IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 24, 2020, accepted May 6, 2020, date of publication May 11, 2020, date of current version May 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993650

Continuous Sign Language Recognition Through Cross-Modal Alignment of Video and Text Embeddings in a Joint-Latent Space

ILIAS PAPASTRATIS, KOSMAS DIMITROPOULOS, DIMITRIOS KONSTANTINIDIS, AND PETROS DARAS, (Senior Member, IEEE)

Visual Computing Lab, Centre for Research and Technology Hellas-Information Technologies Institute, 57001 Thessaloniki, Greece

Corresponding author: Ilias Papastratis (papastrat@iti.gr)

This work was supported by the Greek General Secretariat of Research and Technology under Contract T1EΔK-02469 EPIKOINONO.

ABSTRACT Continuous Sign Language Recognition (CSLR) refers to the challenging problem of recognizing sign language glosses and their temporal boundaries from weakly annotated video sequences. Previous methods focus mostly on visual feature extraction neglecting text information and failing to effectively model the intra-gloss dependencies. In this work, a cross-modal learning approach that leverages text information to improve vision-based CSLR is proposed. To this end, two powerful encoding networks are initially used to produce video and text embeddings prior to their mapping and alignment into a joint latent representation. The purpose of the proposed cross-modal alignment is the modelling of intra-gloss dependencies and the creation of more descriptive video-based latent representations for CSLR. The proposed method is trained jointly with video and text latent representations. Finally, the aligned video latent representations are classified using a jointly trained decoder. Extensive experiments on three well-known sign language recognition datasets and comparison with state-of-the-art approaches demonstrate the great potential of the proposed approach.

INDEX TERMS Computer vision, continuous sign language recognition, cross-modal learning, deep-learning, joint latent space.

I. INTRODUCTION

Sign language (SL) is the primary communication tool for deaf-mute people, making use of gestures produced with the body and perceived with the eyes. SLs have independent vocabularies and grammatical structures just like spoken ones [1]. Signs, which have an internal structure similar to spoken words, are characterized by a combination of hand shapes, positions and motion trajectories, orientations of palm and fingers and facial expressions. The closest meaning of a visual sign is a gloss, which is the fundamental building block of SLs.

Sign Language Recognition (SLR) is the task of recognizing glosses from video captures of sign language. SLR is of great significance to the deaf community, as it enables the communication of deaf people with the world, removing accessibility barriers and improving their social inclusion. The SLR tasks can be divided into two categories: isolated sign language recognition (ISLR) [2]–[5] and continuous sign language recognition (CSLR) [6], [7]. In ISLR, the annotation boundaries of the signs in videos are predefined in a similar way to gesture and action classification. On the other hand, CSLR is more challenging than ISLR since only the temporal order of the gloss sequence is given without any prior segmentation information.

Early machine learning techniques were mainly focusing on isolated gloss classification or gesture spotting. Such techniques were often making use of handcrafted features with temporal modelling methods, such as hidden Markov models (HMM) [8], [9] and conditional random fields [10]. Recently, deep learning methods have shown their potential to outperform conventional machine learning approaches in several computer vision tasks, such as gesture recognition and human action recognition [11]–[13]. As a result, current CSLR approaches take advantage of deep learning concepts, such as convolutional neural networks (CNNs) to capture powerful image and video representations and recurrent neural networks (RNNs) to accurately model

The associate editor coordinating the review of this manuscript and approving it for publication was Sudipta Roy.

temporal dependencies, thus improving recognition accuracy [14]–[17]. However, most CLSR methods either extract frame-level representations that are inadequate for modelling the temporal dynamics (present in a gloss) or they fail to effectively model intra-gloss dependencies. Additionally, there is limited research on ways to exploit the relationship between visual content and text information for improving recognition accuracy in CSLR.

To overcome the aforementioned shortcomings, in this work, a novel unified deep learning framework for CSLR is proposed. The proposed approach consists of two encoders that learn the individual video and text embeddings, which are then projected into a joint latent space through linear transformations. A common loss function is used to align the latent representations and minimize their distance, while the final classification of the aligned video latent representations is performed by a jointly trained decoder. During training, both video and text information is employed, while during inference, only video information is used as input.

More specifically, the main contributions of this work are summarized as follows:

- A novel cross-modal learning approach for video-based CSLR is introduced. The proposed method leverages text information to model intra-gloss dependencies and create more descriptive video-based latent representations that improve the recognition accuracy.
- A new approach for the alignment of video and text embeddings using a joint loss function is proposed. The joint loss function aims to minimize the distance between the corresponding embeddings of the two modalities enabling the creation of a common latent representation. The inference of the aligned video latent representations is finally performed by a jointly trained decoder network.
- The proposed approach is evaluated on three challenging sign language recognition datasets and compared with several state-of-the-art CSLR methods, showing promising results.

The remainder of this paper is organized as follows. In Section II related work in SLR is described. The components of the proposed CSLR method and the optimization process are described in Section III and Section IV, respectively. Finally, the implementation details and the experimental results are discussed in Section V, while conclusions are drawn in Section VI.

II. RELATED WORK

Early SLR works were relying on handcrafted feature extraction such as hand shape, appearance and motion trajectories [2], [18], while recent approaches are automatically extracting features with the use of deep neural networks. Most CSLR methods consist of a feature extractor followed by a sequence modelling module. In [9], [19], the authors employed a CNN feature extractor, whose output is fed into a HMM for temporal modelling. They used frame-state alignments generated from the HMM to train the CNN. Later, they extended their work by incorporating a long short-term memory (LSTM) unit in top of a CNN [20]. In their most recent work [16], the authors introduced two additional streams of cropped hands and mouth modalities. The full architecture was trained iteratively by frame-state alignments provided by the HMM. However, frame-state alignments can be noisy due to the lack of frame-level ground truth annotations and such methods are forced to make strong initial assumptions on gloss boundaries in order to overcome HMM's limited learning capacity [14].

Other methods make use of connectionist temporal classification (CTC) [21], which is designed for sequence labelling problems, such as speech recognition and handwriting recognition. CTC can effectively deal with weakly labelled data, making it appropriate for continuous SLR. Camgoz *et al*. [22] were among the first ones who proposed a shallow CNN-LSTM architecture trained end-to-end with CTC. In [23], the authors employed a 2D-CNN-LSTM architecture with CTC loss in parallel with a gloss-detection network to refine predictions. Later, they extended their work using temporal convolutions and a new iterative training scheme, achieving superior performance in CSLR datasets [14]. However, the major weakness of CTC is the conditional independence assumption and therefore it fails to model intra-gloss dependencies.

On the other hand, a crucial issue for SLR is video representation, i.e., the extraction of video embeddings. 3D-CNNs have strong video representation capabilities since they extract motion features unlike 2D-CNNs and have also been adopted in CSLR task. Huang *et al*. [24] proposed a 3D-CNN network along with a hierarchical attention network for recognition. Yang *et al*. [25] proposed a shallow hybrid CNN with 2D and 3D convolutions followed by two LSTM networks for sequence modelling at gloss and sentence level respectively, which can be trained end-to-end with CTC loss. Pu *et al*. [26] adopted a 3D-ResNet to extract video representations with stacked dilated temporal convolutions instead of a LSTM to alleviate the problem of backpropagation through a recurrent network. In [17], the authors proposed a framework with a 3D-ResNet integrating an encoder-decoder network with a CTC decoder, jointly trained and aligned with soft-DTW (Dynamic Time Warping) [27]. Pseudo-labels were inferred from the decoders' alignment to train the 3D-CNN. In [28], the authors adopted the I3D architecture from the action recognition field [12] with a gated recurrent unit (GRU) for sequence modelling. The whole architecture was trained iteratively with CTC and a new dynamic pseudo-labelling method. However, training 3D architectures with limited data in a weakly supervised setting is challenging. In [29], the authors used deep temporal convolution layers instead of RNN to model the short- and long-term dependencies simultaneously. They utilized several classifiers in each temporal convolution layer and fused the predictions in a CTC decoder for increased performance. Guo *et al*. [30] proposed a hierarchical adaptive recurrent network with temporal pooling and attention-aware weighting mechanisms.

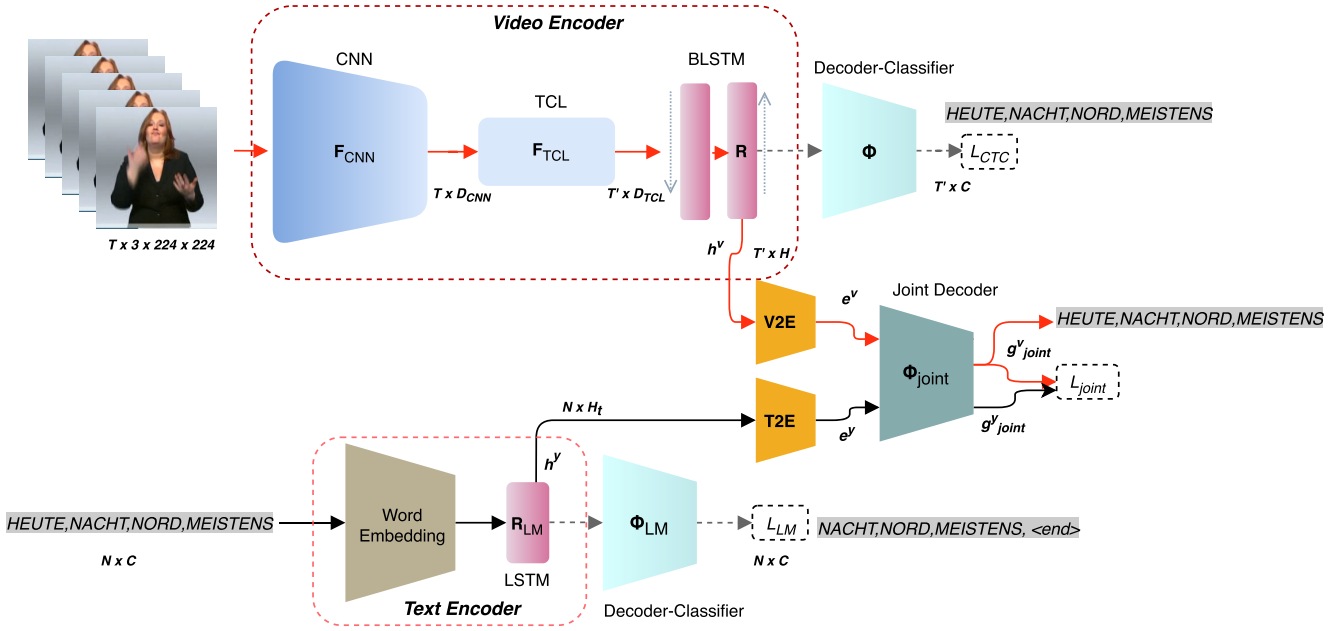

FIGURE 1. Overview of the proposed method. The text embeddings e^y are used only during training, while during inference only video information (red arrows) is fed to the network for the estimation of output probability distribution g^v_{joint}.

In [31], the authors fused 2D and 3D-CNN features to learn short-term temporal dependencies and a new decoding algorithm, which learns a temporal mapping among features, sign labels and the generated gloss sequence.

Cross-modal methods have been successfully applied to various fields, such as action recognition and video captioning. In [32], the authors employed transfer learning from the image domain to enhance video action recognition, while in [33], the authors proposed a Generative Adversarial Network that learns a common feature space of images and videos to improve recognition accuracy. Finally, in [34], the authors integrated images and videos into a common representation using cross-modal similarity metrics to enhance the action recognition accuracy. In this work, a cross-modal method for CSLR is proposed, which takes advantage of the ability of CTC to handle weakly labeled data, while simultaneously leverages text information to model intra-gloss dependencies through the cross-modal alignment of video and text embeddings.

III. PROPOSED METHOD

In this work, a video encoder is proposed that consists of a CNN for spatial feature extraction, stacked 1D temporal convolution layers (TCL) for short-term temporal modelling and a bidirectional long short-term memory (BLSTM) units for global context learning. Furthermore, a text encoder is implemented using a unidirectional LSTM to model the sequences of sign language glosses. The outputs of both encoders are projected into a joint latent space through linear transformations. In addition, alignment is achieved by using a common loss function for minimizing the distance of video and text embeddings. An overview of the proposed approach is depicted in Figure 1. In the remainder of this section the encoding of each modality is initially formulated and then the joint latent space representation is described.

A. VIDEO ENCODER

The proposed video encoder adopts a 2D-CNN followed by temporal convolution layers to extract spatiotemporal features from the input video. The extracted features are then processed through a BLSTM layer to learn long-term dependencies over all timesteps.

1) FEATURE EXTRACTION

Let $\boldsymbol{x} = (x_1, \cdots, x_T)$ be the input frame sequence of length T, where x_τ is the τ^{th} frame of the video sequence. The CNN represented as function $\mathbf{F}_{\text{CNN}}$ extracts a spatial representation $f_\tau = \mathbf{F}_{\text{CNN}}(x_\tau)$ for each frame with $f_\tau \in \mathbb{R}^{D_{CNN}}$, where D_{CNN} is the feature dimension of the CNN. Therefore, all features are represented as follows:

$$\boldsymbol{f} = (f_1, f_2, \cdots, f_T) = \{\mathbf{F}_{\text{CNN}}(x_\tau)\}_{\tau=1}^{T} \tag{1}$$

The feature sequence $\boldsymbol{f} \in \mathbb{R}^{T \times D_{CNN}}$ is processed by the TCL module, represented by the function $\mathbf{F}_{\text{TCL}}$, modelling the temporal dependencies between adjacent frames. The TCL module consists of stacked 1D convolutions and pooling layers that learn short-term temporal dependencies between frames. The receptive field of the TCL module depends on the layers' filter size k, pooling size p, stride s and dilation factor d. The TCL module extracts a spatiotemporal feature sequence represented as:

$$\boldsymbol{r} = (r_1, r_2, \cdots, r_{T'}) = \{\mathbf{F}_{\text{TCL}}(f_\tau)\}_{\tau=1}^{T}, \tag{2}$$

where $\boldsymbol{r} \in \mathbb{R}^{T' \times D_{TCL}}$, D_{TCL} is the feature dimension of the TCL module and $T' = T/\sigma$ is the length of the extracted spatiotemporal sequence, with σ depending on the receptive field of the TCL module.

2) SEQUENCE MODELLING

Recurrent neural networks have been successfully applied to many sequence-to-sequence problems, such as speech recognition and neural machine translation. LSTMs are able to learn long-term temporal dependencies avoiding vanishing gradients due to backpropagation through all timesteps in contrast to traditional RNNs. However, LSTMs compute the current output based only on previous timesteps. In CSLR, the signed video is mapped to a sentence with grammatical rules, meaning that each sign depends on the previous and succeeding context. To this end, a BLSTM layer is chosen instead of a unidirectional LSTM layer to learn the complete sequential information over all timesteps. Using $\mathbf{R}$ to represent the BLSTM layer with H hidden units, the outputs are computed as:

$$\boldsymbol{h}^v = (h_1^v, h_2^v, \cdots, h_{T'}^v) = \{\mathbf{R}(r_t)\}_{t=1}^{T'} \tag{3}$$

where $\boldsymbol{h}^v \in \mathbb{R}^{T' \times H}$ are the concatenated forward and backward hidden state sequences. The concatenated hidden state sequence is passed through a fully connected and a softmax layer denoted as $\mathbf{\Phi}$ that produces the gloss label probabilities from a given vocabulary of C classes.

$$\boldsymbol{g}^v = (g_1^v, g_2^v, \cdots, g_{T'}^v) = \{\mathbf{\Phi}(h_t^v)\}_{t=1}^{T'} \tag{4}$$

where $\boldsymbol{g}^v \in [0,1]^{T' \times C}$ is the output probability distribution among C classes.

B. TEXT ENCODER

The proposed text encoder is a RNN Language Model (RNNLM) [35], [36] that models the probability of a word occurrence under the condition of its previous words in a sentence, i.e., it aims to learn the structure and syntax of sign language. The model maximizes the log-likelihood of the target sentence given the hidden states and the previous words. The text encoder employs a word embedding layer and a LSTM layer with H_{text} hidden units. Each gloss $\boldsymbol{y}_k$ of the input sentence is passed through a word embedding layer, which is a fully connected layer that learns a linear projection from discrete gloss categories to a denser vector denoted as we_k. In other words, the gloss $\boldsymbol{y}_k$, which is represented by a unique one-hot vector, is transformed into a continuous vector with smaller dimension compared to the gloss vocabulary size. The hidden state of the LSTM layer h_k^y encapsulates the history of the sentence up to gloss $\boldsymbol{y}_k$, i.e., all previous words. The hidden states are generated as follows,

$$\boldsymbol{h}^y = (h_1^y, h_2^y, \cdots, h_K^y) = \{\mathbf{R}_{\text{LM}}(we_k, h_{k-1}^y)\}_{k=1}^{K} \tag{5}$$

where $\boldsymbol{h}^y \in \mathbb{R}^{K \times H_{text}}$ and K is the length of the sentence. Then, the hidden states of the LSTM layer are passed through a fully connected and a softmax layer, denoted as $\mathbf{\Phi}_{\text{LM}}$ to output the gloss label probabilities as:

$$\boldsymbol{g}^y = (g_1^y, g_2^y, \cdots, g_K^y) = \{\mathbf{\Phi}_{\text{LM}}(h_k^y)\}_{k=1}^{K} \tag{6}$$

where $\boldsymbol{g}^y \in [0,1]^{K \times C}$ is the output probability distribution among C classes.

C. JOINT LATENT SPACE

The hidden states $\boldsymbol{h}^v = \{h_t^v\}_{t=1}^{T'}$ of the video encoder and the hidden states $\boldsymbol{h}^y = \{h_k^y\}_{k=1}^{K}$ of the text encoder are mapped into the joint latent space through two mapping networks, **V2E** and **T2E**, respectively. Each mapping network consists of a fully connected layer that computes the following latent representations:

$$\boldsymbol{e}^v = \{\mathbf{V2E}(h_t^v)\}_{t=1}^{T'} = \{W^v h_t^v + b^v\}_{t=1}^{T'} \tag{7}$$

$$\boldsymbol{e}^y = \{\mathbf{T2E}(h_k^y)\}_{k=1}^{K} = \{W^y h_k^y + b^y\}_{k=1}^{K} \tag{8}$$

where Z is the latent space dimension, $\boldsymbol{e}^v \in \mathbb{R}^{T' \times Z}$ and $\boldsymbol{e}^y \in \mathbb{R}^{K \times Z}$ are the video and text representations in the joint latent space, respectively.

The above latent representations are passed through a joint decoder $\mathbf{\Phi}_{\text{joint}}$ that consists of a fully-connected and a softmax layer to obtain gloss probabilities. Both modalities share the same joint decoder weights to enforce a common representation between them.

$$\boldsymbol{g}_{\text{joint}}^v = \{\mathbf{\Phi}_{\text{joint}}(e_t^v)\}_{t=1}^{T'} \tag{9}$$

$$\boldsymbol{g}_{\text{joint}}^y = \{\mathbf{\Phi}_{\text{joint}}(e_k^y)\}_{k=1}^{K} \tag{10}$$

where $\boldsymbol{g}_{\text{joint}}^v \in [0,1]^{T' \times C}$ and $\boldsymbol{g}_{\text{joint}}^y \in [0,1]^{K \times C}$ are the output probability distributions computed from the video and text latent representations, respectively.

IV. OPTIMIZATION

A. LEARNING EMBEDDINGS

The proposed framework employs a CTC loss function to train the video encoder given the frame sequence and the joint decoder given the video latent space representations, respectively. The objective of using the CTC loss is to maximize the sum of probabilities of all possible mappings between input and target sequences. CTC extends the vocabulary C with a *blank* label "$-$", representing the silence or transition between two consecutive labels. The extended vocabulary can be defined as $V = C \cup \{blank\}$. Given a frame sequence $\boldsymbol{x} = \{\boldsymbol{x}_\tau\}_{\tau=1}^{T}$ of length T, the proposed framework outputs two gloss probability distributions $\boldsymbol{g}^v$ and $\boldsymbol{g}_{\text{joint}}^v$ with length T' to predict the corresponding sequence of target glosses $\boldsymbol{y} = \{\boldsymbol{y}_k\}_{k=1}^{K}$ of length K.

The emission probability $p(j, t|\boldsymbol{x})$ of label j at time-step t is denoted as $g_{j,t}$ and can be modelled either from the video encoder or the joint decoder. An alignment path is defined as $\pi = \{\pi_t\}_{t=1}^{T'}$, where label $\pi_t \in V$. The posterior probability of a CTC alignment path π is defined as:

$$p(\pi|\boldsymbol{x}) = \prod_{t=1}^{T'} g_{\pi_t,t} \tag{11}$$

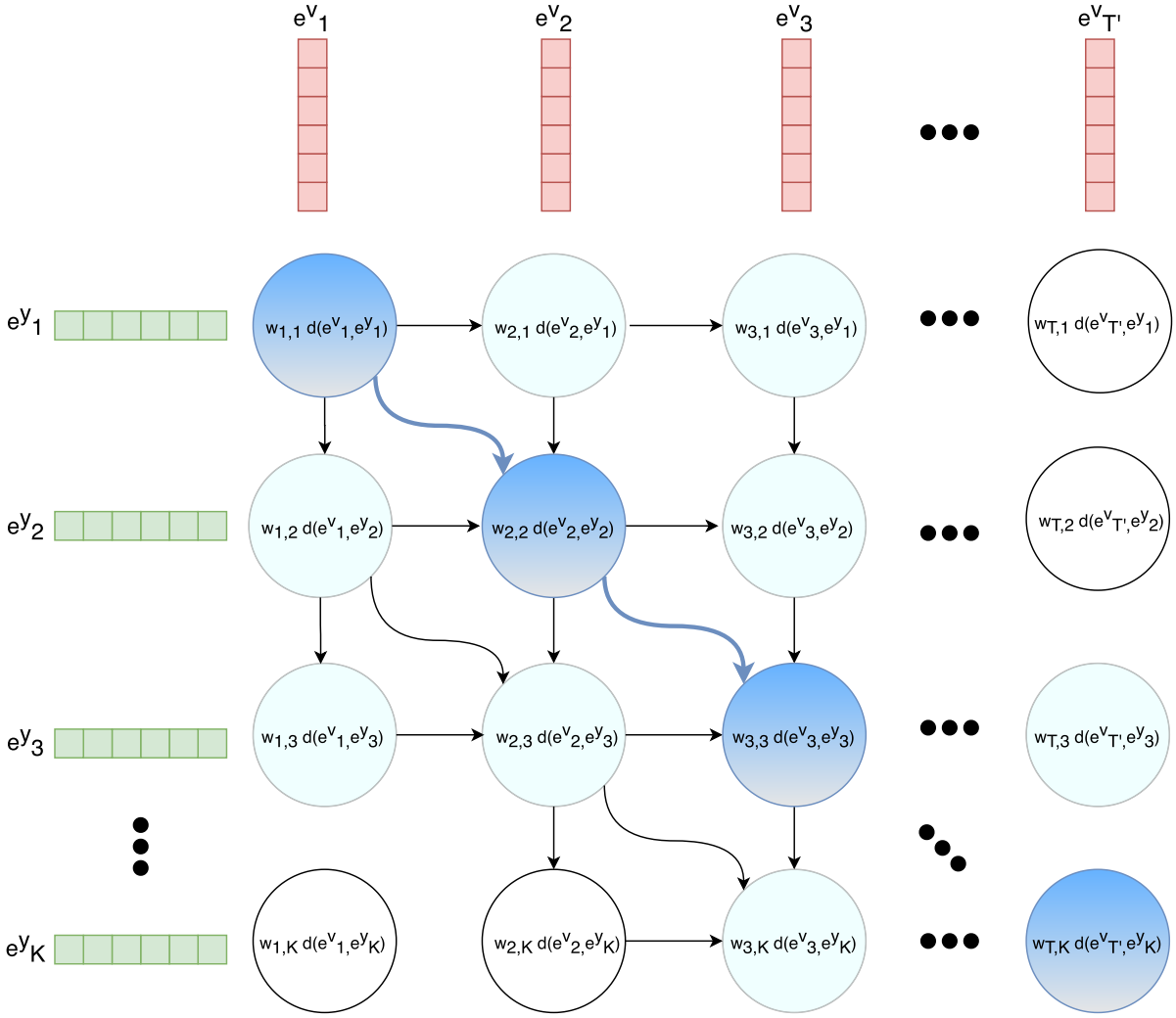

FIGURE 2. Alignment of video and text embeddings based on the loss function L_{map}. The dominant alignment path is marked with blue color, while softer and infeasible alignments are shown with cyan and white colors respectively.

The alignment path π is mapped to the target sequence $\boldsymbol{y}$ with a many-to-one mapping operation B that removes repeated labels and blanks from the given path. Subsequently, an inverse operation $B^{-1}(\boldsymbol{y}) = \{\pi | B(\pi) = \boldsymbol{y}\}$ is used that represents all the possible alignments corresponding to target labels $\boldsymbol{y}$. The conditional probability of $\boldsymbol{y}$ is defined as the sum of the probabilities of all corresponding paths π:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{\pi \in B^{-1}(\boldsymbol{y})} p(\pi|\boldsymbol{x}) \tag{12}$$

Furthermore, to allow for blank labels in the computed alignment paths, a modified label sequence $\boldsymbol{y'}$ of length $K' = 2K+1$ is defined and used as target sequence in the proposed method by adding blanks before and after each label in $\boldsymbol{y}$. Since single labelling can be derived from a huge amount of paths, a method is required to efficiently calculate $p(\boldsymbol{y}|\boldsymbol{x})$. CTC employs dynamic programming to compute the sum over different paths for a single labelling iteratively, using forward and backward variables $\boldsymbol{\alpha} \in \mathbb{R}^{T' \times K'}$ and $\boldsymbol{\beta} \in \mathbb{R}^{T' \times K'}$, respectively.

The total probability $\boldsymbol{\alpha}_{t,s}$ of $\boldsymbol{y'}_{1:s}$ (i.e., the first s symbols of modified label sequence $\boldsymbol{y'}$) at time-step t is defined as:

$$\boldsymbol{\alpha}_{t,s} = \sum_{B(\pi_{1:t}) = \boldsymbol{y'}_{1:s}} \prod_{t'=1}^{t} g^{t'}_{\pi_{t'}} \tag{13}$$

and correspondingly the total probability $\boldsymbol{\beta}_{t,s}$ of $\boldsymbol{y'}_{s:K'}$ at time-step t is equal to:

$$\boldsymbol{\beta}_{t,s} = \sum_{B(\pi_{t:T'}) = \boldsymbol{y'}_{s:K'}} \prod_{t'=t}^{T'} g^{t'}_{\pi_{t'}} \tag{14}$$

Therefore, $p(\boldsymbol{y}|\boldsymbol{x})$ for any t is calculated as follows:

$$p(\boldsymbol{y}|\boldsymbol{x}) = \sum_{s=1}^{K'} \frac{\boldsymbol{\alpha}_{t,s}\boldsymbol{\beta}_{t,s}}{g^{t}_{y'_s}} \tag{15}$$

The objective function L_{CTC} that guides the training process, is derived from the principle of maximum likelihood [37] and is used to optimize the video encoder. The loss function of the video encoder is formulated as:

$$L_{CTC} = -\log p(\boldsymbol{y}|\boldsymbol{x}) \tag{16}$$

The text encoder is used as a language model. The objective is to maximize the probability of the current word given the previous hidden states and it is trained using the cross-entropy criterion denoted as L_{LM}.

$$L_{LM} = -\frac{1}{K}\sum_{k=1}^{K}\log p(\boldsymbol{y}_k|h^{y}_{k-1}) \tag{17}$$

B. LATENT SPACE ALIGNMENT

The cross-modal alignment aims to jointly encode and project video and text information into a common latent space by minimizing the distance between video and text embeddings. Due to the different length of video and text embedding sequences, the alignment paths are calculated from the non-blank probabilities $\boldsymbol{\alpha}'$ and $\boldsymbol{\beta}' \in \mathbb{R}^{T'\times K}$ of the CTC forward-backward algorithm [38]. Non-blank probabilities $\boldsymbol{\alpha}'$ and $\boldsymbol{\beta}'$ are calculated from $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, respectively, by removing the probabilities that correspond to blank labels of the modified label sequence $\boldsymbol{y}'$ recursively as:

$$\boldsymbol{\alpha}'_{t,k} = \boldsymbol{\alpha}_{t,2k+1} \tag{18}$$

$$\boldsymbol{\beta}'_{t,k} = \boldsymbol{\beta}_{t,2k+1} \tag{19}$$

Then, the soft alignments $\boldsymbol{w} \in \mathbb{R}^{T'\times K}$ are defined as:

$$\boldsymbol{w}_{t,k} = \frac{\boldsymbol{\beta}'_{t,k}\boldsymbol{\alpha}'_{t,k}}{\sum_{k=1}^{K}\boldsymbol{\beta}'_{t,k}\boldsymbol{\alpha}'_{t,k}} \tag{20}$$

Intuitively, $\boldsymbol{w}_{t,k}$ is the probability of gloss k in target sequence $\boldsymbol{y}$ occurring at time-step t and is used as a weighting factor between the possible alignments. To minimize the distance of the video and text latent representations, a mapping loss is defined as:

$$L_{map} = \frac{1}{K\cdot T'}\sum_{k=1}^{K}\sum_{t=1}^{T'}\boldsymbol{w}_{t,k}d(\boldsymbol{e}^{v}_{t},\boldsymbol{e}^{y}_{k}), \tag{21}$$

with $d(\boldsymbol{e}^{v}_{t},\boldsymbol{e}^{y}_{k}) = ||\,\boldsymbol{e}^{v}_{t} - \boldsymbol{e}^{y}_{k}\,||^2$ being the Euclidean distance between two vectors. The L_{map} function is illustrated in Figure 2. The purpose of the L_{map} loss function is to drive the video and text embeddings closer to one another using the weighting factor $\boldsymbol{w}$ computed by the CTC alignment. The factor $\boldsymbol{w}_{t,k}$ is a probability in the range of [0, 1] that expresses the degree the predicted gloss at time t matches the ground truth gloss $\boldsymbol{y}_k$. When $\boldsymbol{w}_{t,k}$ is high, the corresponding video segment at time t matches the target gloss $\boldsymbol{y}_k$ and L_{map} is significantly affected by the Euclidean distance between the video and text embeddings in an attempt to bring them closer. When $\boldsymbol{w}_{t,k}$ is close to 0, the corresponding video segment at time t is not matched with the target gloss $\boldsymbol{y}_k$ and L_{map} is only slightly affected by the Euclidean distance between the video and text embeddings. In this way, L_{map} aligns the video and text embeddings only when the alignment is meaningful (i.e., the predicted sequence is close to the ground truth and $\boldsymbol{w}_{t,k}$ is close to 1).

The joint decoder is trained using L_{CTC} and L_{LM} for the video and text latent representations, respectively. The video and text encoders, the latent space mapping networks and the joint decoder are jointly trained with the following objective function:

$$L_{joint} = L_{map} + aL_{CTC} + bL_{LM} \tag{22}$$

where a, b are tunable hyperparameters to balance the effect of each latent representation in the training procedure.

C. OPTIMIZATION STRATEGY

In this work, a two-stage optimization process is followed. It has been shown that training the video encoder only with L_{CTC} end-to-end has limited contribution to the parameters of CNN as the gradients are vanished after backpropagation through the BLSTM layer due to the chain rules of backpropagation [17], [28]. At the first stage, the proposed video encoder (i.e., 2D-CNN, TCL, BLSTM and Decoder-Classifier modules) is optimized with L_{CTC}. At the second stage, the feature extractor (i.e., 2D-CNN and TCL modules) of the video encoder is optimized using pseudo-labels generated from the soft alignments $\boldsymbol{w}$ with cross-entropy loss as a stronger supervision. Then, the video encoder learns a better video representation and generates more accurate pseudo-labels. The two stages are performed iteratively until no further improvement in recognition error is observed. Both video and text encoders are trained until convergence. Then, the latent space mapping modules are optimized with L_{map} to align and learn the embeddings. After removing the two decoders-classifiers from the video and text encoders, the full architecture (including the latent space and the joint decoder) is trained with L_{joint} loss to fine-tune the proposed CSLR method.

V. EXPERIMENTS

In this section, the implementation details of the proposed method are initially described. Then, experimental results on three well-known CSLR datasets are presented and discussed.

A. EVALUATION

The proposed method is evaluated on three publicly available datasets, RWTH-Phoenix-Weather-2014 [6], RWTH-Phoenix-Weather-2014T [39] and CSL [24]. To evaluate performance in CSLR datasets, the word error rate (WER) metric has been adopted, which measures the similarity between predicted and ground truth gloss sequences. WER calculates the least number of operations needed to transform the aligned predicted sequence to the ground truth and can be defined as:

$$WER = \frac{S + D + I}{N}, \tag{23}$$

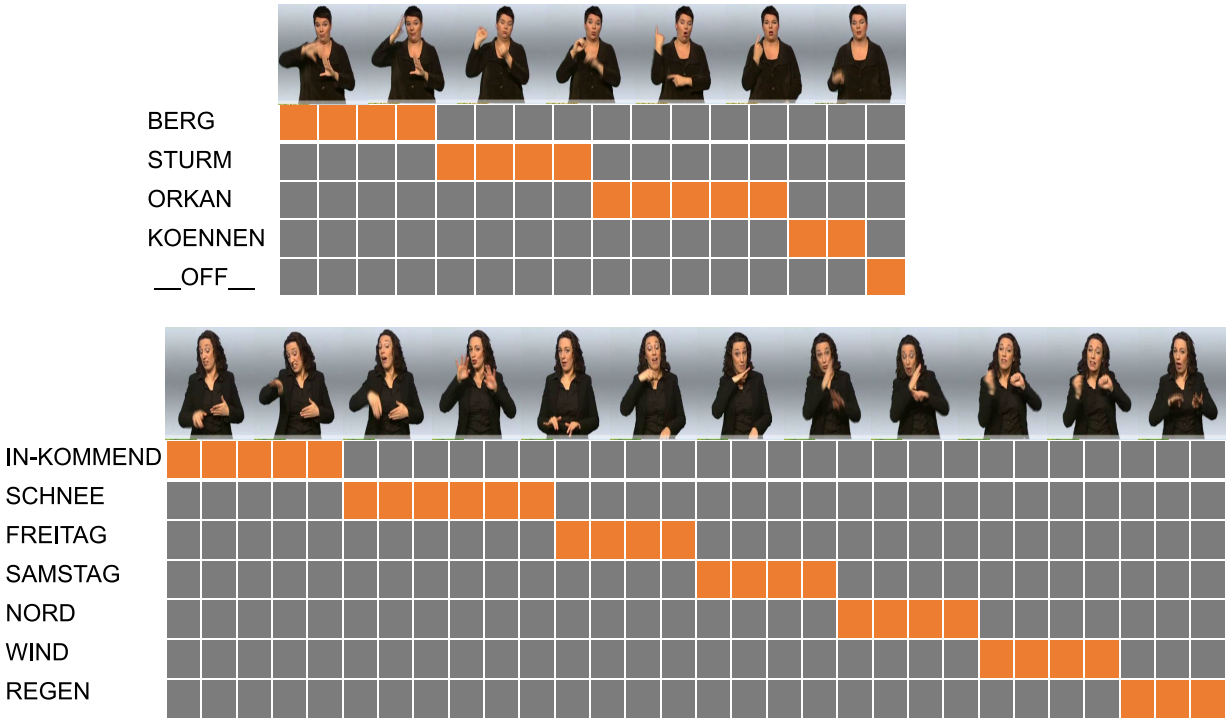

FIGURE 3. Soft alignments computed from w between video and text embeddings for two testing videos of the RWTH-Phoenix-2014 dataset. The dominant alignment paths are highlighted in orange.

where S is the total number of substitutions, D is the total number of deletions, I is the total number of insertions and N is the total number of glosses in the ground truth.

B. IMPLEMENTATION DETAILS

For the proposed video encoder, a 2D-CNN (BN-Inception) network [40] is used that is initialized with weights pretrained on the ImageNet dataset. The kernel and stride sizes of the TCL module are manually tuned to approximately cover the average gloss duration. TCL has two 1D convolutional layers with 1024 filters and two max-pooling layers. For the CSL dataset, the convolutional layers have kernel sizes equal to 7, while the pooling layers have kernel and stride sizes equal to 3 and cover the average gloss duration of 58 frames. For the RWTH-Phoenix-Weather-2014T and RWTH-Phoenix-Weather-2014 datasets, the convolutional layers of the TCL module have kernel sizes equal to 5, while the pooling layers have kernel and stride size equal to 2 resulting in a receptive field of 16 frames. The BLSTM layer consists of 2 LSTMs with 512 hidden units each. The text encoder has 1 LSTM layer with 512 hidden units. It should be noted that a BLSTM layer can also be adopted for modelling the text information. However, in the experimental results the performance was similar and a LSTM was chosen due to its smaller computational complexity.

The following data processing techniques are used for all datasets. Each frame is resized to 256×256 and cropped at a random position to a fixed size of 224×224. Random temporal frame sampling is used up to 80% of video length. Brightness, contrast, saturation and hue values of frames are randomly jittered up to 10%. The full architecture is trained with Adam optimizer with an initial learning rate $\lambda_0 = 5 * 10^{-5}$ and a batch size of 1 because of the computational cost and the fact that each video sequence consists of a different number of frames. Long videos are downsampled to a maximum length of 250 frames, if necessary. The learning rate is decreased by a factor of 0.5 when validation loss starts to plateau. The training process lasts 10 epochs for the CSL dataset and 20 epochs for the other two datasets. The proposed method is implemented in PyTorch and the experiments are conducted in a NVIDIA GeForce GTX-1080-Ti GPU.

TABLE 1. The effect of different joint latent space sizes reported in WER.

Latent space size	Dev	Test
256	26.2	26.3
512	25.7	26.0
1024	25.1	25.0
2048	24.9	24.9

C. RESULTS

To define the optimal hyperparameters of the network and study the effectiveness of each module, extensive experiments are conducted using the RWTH-Phoenix-Weather-2014 dataset, which is the most popular CSLR dataset.

Initially, the relationship between performance and dimensionality of the joint latent space is investigated. To this

FIGURE 4. Recognition results with multiple model configurations. The proposed method with the cross-modal alignment and the joint decoder gives the closest to ground truth CSLR results.

TABLE 2. Ablation study with different training stages and loss functions measured in WER.

Method	Dev	Test
Video Encoder	26.1	26.5
Proposed+L_{map}	25.1	25.0
Proposed+L_{joint}	**23.9**	**24.0**

end, experimental results with different latent space sizes are presented in Table 1, showing that by increasing the size of the joint latent space, the WER is further reduced. In the experiments, a latent space dimensionality of 1024, instead of 2048, is chosen since further increase leads to a decrease in WERs of only 0.2% and 0.1% on Dev set (i.e., validation set) and Test set, respectively, but in the cost of a higher number of parameters and slower training speed. Subsequently, to evaluate the effectiveness of loss functions L_{map} and L_{joint}, a series of experiments are conducted. At first, it is observed that when L_{map} is introduced in the early stages of training, performance drops by 5% in WER. The main reason is that the network produces unstable probability distributions. To this end, L_{map} is introduced at a later training stage when CTC has already converged. As shown in Table 2, the overall CSLR performance is improved by 1.5% when L_{map} is introduced at a later training stage. This means that by bringing closer video and text embeddings using L_{map}, the intra-gloss dependencies are effectively modelled decreasing the WER of the network, despite any additional errors that the text encoder may introduce.

After learning the joint latent space, the joint decoder is trained with L_{joint} using video and text latent representations. Finally, the outputs of the joint decoder when it is fed with the video latent representations are used for CSLR. In order to set optimal hyperparameters a and b, experimental results using different values for the two hyperparameters are conducted. Note that when $a = 0$ or $b = 0$ the joint decoder is trained only using the text or video embeddings, respectively. In the case of training the joint decoder using only text embeddings ($a = 0, b = 1$), the CSLR performance was not satisfactory with WER of only 87.0% on Dev set and 88.1% on Test set, respectively. However, training the joint decoder using only video embeddings ($a = 1, b = 0$), a CLSR performance with WER of 24.5% on Dev set and 24.4% on Test set, respectively is reached. After varying the hyperparameters a, b in the range [0, 1], the optimal values are set to 0.9 and 0.1, respectively, with WER of 23.9% on Dev set and 24.0% on Test set. Further increase in the contribution of the L_{LM} loss function (e.g., $b = 0.2$) was found to decrease the performance of the proposed method (with WER 24.9% and 24.8% on Test set for $a = 0.9$ and 0.8, respectively).

1) EVALUATION ON THE RWTH-PHOENIX-WEATHER-2014 DATASET

In this section, the proposed method is evaluated on the RWTH-Phoenix-Weather-2014 dataset, which contains recordings of weather forecasts. Videos are recorded with 9 different signers at a frame rate of 25 frames per second. The vocabulary size is 1295 and the dataset contains 5672, 540 and 629 videos for training, validation and testing respectively. In Table 3, the proposed approach is compared to several state-of-the-art approaches. It can be observed that the proposed method outperforms all state-of-the-art approaches, achieving a WER of 24.0% on the Test set. This indicates the advantage of exploring the correlation between sentence

TABLE 3. Comparison of CSLR approaches on the RWTH-Phoenix-Weather-2014 dataset measured in WER.

Method	Modality	Dev	Test
SubUNets [22]	full + hand	40.8	40.7
Staged-Opt [23]	hand	39.4	38.7
CNN-Hybrid [9]	hand	38.3	38.8
Dilated [26]	full	38.0	37.3
Align-iOpt [17]	full	37.1	36.7
DenseTCN [29]	full	35.9	36.5
DPD [28]	full	35.6	34.5
Re-Sign [20]	full	27.1	26.8
CNN-LSTM-HMM [16]	full+ hand	26.0	26.0
CNN-TEMP-RNN [14]	full	23.8	24.4
Proposed+L_{joint}	full	23.9	**24.0**

semantics and video. More specifically, the proposed method reduces WER by 0.4% with respect to CNN-TEMP-RNN [14], which justifies the importance of modelling intra-gloss dependencies. Furthermore, the proposed method reduces WER by 2% and 2.8% with respect to Re-Sign [20] and CNN-LSTM-HMM [16] methods, respectively, that use HMM frame-state alignments for network training. An example of alignment paths between video and text embeddings is shown in Figure 3. Each video embedding is aligned to its corresponding text embedding. The proposed method minimizes the distance of video and text embeddings using the alignment path. Qualitative recognition results with different model settings are shown in Figure 4. It can be observed that the video encoder without the latent space alignment is more prone to recognition errors. However, when introducing the joint latent space to combine and align video and text embeddings, the network yields better recognition results, while the use of the joint decoder leads to an even better performance.

2) EVALUATION ON THE RWTH-PHOENIX-WEATHER-2014T DATASET

RWTH-Phoenix-Weather-2014T [39] is an extended database of RWTH-Phoenix-Weather-2014, providing spoken language translations and gloss level annotations for German sign language videos of weather broadcasts. It contains 8257 videos from 9 different signers performing 1088 unique signs. The spoken language translations consist of 2887 different words. All videos are recorded with 25 frames per second and resolution of 210 × 260. The dataset is divided into three splits for training, validation and testing and there is no overlap with the previous version of the dataset in any split. As shown in Table 4, the proposed method achieves a WER of 24.1% on Dev set and 24.3% WER on Test set. The proposed method achieves a relative reduction in WER by 9.0% on the Test set compared to the CNN-LSTM-HMM method [16] that adopts HMM for temporal modelling and uses frame-level alignments.

3) EVALUATION ON THE CSL DATASET

The Chinese Sign Language (CSL) dataset [24] is a popular SLR dataset with a smaller vocabulary compared to RWTH-Phoenix-Weather-2014. Videos are recorded in a predefined laboratory environment with Chinese words widely used in daily conversations. It contains 100 sentences performed 5 times from 50 signers with 25000 videos in total. The signer independent split of train and test set in [32] is adopted, meaning that videos from 40 and 10 signers are used for training and testing, respectively. The dataset also provides an isolated version that contains 500 unique words. The proposed method is pretrained on the isolated version of the dataset achieving similar performance to other methods without time-consuming iterations. In Table 5, the proposed method is compared against several state-of-the-art approaches evaluated on the CSL dataset. The proposed method shows again superior performance achieving 2.4% WER, i.e., a 2.3% absolute reduction (95% relative) compared to the DPD method [28] that uses a deep 3D-CNN architecture.

TABLE 4. Evaluation comparison on the RWTH-Phoenix-Weather-2014T dataset measured in WER.

Method	Modality	Dev	Test
Re-Sign [20]	full	25.7	26.6
CNN-LSTM-HMM [16]	full	24.5	26.5
Proposed+L_{joint}	full	**24.1**	**24.3**

TABLE 5. Evaluation comparison on the CSL dataset measured in WER.

Method	Modality	Test
LS-HAN [24]	full	17.3
DenseTCN [29]	full	14.3
CTF [41]	full	11.2
HLSTM-att [42]	full	10.2
Align-iOpt [17]	full	6.1
DPD [28]	full	4.7
Proposed+L_{joint}	full	**2.4**

VI. CONCLUSION

In this paper, a novel deep learning method for continuous sign language recognition was introduced. In contrast to previous state-of-the-art approaches, the proposed method applies a cross-modal alignment between video and text embeddings to better model the intra-gloss dependencies in sign language recognition. Experimental results on the three most widely used CSLR datasets demonstrate the ability of the proposed method to provide highly accurate CSLR results.

Concerning future work, integrating other modalities, such as cropped hands, optical flow and skeletal keypoints can also be explored. The incorporation of additional modalities in a joint latent space could further enhance CSLR performance. Finally, it would be interesting to extend the proposed method for Sign Language Translation and exploit the relationship of video, sign language and spoken language simultaneously.

REFERENCES

[1] W. Sandler and D. Lillo-Martin, *Sign Language and Linguistic Universals*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[2] C. Wang, Z. Liu, and S.-C. Chan, "Superpixel-based hand gesture recognition with kinect depth camera," *IEEE Trans. Multimedia*, vol. 17, no. 1, pp. 29–39, Jan. 2015.

[3] D. Konstantinidis, K. Dimitropoulos, and P. Daras, "Sign language recognition based on hand and body skeletal data," in *Proc. 3DTV-Conf., True Vis.-Capture, Transmiss. Display 3D Video (3DTV-CON)*, Jun. 2018, pp. 1–4.

[4] H. Cooper, E.-J. Ong, N. Pugeault, and R. Bowden, "Sign language recognition using sub-units," *J. Mach. Learn. Res.*, vol. 13, pp. 2205–2231, Jul. 2012.

[5] D. Konstantinidis, K. Dimitropoulos, and P. Daras, "A deep learning approach for analyzing video and skeletal features in sign language recognition," in *Proc. IEEE Int. Conf. Imag. Syst. Techn. (IST)*, Oct. 2018, pp. 1–6.

[6] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," *Comput. Vis. Image Understand.*, vol. 141, pp. 108–125, Dec. 2015.

[7] U. Von Agris and K.-F. Kraiss, "Towards a video corpus for signer-independent continuous sign language recognition," in *Proc. Gesture Hum.-Comput. Interact. Simulation*, Lisbon, Portugal, May 2007, pp. 1–6.

[8] J. Zhang, W. Zhou, C. Xie, J. Pu, and H. Li, "Chinese sign language recognition with adaptive HMM," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2016, pp. 1–6.

[9] O. Koller, S. Zargaran, H. Ney, and R. Bowden, "Deep sign: Hybrid CNN-HMM for continuous sign language recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 1–12.

[10] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 1521–1527.

[11] D. Konstantinidis, K. Dimitropoulos, and P. Daras, "Skeleton-based action recognition based on deep learning and Grassmannian pyramids," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 2045–2049.

[12] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," *CoRR*, vol. 2, p. 3, 2017.

[13] F. Patrona, A. Chatzitofis, D. Zarpalas, and P. Daras, "Motion analysis: Action detection, recognition and evaluation based on motion capture data," *Pattern Recognit.*, vol. 76, pp. 612–622, Apr. 2018.

[14] R. Cui, H. Liu, and C. Zhang, "A deep neural framework for continuous sign language recognition by iterative training," *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1880–1891, Jul. 2019.

[15] O. Koller, S. Zargaran, H. Ney, and R. Bowden, "Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs," *Int. J. Comput. Vis.*, vol. 126, no. 12, pp. 1311–1325, Dec. 2018.

[16] O. Koller, C. Camgoz, H. Ney, and R. Bowden, "Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 15, 2019, doi: 10.1109/TPAMI.2019.2911077.

[17] J. Pu, W. Zhou, and H. Li, "Iterative alignment network for continuous sign language recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4165–4174.

[18] M. W. Kadous, "Machine recognition of auslan signs using powergloves: Towards large-lexicon recognition of sign language," in *Proc. Workshop Integr. Gesture Lang. Speech*, vol. 165, 1996, pp. 1–10.

[19] O. Koller, H. Ney, and R. Bowden, "Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3793–3802.

[20] O. Koller, S. Zargaran, and H. Ney, "Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4297–4305.

[21] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, 2006, pp. 369–376.

[22] N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden, "SubUNets: End-to-end hand shape and continuous sign language recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3075–3084.

[23] R. Cui, H. Liu, and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 7361–7369.

[24] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

[25] Z. Yang, Z. Shi, X. Shen, and Y.-W. Tai, "SF-Net: Structured feature network for continuous sign language recognition," 2019, *arXiv:1908.01341*. [Online]. Available: http://arxiv.org/abs/1908.01341

[26] J. Pu, W. Zhou, and H. Li, "Dilated convolutional network with iterative optimization for continuous sign language recognition," in *Proc. IJCAI*, vol. 3, 2018, p. 7.

[27] M. Cuturi and M. Blondel, "Soft-DTW: A differentiable loss function for time-series," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 894–903.

[28] H. Zhou, W. Zhou, and H. Li, "Dynamic pseudo label decoding for continuous sign language recognition," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2019, pp. 1282–1287.

[29] D. Guo, S. Wang, Q. Tian, and M. Wang, "Dense temporal convolution network for sign language translation," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 744–750.

[30] D. Guo, W. Zhou, A. Li, H. Li, and M. Wang, "Hierarchical recurrent deep fusion using adaptive clip summarization for sign language translation," *IEEE Trans. Image Process.*, vol. 29, pp. 1575–1590, 2020.

[31] D. Guo, S. Tang, and M. Wang, "Connectionist temporal modeling of video and language: A joint model for translation and sign labeling," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 751–757.

[32] J. Zhang, Y. Han, J. Tang, Q. Hu, and J. Jiang, "Semi-supervised image-to-video adaptation for video action recognition," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 960–973, Apr. 2017.

[33] F. Yu, X. Wu, J. Chen, and L. Duan, "Exploiting images for video recognition: Heterogeneous feature augmentation via symmetric adversarial learning," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5308–5321, Nov. 2019.

[34] Y. Liu, Z. Lu, J. Li, T. Yang, and C. Yao, "Deep image-to-video adaptation and fusion networks for action recognition," *IEEE Trans. Image Process.*, vol. 29, pp. 3168–3182, 2020.

[35] A. Mousa and B. Schuller, "Contextual bidirectional long short-term memory recurrent neural network language models: A generative approach to sentiment analysis," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 1023–1032.

[36] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc.*, 2010, pp. 1–24.

[37] S. S. Wilks, "The large-sample distribution of the likelihood ratio for testing composite hypotheses," *Ann. Math. Statist.*, vol. 9, no. 1, pp. 60–62, Mar. 1938.

[38] J. Heymann, K. C. Sim, and B. Li, "Improving CTC using stimulated learning for sequence modeling," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 5701–5705.

[39] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7784–7793.

[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[41] S. Wang, D. Guo, W.-G. Zhou, Z.-J. Zha, and M. Wang, "Connectionist temporal fusion for sign language translation," in *Proc. ACM Multimedia Conf. (MM)*, 2018, pp. 1483–1491.

[42] D. Guo, W. Zhou, H. Li, and M. Wang, "Hierarchical LSTM for sign language translation," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–8.

ILIAS PAPASTRATIS received the Diploma degree in electrical and computer engineering from the University of Patras, in 2018. He is currently a Research Assistant with the Visual Computing Lab (VCL), CERTH-ITI. His main research interests include the fields of computer vision, machine learning, and robotics.

KOSMAS DIMITROPOULOS received the B.Sc. degree in electrical and computer engineering from the Democritus University of Thrace, in 2001, and the Ph.D. degree in applied informatics from the University of Macedonia, Greece, in 2007. He is currently a Researcher with the Visual Computing Lab (VCL), CERTH-ITI, and an Adjunct Lecturer with the Postgraduate Programme of the University of Macedonia. His main research interests include 2-D/3-D data modelling, analysis and visualization, pattern recognition, and human–computer interaction. His involvement with those research areas has led to the coauthoring of more than 100 publications in refereed journals and international conferences. He has received as a coauthor in scientific articles: the IET ITS Premium Award (The IET Premium Awards 2012, London), the Euromed 2012 Best Full Paper Award, and the CONTACT/ECCV 2014 Best Student Paper Award. He has participated in several European and national research projects (as the Deputy Project Coordinator and the Quality Project Manager or the Work-Package Leader). He has served as a regular Reviewer for a number of international journals and conferences.

DIMITRIOS KONSTANTINIDIS received the B.Sc. degree in electrical and computer engineering from the Aristotle University of Thessaloniki (AUTH), in 2009, the Advanced master's degree in artificial intelligence from KU Leuven, in 2012, and the Ph.D. degree from the Imperial College of London with the topic of monitoring urban changes from satellite images, in 2017. He is currently a Postdoctoral Researcher with the Visual Computing Lab (VCL), CERTH-ITI. His main research interests include the fields of computer vision, image processing, machine/deep learning, and artificial intelligence.

PETROS DARAS (Senior Member, IEEE) received the Diploma degree in electrical and computer engineering and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Aristotle University of Thessaloniki, Greece, in 1999, 2002, and 2005, respectively. He is currently a Senior Researcher Grade B (Associate Professor) and the Chair of the Visual Computing Lab. His main research interests include visual content processing, multimedia indexing, search engines, recommendation algorithms, and relevance feedback. His involvement with those research areas has led to the coauthoring of more than 150 articles in refereed journals and international conferences. He has been involved in more than 20 projects, funded by the EC and the Greek Ministry of Research and Technology. Among them, he is the Technical Manager of the EC projects VICTORY, I-SEARCH, and ADVISE. He regularly acts as a Reviewer of the European Commission and the GSRT.