLYoga3D	0.622	0.629	0.663	0.652	0.698	0.652	0.641	0.595	0.601	0.629	0.619	0.689	0.701	0.601	0.622
	KL3D_MVaction	0.619	0.604	0.601	0.629	0.609	0.59	0.616	0.576	0.563	0.565	0.559	0.626	0.649	0.601	0.62
Inception - V4	KLEF3DSL_2Dskeletal	0.671	0.691	0.706	0.687	0.736	0.676	0.686	0.631	0.616	0.624	0.626	0.707	0.738	0.668	0.694
	NTU RGB-D	0.65	0.703	0.712	0.692	0.748	0.694	0.672	0.673	0.671	0.696	0.666	0.683	0.734	0.674	0.733
	SBU Kinect Interaction	0.652	0.665	0.661	0.673	0.656	0.663	0.667	0.603	0.601	0.572	0.592	0.638	0.691	0.674	0.662
	KLYoga3D	0.694	0.701	0.735	0.724	0.77	0.724	0.713	0.667	0.673	0.701	0.691	0.761	0.773	0.673	0.694
	KL3D_MVaction	0.691	0.676	0.673	0.701	0.681	0.662	0.688	0.648	0.635	0.637	0.631	0.698	0.721	0.673	0.692
GoogleNet	KLEF3DSL_2Dskeletal	0.708	0.728	0.743	0.724	0.773	0.713	0.723	0.668	0.653	0.661	0.663	0.744	0.775	0.705	0.731
	NTU RGB-D	0.687	0.74	0.749	0.729	0.785	0.731	0.709	0.71	0.708	0.733	0.703	0.72	0.771	0.711	0.77
	SBU Kinect Interaction	0.689	0.702	0.698	0.71	0.693	0.7	0.704	0.64	0.638	0.609	0.629	0.675	0.728	0.711	0.699
	KLYoga3D	0.731	0.738	0.772	0.761	0.807	0.761	0.75	0.704	0.71	0.738	0.728	0.798	0.81	0.71	0.731
	KL3D_MVaction	0.728	0.713	0.71	0.738	0.718	0.699	0.725	0.685	0.672	0.674	0.668	0.735	0.758	0.71	0.729
ResNet - 50	KLEF3DSL_2Dskeletal	0.665	0.718	0.727	0.707	0.763	0.709	0.687	0.688	0.686	0.711	0.681	0.698	0.749	0.689	0.748
	NTU RGB-D	0.667	0.68	0.676	0.688	0.671	0.678	0.682	0.618	0.616	0.587	0.607	0.653	0.706	0.689	0.677
	SBU Kinect Interaction	0.709	0.716	0.75	0.739	0.785	0.739	0.728	0.682	0.688	0.716	0.706	0.776	0.788	0.688	0.709
	KLYoga3D	0.706	0.691	0.688	0.716	0.696	0.677	0.703	0.663	0.65	0.652	0.646	0.713	0.736	0.688	0.707
	KL3D_MVaction	0.735	0.755	0.77	0.751	0.8	0.74	0.75	0.695	0.68	0.688	0.69	0.771	0.802	0.732	0.758

TABLE II. MANY – TO – ONE PERFORMANCE EVALUATION OF THE CLASSIFIERS TRAINED WITH MULTIPLE SETS OF TRAINING VIEWS AND TESTED WITH ONLY ONE TARGET VIEW FEATURE GENERATED USING MVSTFE

Classifiers	Training Views Datasets	1	1	3	4	5	6	7	8	9	10	11	12	13	14	15
Tiny VGG – 16	KLEF3DSL_2Dskeletal	0.597	0.602	0.612	0.613	0.662	0.67	0.674	0.698	0.71	0.731	0.757	0.767	0.817	0.838	0.871
	NTU RGB-D	0.587	0.607	0.623	0.65	0.67	0.696	0.707	0.727	0.748	0.77	0.794	0.812	0.84	0.852	0.893
	SBU Kinect Interaction	0.58	0.596	0.608	0.605	0.617	0.628	0.66	0.687	0.703	0.724	0.756	0.785	0.816	0.839	0.866
	KLYoga3D	0.585	0.599	0.613	0.63	0.65	0.664	0.674	0.699	0.716	0.738	0.77	0.788	0.797	0.827	0.872
	KL3D_MVaction	0.586	0.608	0.612	0.617	0.65	0.657	0.667	0.7	0.727	0.737	0.76	0.77	0.8	0.832	0.861
Inception - V4	KLEF3DSL_2Dskeletal	0.652	0.657	0.667	0.668	0.717	0.725	0.729	0.732	0.744	0.765	0.791	0.801	0.851	0.872	0.905
	NTU RGB-D	0.642	0.662	0.678	0.705	0.725	0.751	0.762	0.761	0.782	0.804	0.828	0.846	0.874	0.886	0.927
	SBU Kinect Interaction	0.635	0.651	0.663	0.66	0.672	0.683	0.715	0.721	0.737	0.758	0.79	0.819	0.85	0.873	0.9
	KLYoga3D	0.64	0.654	0.668	0.685	0.705	0.719	0.729	0.733	0.75	0.772	0.804	0.822	0.831	0.861	0.906
	KL3D_MVaction	0.641	0.663	0.667	0.672	0.705	0.712	0.722	0.734	0.761	0.771	0.794	0.804	0.834	0.866	0.895
GoogleNet	KLEF3DSL_2Dskeletal	0.62	0.625	0.635	0.636	0.685	0.701	0.705	0.708	0.72	0.741	0.767	0.786	0.836	0.857	0.89
	NTU RGB-D	0.61	0.63	0.646	0.673	0.693	0.727	0.738	0.737	0.758	0.78	0.804	0.831	0.859	0.871	0.912
	SBU Kinect Interaction	0.603	0.619	0.631	0.628	0.64	0.659	0.691	0.697	0.713	0.734	0.766	0.804	0.835	0.858	0.885
	KLYoga3D	0.608	0.622	0.636	0.653	0.673	0.695	0.705	0.709	0.726	0.748	0.78	0.807	0.816	0.846	0.891
	KL3D_MVaction	0.609	0.631	0.635	0.64	0.673	0.688	0.698	0.71	0.737	0.747	0.77	0.789	0.819	0.851	0.88
ResNet - 50	KLEF3DSL_2Dskeletal	0.601	0.606	0.616	0.617	0.666	0.674	0.678	0.702	0.714	0.735	0.761	0.771	0.821	0.842	0.875
	NTU RGB-D	0.591	0.611	0.627	0.654	0.674	0.7	0.711	0.731	0.752	0.774	0.798	0.816	0.844	0.856	0.897
	SBU Kinect Interaction	0.584	0.6	0.612	0.609	0.621	0.632	0.664	0.691	0.707	0.728	0.76	0.789	0.82	0.843	0.87
	KLYoga3D	0.589	0.603	0.617	0.634	0.654	0.668	0.678	0.703	0.72	0.742	0.774	0.792	0.801	0.831	0.876
	KL3D_MVaction	0.59	0.612	0.616	0.621	0.654	0.661	0.671	0.704	0.731	0.741	0.764	0.774	0.804	0.836	0.865

classes. Moreover, the proposed work also highlights the used of any single view for testing as against the previous models, where all views are required as input. Consequently, it will be interesting to test the many – to – one cross view performance, where the models are trained with view specific features and tested with only one target view invariant feature.

C. Many – to – One Classifier Performance Evaluation

Here, we train the classifiers with all the views and test it only one target view feature. Table II shows mRA values for multiple sets of training views. The results in Table II show that the performance of the MVSTFE model has increased when trained with multiple view features. On the other hand, Inception – V4 has shown to outperform all other classifiers used for experimentation due to the fact that it contains multiple attention layers for selecting maximally contributing vectors.

D. Comparisons against other View Invariant Generation Techniques

The previous models applied spectral clustering with matrix factorization [28], auto – weighted spectral clustering [7] and multi view temporal ensemble [6] are designed to generate complimentary views and correspondingly reconstructing a global view. Additionally, the number of views used in these models is comparatively lower than our proposed work. Increasing the number of views in the above models will increase the computational complexity, which was reduced in MVSTFE. Table III presents the comparisons of the above multi view recognition methods with MVSTFE.

E. Validation of MVSTFE with State – of – the – Art Multi View Methods

Historical validation of the proposed MVSTFE is performed by comparing it with state – of – the – art multi view methods in Table IV. The methods selected for comparison have applied some kind of deep learning algorithms

TABLE III. PRESENTS THE RESULTS OF [28], [7] AND [6] ALONG WITH OUR PROPOSED MVSTFE MODEL ON BENCHMARK DATASETS

Multi View Algorithms	Classifiers	Tiny VGG – 16		Inception - V4		GoogleNet		ResNet – 50	
	Train Test Methods / Datasets	One – to – one	Many – to – one	One – to – one	Many – to – one	One – to – one	Many – to – one	One – to – one	Many – to – one
Spectral clustering via structured low-rank matrix factorization [28]	KLEF3DSL_2Dskeletal	0.526	0.621	0.598	0.716	0.524	0.675	0.568	0.666
	NTU RGB-D	0.558	0.719	0.627	0.755	0.602	0.736	0.594	0.714
	SBU Kinect Interaction	0.504	0.657	0.575	0.67	0.549	0.648	0.551	0.643
	KLYoga3D	0.559	0.696	0.638	0.777	0.617	0.749	0.611	0.728
	KL3D_MVaction	0.539	0.652	0.611	0.711	0.586	0.7	0.587	0.696
Auto-weighted multi-view clustering via spectral embedding [7]	KLEF3DSL_2Dskeletal	0.623	0.718	0.67	0.813	0.621	0.752	0.665	0.763
	NTU RGB-D	0.655	0.816	0.724	0.852	0.699	0.833	0.691	0.811
	SBU Kinect Interaction	0.601	0.754	0.672	0.767	0.646	0.745	0.648	0.74
	KLYoga3D	0.656	0.793	0.735	0.874	0.714	0.846	0.708	0.825
	KL3D_MVaction	0.636	0.749	0.708	0.808	0.683	0.797	0.684	0.793
Multi-view temporal ensemble [6]	KLEF3DSL_2Dskeletal	0.549	0.674	0.591	0.749	0.554	0.675	0.568	0.671
	NTU RGB-D	0.581	0.702	0.62	0.768	0.599	0.738	0.595	0.715
	SBU Kinect Interaction	0.527	0.652	0.575	0.679	0.552	0.657	0.544	0.648
	KLYoga3D	0.592	0.693	0.637	0.778	0.612	0.751	0.607	0.728
	KL3D_MVaction	0.572	0.65	0.607	0.721	0.589	0.7	0.585	0.692
MVSTFE Proposed	KLEF3DSL_2Dskeletal	0.668	0.793	0.71	0.868	0.673	0.794	0.687	0.79
	NTU RGB-D	0.7	0.821	0.739	0.887	0.718	0.857	0.714	0.834
	SBU Kinect Interaction	0.646	0.771	0.694	0.798	0.671	0.776	0.663	0.767
	KLYoga3D	0.711	0.812	0.756	0.897	0.731	0.87	0.726	0.847
	KL3D_MVaction	0.691	0.769	0.726	0.84	0.708	0.819	0.704	0.811

TABLE IV. COMPARISON AMONG DIFFERENT VIEW-BASED RECOGNITION TECHNIQUES

		V1	V2	V3	V4	V5	V6	V7	V8	V9	V10	V11	V12	V13	V14	V15
NTU RGB+D	[16]	0.608	0.59	0.574	0.61	0.645	0.659	0.574	0.55	0.601	0.572	0.579	0.568	0.581	0.61	0.568
	[17]	0.617	0.599	0.583	0.619	0.654	0.668	0.583	0.559	0.61	0.581	0.588	0.577	0.567	0.604	0.638
	[18]	0.587	0.569	0.553	0.589	0.624	0.638	0.553	0.529	0.58	0.551	0.558	0.547	0.541	0.578	0.612
	[19]	0.616	0.598	0.581	0.618	0.652	0.667	0.581	0.557	0.608	0.579	0.586	0.575	0.576	0.612	0.647
	[21]	0.59	0.572	0.555	0.592	0.626	0.641	0.555	0.532	0.582	0.553	0.56	0.549	0.581	0.618	0.652
	[25]	0.625	0.607	0.59	0.626	0.661	0.675	0.59	0.566	0.617	0.588	0.595	0.584	0.57	0.606	0.641
	MVSTFE	0.723	0.776	0.785	0.765	0.821	0.767	0.745	0.746	0.744	0.769	0.739	0.756	0.807	0.747	0.806
SBU Kinect Interaction	[16]	0.63	0.612	0.595	0.632	0.666	0.681	0.595	0.572	0.623	0.593	0.6	0.589	0.579	0.615	0.65
	[17]	0.628	0.61	0.594	0.63	0.665	0.679	0.594	0.57	0.621	0.592	0.599	0.588	0.549	0.585	0.62
	[18]	0.637	0.619	0.603	0.639	0.674	0.688	0.603	0.579	0.63	0.601	0.608	0.597	0.577	0.614	0.648
	[19]	0.607	0.589	0.573	0.609	0.644	0.658	0.573	0.549	0.6	0.571	0.578	0.567	0.551	0.588	0.622
	[21]	0.636	0.618	0.601	0.638	0.672	0.687	0.601	0.577	0.628	0.599	0.606	0.595	0.586	0.622	0.657
	[25]	0.61	0.592	0.575	0.612	0.646	0.661	0.575	0.552	0.602	0.573	0.58	0.569	0.591	0.628	0.662
	MVSTFE	0.725	0.738	0.734	0.746	0.729	0.736	0.74	0.676	0.674	0.645	0.665	0.711	0.764	0.747	0.735
KLYoga3D	[16]	0.645	0.627	0.61	0.646	0.681	0.695	0.61	0.586	0.637	0.608	0.615	0.604	0.53	0.566	0.601
	[17]	0.65	0.632	0.615	0.652	0.686	0.701	0.615	0.592	0.643	0.613	0.62	0.609	0.539	0.575	0.61
	[18]	0.638	0.62	0.604	0.64	0.675	0.689	0.604	0.58	0.631	0.602	0.609	0.598	0.509	0.545	0.58
	[19]	0.647	0.629	0.613	0.649	0.684	0.698	0.613	0.589	0.64	0.611	0.618	0.607	0.537	0.574	0.608
	[21]	0.617	0.599	0.583	0.619	0.654	0.668	0.583	0.559	0.61	0.581	0.588	0.577	0.511	0.548	0.582
	[25]	0.646	0.628	0.611	0.648	0.682	0.697	0.611	0.587	0.638	0.609	0.616	0.605	0.546	0.582	0.617
	MVSTFE	0.767	0.774	0.808	0.797	0.843	0.797	0.786	0.74	0.746	0.774	0.764	0.834	0.846	0.746	0.767
KL3D_MVaction	[16]	0.62	0.602	0.585	0.622	0.656	0.671	0.585	0.562	0.612	0.583	0.59	0.579	0.551	0.588	0.622
	[17]	0.655	0.637	0.62	0.656	0.691	0.705	0.62	0.596	0.647	0.618	0.625	0.614	0.586	0.622	0.657
	[18]	0.66	0.642	0.625	0.662	0.696	0.711	0.625	0.602	0.653	0.623	0.63	0.619	0.591	0.628	0.662
	[19]	0.598	0.58	0.564	0.6	0.635	0.649	0.564	0.54	0.591	0.562	0.569	0.558	0.53	0.566	0.601
	[21]	0.607	0.589	0.573	0.609	0.644	0.658	0.573	0.549	0.6	0.571	0.578	0.567	0.539	0.575	0.61
	[25]	0.577	0.559	0.543	0.579	0.614	0.628	0.543	0.519	0.57	0.541	0.548	0.537	0.509	0.545	0.58
	MVSTFE	0.764	0.749	0.746	0.774	0.754	0.735	0.761	0.721	0.708	0.71	0.704	0.771	0.794	0.746	0.765
KLEF3DSL_2Dskeletal	[16]	0.606	0.588	0.571	0.608	0.642	0.657	0.571	0.547	0.598	0.569	0.576	0.565	0.537	0.574	0.608
	[17]	0.58	0.562	0.545	0.582	0.616	0.631	0.545	0.522	0.572	0.543	0.55	0.539	0.511	0.548	0.582
	[18]	0.615	0.597	0.58	0.616	0.651	0.665	0.58	0.556	0.607	0.578	0.585	0.574	0.546	0.582	0.617
	[19]	0.62	0.602	0.585	0.622	0.656	0.671	0.585	0.562	0.613	0.583	0.59	0.579	0.551	0.588	0.622
	[21]	0.583	0.565	0.548	0.585	0.619	0.634	0.548	0.524	0.575	0.546	0.553	0.542	0.514	0.551	0.585
	[25]	0.557	0.539	0.522	0.559	0.593	0.608	0.522	0.499	0.549	0.52	0.527	0.516	0.488	0.525	0.559
	MVSTFE	0.744	0.764	0.779	0.76	0.809	0.749	0.759	0.704	0.689	0.697	0.699	0.78	0.811	0.741	0.767

for generation and classification of view video data. Since the data used in these methods were different, we recreated these models from scratch as given in their respective manuscripts. All the experiments were conducted on the benchmark skeletal datasets used in this work with one – to – one train – test pattern. We presented our best result obtained from inception V4 classifier in this comparison. However, the hyper parameters for the comparison networks was adopted from our Inception V4. The proposed MVSTFE has outperformed the existing models as can be seen in Table IV.

V. CONCLUSION

This work proposed a deep learning based spectral embedding method for generating a single global view from a set of multi view features. We trained a 3D CNN on each of the available views and inferencing on a target view video data to extract features. Eventually, these target features are combined linearly by calculating the mixing coefficients for making a global feature representation for all possible views. Consequently, the mixing coefficients are computed using spectral embedding in Laplacian eigen space which preserves proximity between views within the class label. Experimentation has shown that the proposed MVSTEF on 2D video based skeletal sign language dataset and the benchmark action

datasets has outperformed the previous multiview baseline models.

REFERENCES

[1] Razieh Rastgoo, Kourosh Kiani, and Sergio Escalera. Sign language recognition: A deep survey. *Expert Systems with Applications*, 164:113794, 2021.

[2] Imen Jegham, Anouar Ben Khalifa, Ihsen Alouani, and Mohamed Ali Mahjoub. Vision-based human action recognition: An overview and real world challenges. *Forensic Science International: Digital Investigation*, 32:200901, 2020.

[3] Eepuri Kiran Kumar, PVV Kishore, Maddala Teja Kiran Kumar, Dande Anil Kumar, and ASCS Sastry. Three-dimensional sign language recognition with angular velocity maps and connived feature resnet. *IEEE Signal Processing Letters*, 25(12):1860–1864, 2018.

[4] Jun Liu, Gang Wang, Ling-Yu Duan, Kamila Abdiyeva, and Alex C Kot. Skeleton-based human action recognition with global context-aware attention lstm networks. *IEEE Transactions on Image Processing*, 27(4):1586–1599, 2017.

[5] E Kiran Kumar, PVV Kishore, M Teja Kiran Kumar, and D Anil Kumar. 3d sign language recognition with joint distance and angular coded color topographical descriptor on a 2–stream cnn. *Neurocomputing*, 372:40–54, 2020.

[6] Bee Hock David Koh and Wai Lok Woo. Multi-view temporal ensemble for classification of non-stationary signals. *IEEE Access*, 7:32482–32491, 2019.

[7] Shaojun Shi, Feiping Nie, Rong Wang, and Xuelong Li. Auto-weighted multi-view clustering via spectral embedding. *Neurocomputing*, 399:369–379, 2020.

[8] PVV Kishore, D Anil Kumar, AS Chandra Sekhara Sastry, and E Kiran Kumar. Motionlets matching with adaptive kernels for 3-d indian sign language recognition. *IEEE Sensors Journal*, 18(8):3327–3337, 2018.

[9] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1010–1019, 2016.

[10] Meng Li and Howard Leung. Multiview skeletal interaction recognition using active joint interaction graph. *IEEE Transactions on Multimedia*, 18(11):2293–2302, 2016.

[11] Teja Kiran Kumar Maddala, PVV Kishore, Kiran Kumar Eepuri, and Anil Kumar Dande. Yoganet: 3-d yoga asana recognition using joint angular displacement maps with convnets. *IEEE Transactions on Multimedia*, 21(10):2492–2503, 2019.

[12] D Srihari, PVV Kishore, E Kiran Kumar, D Anil Kumar, M Kumar, MVD Prasad, Ch Prasad, et al. A four-stream convnet based on spatial and depth flow for human action classification using rgb-d data. *Multimedia Tools and Applications*, 79(17):11723–11746, 2020.

[13] Neena Aloysius and M Geetha. Understanding vision-based continuous sign language recognition. *Multimedia Tools and Applications*, 79(31):22177–22209, 2020.

[14] Yuling Xing and Jia Zhu. Deep learning-based action recognition with 3d skeleton: A survey, 2021.

[15] Tanveer Hussain, Khan Muhammad, Weiping Ding, Jaime Lloret, Sung Wook Baik, and Victor Hugo C de Albuquerque. A comprehensive survey of multi-view video summarization. *Pattern Recognition*, 109:107567, 2021.

[16] Zan Gao, Hua Zhang, GP Xu, YB Xue, and Alexander G Hauptmann. Multi-view discriminative and structured dictionary learning with group sparsity for human action recognition. *Signal Processing*, 112:83–97, 2015.

[17] Jingjing Zheng, Zhuolin Jiang, and Rama Chellappa. Cross-view action recognition via transferable dictionary learning. *IEEE Transactions on Image Processing*, 25(6):2542–2556, 2016.

[18] Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE international conference on computer vision*, pages 2117–2126, 2017.

[19] Amin Ullah, Khan Muhammad, Tanveer Hussain, and Sung Wook Baik. Conflux lstms network: A novel approach for multi-view action recognition. *Neurocomputing*, 435:321–329, 2021.

[20] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. An attention enhanced graph convolutional lstm network for skeleton-based action recognition. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1227–1236, 2019.

[21] Dongang Wang, Wanli Ouyang, Wen Li, and Dong Xu. Dividing and aggregating network for multi-view action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 451–467, 2018.

[22] PVV Kishore, Manoj VD Prasad, Ch Raghava Prasad, and R Rahul. 4-camera model for sign language recognition using elliptical fourier descriptors and ann. In *2015 International Conference on Signal Processing and Communication Engineering Systems*, pages 34–38. IEEE, 2015.

[23] Imran N Junejo, Emilie Dexter, Ivan Laptev, and Patrick Púrez. Cross-view action recognition from temporal self-similarities. In *European Conference on Computer Vision*, pages 293–306. Springer, 2008.

[24] Hossein Rahmani and Ajmal Mian. Learning a non-linear knowledge transfer model for cross-view action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2458–2466, 2015.

[25] Hossein Rahmani, Arif Mahmood, Du Huynh, and Ajmal Mian. Histogram of oriented principal components for cross-view action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 38(12):2430–2443, 2016.

[26] Jiang Wang, Xiaohan Nie, Yin Xia, Ying Wu, and Song-Chun Zhu. Cross-view action modeling, learning and recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2649–2656, 2014.

[27] Lichen Wang, Zhengming Ding, Zhiqiang Tao, Yunyu Liu, and Yun Fu. Generative multi-view human action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6212–6221, 2019.

[28] Yang Wang, Lin Wu, Xuemin Lin, and Junbin Gao. Multiview spectral clustering via structured low-rank matrix factorization. *IEEE transactions on neural networks and learning systems*, 29(10):4833–4843, 2018.

[29] Chuan Sun, Imran Nazir Junejo, Marshall Tappen, and Hassan Foroosh. Exploring sparseness and self-similarity for action recognition. *IEEE Transactions on Image Processing*, 24(8):2488–2501, 2015.

[30] Yu Kong, Zhengming Ding, Jun Li, and Yun Fu. Deeply learned view-invariant features for cross-view action recognition. *IEEE Transactions on Image Processing*, 26(6):3028–3037, 2017.