SubUNets: End-to-end Hand Shape and Continuous Sign Language Recognition

Necati Cihan Camgoz
University of Surrey
Guildford, UK
n.camgoz@surrey.ac.uk

Simon Hadfield
University of Surrey
Guildford, UK
s.hadfield@surrey.ac.uk

Oscar Koller
RWTH Aachen University, Germany
koller@cs.rwth-aachen.de

Richard Bowden
University of Surrey
Guildford, UK
r.bowden@surrey.ac.uk

Abstract

We propose a novel deep learning approach to solve simultaneous alignment and recognition problems (referred to as "Sequence-to-sequence" learning). We decompose the problem into a series of specialised expert systems referred to as SubUNets. The spatio-temporal relationships between these SubUNets are then modelled to solve the task, while remaining trainable end-to-end.

The approach mimics human learning and educational techniques, and has a number of significant advantages. SubUNets allow us to inject domain-specific expert knowledge into the system regarding suitable intermediate representations. They also allow us to implicitly perform transfer learning between different interrelated tasks, which also allows us to exploit a wider range of more varied data sources. In our experiments we demonstrate that each of these properties serves to significantly improve the performance of the overarching recognition system, by better constraining the learning problem.

The proposed techniques are demonstrated in the challenging domain of sign language recognition. We demonstrate state-of-the-art performance on hand-shape recognition (outperforming previous techniques by more than 30%). Furthermore, we are able to obtain comparable sign recognition rates to previous research, without the need for an alignment step to segment out the signs for recognition.

1. Introduction

Perception is a hierarchical process; our understanding of the world as a whole, is based on recognising different parts of the world and understanding their spatio-temporal interactions. As an example, for recognising human actions we not only recognise where the different body parts are located, but how they move relative to each other and in relation to surrounding objects. More generally, most spatio-temporal learning problems can be broken down into meaningful "subunit" problems. However, the subunits often have complex, unsynchronised, causal relationships, making it very challenging to model them jointly.

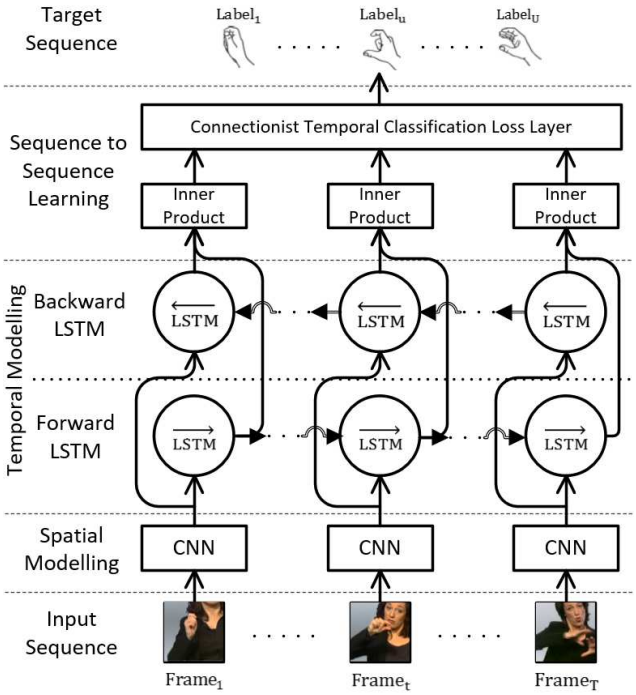

Figure 1. Overview of a SubUNet and its building blocks. In this example our input sequences are hand patch videos and target sequences are hand shape classes. Hand-Icons from [31].

Until recent years, most spatio-temporal computer vision techniques have extracted hand-crafted intermediate representations and then used classical temporal modelling approaches such as Hidden Markov Models and Conditional Random Fields [35]. The emergence of modern deep learning methods [30, 13, 34] has removed the need for such tailored representations and enabled systems to implicitly learn both the spatial and the temporal features. However, the disadvantage of deep learning is that it can be difficult to encode expert knowledge (such as suitable subunits or intermediate representations). This is especially true when dealing with sequence-to-sequence modelling problems, where the different subunits may not be synchronised with each other and can exhibit complex causal relationships.

In this paper, we present *SubUNets*[1], a novel deep learning architecture for sequence-to-sequence learning tasks, where the systems are expected to produce a sequence of outputs from a given video. Contrary to other video to text approaches, our method explicitly models the contextual subunits of the task while training the network for the main task. This allows us not only to encode expert knowledge about the properties of the task, but also to exploit a much wider range of annotation sources, and to exploit implicit transfer learning between tasks. We demonstrate this approach for the problem of Continuous Sign Language recognition, where the recognition systems are expected to detect and recognise the individual signs in a given video and produce a text translation. This problem is particularly well suited to our SubUNets approach as unlike spoken languages, sign is famously multi-channel. Information is carried in the hand shape, motions, body pose and even facial gestures. Additionally, there is a wealth of expert linguistic knowledge relating to sign language and the interactions between it's different modalities.

The contributions of this paper can be listed as:

- An end-to-end framework for explicitly modelling the subunits during sequence-to-sequence learning.
- The first end-to-end system for continuous sign language recognition alignment and recognition, based on explicit subunit modelling.
- A thorough comparison of different decoding schemes for networks using CTC loss.

The rest of the paper is organized as follows: In Section 2 we go over the related work on sequence-to-sequence modelling, and continuous sign language recognition. In Section 3 we depict SubUNets and go further into detail of its components. First we apply SubUNets to the problem of hand shape recognition in Section 4, achieving state-of-the-art performance without needing to realign the data. Then we describe our application of SubUNets to the challenge of Continuous Sign Language recognition in Section 5. Here we demonstrate how SubUNets can be combined to model the asynchronous relationship between different channels of information and that combining different loss layers allows expert knowledge to be incorporated which increases recognition performance. Finally, we conclude the paper in Section 6 by discussing our findings and the possible future work.

2. Related Work

Sequence-to-sequence learning methods can be grouped into two categories: Encoder-Decoder Networks [38] and approaches based on Connectionist Temporal Classification (CTC) [16].

Encoder-Decoder networks first emerged from the field of Neural Machine Translation (NMT) [32]. Kalchbrenner and Blunsom [24] proposed the first encoder-decoder network that uses a single Recurrent Neural Network for both encoding and decoding sequences. Following this Sutzkever et al. [38] and Cho et al. [8] proposed separating the encoding and decoding jobs into two separate RNNs. Although this approach improved their machine translation performance, there were still issues with modelling the long term dependencies between the input and output sequences. To overcome this problem Bahdanau et al. [4] proposed attention mechanisms that were able to learn where to focus on the input sequence depending on the output. These successes in NMT encouraged computer vision researchers to adopt encoder-decoder networks for applications such as image captioning [43], activity recognition [13] and lip-reading [9].

The second group of sequence-to-sequence learning approaches are based on CTC, proposed by Graves et al. [16]. This approach has been widely used in the fields of Speech Recognition [18, 2] and Hand Writing Recognition [17]. As CTC is an ideal method for tasks where the data is weakly labelled, computer vision researchers have also applied this sequence-to-sequence learning method to sentence-level lip reading [3] and action recognition [21].

In this paper, we demonstrate our proposed sequence-to-sequence learning techniques in the domain of continuous sign language recognition. This is due to its multi-channel nature [11], and the large amounts of expert linguistic knowledge available.

Until recently, most sign language recognition research was conducted on isolated sign samples [42, 5]. However, with the availability of large datasets, such as RWTH-PHOENIX-Weather-2014 [14], research interest has started to shift towards continuous sign language recognition. As frame level annotations are hard to come by in continuous datasets, most of the work to date required an alignment step to localize individual signs in videos [10]. The work that is most relevant to this paper is by Koller et al. [27] which combines deep-representations with traditional HMM based temporal modelling.

3. SubUNets

In this section we present a novel deep learning architecture for generic video to sequence learning problems, employing smaller specialized sub-networks. This approach forces the network to explicitly model domain specific expert knowledge, better constraining the overarching recognition problem. We refer to these smaller specialized networks as SubUNets, as they are trained to model subunits of a given task.

Each SubUNet consists of three tiers of neural network. Firstly, Convolutional Neural Networks (CNNs) take images as inputs and extract spatial features. Secondly, Bidirectional Long Short Term Memory Layers (BLSTM) temporally model the spatial features extracted by the CNNs. Finally a Connectionist Temporal Classification (CTC) Loss Layer allows the networks to be trained with different length videos

[1] Not to be confused with U-Nets [36]

and label sequences. We depict a sample SubUNet architecture that learns hand shapes from cropped hand images in Figure 1. In the remainder of this section, we will provide further details on each tier of the SubUNets, and describe how to train them in an end-to-end manner.

3.1. Spatial Feature Extraction: Convolutional Neural Networks

In SubUNets we employ 2D CNNs for learning the spatial feature representations. Given an input image I with c channels, the 2D convolution layers extract the feature map F by convolving the image with the weights w as in

$$F(x,y) = \sum_{c} \sum_{\delta x, \delta y} I(x+\delta x, y+\delta y, c) \times w(\delta x, \delta y) + b \quad (1)$$

where x and y represent the pixel coordinates of the image I and b is the bias term. The spatial neighbourhood that δ_x and δ_y are drawn from is defined by the kernel size of the convolution layer.

Although the SubUNets approach can exploit any CNN architecture for spatial modelling, our experiments use CaffeNet due to its low memory consumption (see Section 3.4 for further details). CaffeNet is a variant of AlexNet [30] that has five convolutional and three fully connected layers. We discard the last fully connected layer and use the weights that were pre-trained on ImageNet [12].

3.2. Temporal Modelling: Bidirectional LSTMs

Two dimensional convolutional neural networks have achieved state-of-the-art performance for many spatial recognition tasks [39]. However they do not have the ability to model temporal transitions of a video sequence. The spatio-temporal convolutional networks [41] can theoretically model temporal change in the spatial domain but their ability to represent state transitions is limited. Instead, we model the temporal aspects of our input sequences using Recurrent Neural Networks (RNNs).

One of the main difficulties when training RNNs is the vanishing gradient problem. The error generated from each time step (and it's associated gradients) diminishes during the course of the sequence [33]. In order to preserve the long term dependencies from the effects of vanishing gradients Hochreiter et al. [20] proposed Long Short Term Memory (LSTM) units.

LSTMs try to overcome the vanishing gradient problem by proposing a cell state in addition to the hidden state that classic RNNs use. Furthermore, it has specialized, input, forget and update gates that minimize the diminishing effects of long term dependencies.

An LSTM unit takes as input, the cell state, C_{t-1} and hidden state, h_{t-1} from the previous time step along side the spatial data F_t at the current time step. It then computes the input gate i_t, forget gate f_t and the update gate $\tilde{C}_t$ as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, F_t] + b_f) \quad (2)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, F_t] + b_i) \quad (3)$$

$$\tilde{C}_t = tanh(W_c \cdot [h_{t-1}, F_t] + b_c) \quad (4)$$

Using the calculated gate values, the LSTM unit calculates the output o_t, cell state C_t and the hidden state h_t values to pass to the next time step as:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (5)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, F_t] + b_o) \quad (6)$$

$$h_t = o_t * tanh(C_t) \quad (7)$$

From these equations, it is obvious that an LSTM produces the output at a time step t using the current spatial information F_t and the information leading up to this point, encoded in the hidden state h_{t-1}. Thus, any time step following t has no effect on the output of the LSTM at time step t. Although, this gives LSTM the ability to operate in real-time, there is useful information in the following frames that is not being used to constrain the current frame's prediction.

Therefore, we deploy BLSTMs as our temporal modelling layer. A BLSTM contains two LSTM layers operating in opposite directions along the time domain (See Figure 1). The outputs of the two LSTMs are then concatenated before being fed deeper into the network. The main idea of the BLSTM is to provide knowledge about the full sequence during prediction. The output of the BLSTM at time t is based on both of the hidden states encoding $F_{1:t-1}$ and $F_{T:t+1}$. In our SubUNets each BLSTM layer has 2048 units, 1024 units in each direction. Although, the use of BLSTM layers limits the real-time capabilities, on-line prediction is still achievable with a sliding window approach.

3.3. Sequence-to-Sequence Learning: Connectionist Temporal Classification

When trained with Cross Entropy Loss [15], both the classic feed-forward and recurrent architectures require a label for each sample or time step. However, nearly all sequence-to-sequence problems have different length input and target sequences. One way to overcome this problem might be to segment the input sequences and assign a corresponding label to each time step. However, this level of annotation for every sub-unit on large datasets would be impractical. Furthermore, segmenting an input sequence in this manner often introduces label ambiguity, as the system is forced to predict the same class across the start, middle and end of a segment. Therefore, an additional structure is required to effectively train sequence-to-sequence models using feed-forward and recurrent neural networks.

Connectionist Temporal Classification (CTC), a loss layer proposed by Graves et al. [16], is one of the most popular

approaches to training sequence-to-sequence models. When using generic loss functions to train a network with L target labels, (vocabulary), we structure our architecture to have $|L|$ outputs, each one corresponding to one of the labels. This allows our network to produce posteriors over each label for every time step. CTC introduces a blank label $\_$ and creates an extended vocabulary L', where $L' = L \cup \{\_\}$, and restructures the network by adding another output unit corresponding to the blank label. The blank label accounts for silence and transitions that may exist between target labels in the sequence, removing the need for per frame annotation.

Although the blank label solves some of the problems, the network still has to learn which parts of the input sequence s^T, with T time steps, corresponds to silence and transitions. To solve this, the CTC defines a mapping function $B : L'^T \rightarrow L^U$ (where $U \leq T$) between extended vocabulary sequences $\pi \in L'^T$ and label sequences $l \in L^U$ by collapsing repetitions and removing the blank labels in π. Given an input sequence s, the probability of observing a label sequence l is computed by marginalising over all extended vocabulary sequences that would give rise to l. In other words, if we define an inverse mapping function B^{-1} which produces every possible extended vocabulary sequence π corresponding to label sequence l, then the probability of l given an input sequence s is:

$$p(l|s) = \sum_{\pi \in B^{-1}(l)} p(\pi|s) \tag{8}$$

However, as the length of label sequence increases, the number of corresponding extended vocabulary sequences π expands drastically. To overcome this, CTC uses dynamic programming to efficiently calculate the loss and its gradient.

3.4. Implementation Details and Training

The proposed architecture is implemented using the BLVC Caffe [23] framework and the CTC implementation of ChWick[2]. Training was done on a single Titan X GPU with Maxwell chip architecture and 12 GB VRAM. The code of our paper is publicly available[3].

While choosing our SubUNet layer architectures, memory usage was of particular importance, so that the combined SubUNets would fit into a single GPU. This is exacerbated by the need for CTC to simultaneously have the posteriors from all frames of a sequence in order to calculate the loss, meaning entire sequences must be processed as a whole. Therefore, a set of preliminary experiments was conducted using a dummy SubUNet (One layer of BLSTM with 100 units in each direction) with all well known CNN architectures, to check the practical limitations of memory use on a single Titan X GPU. In these experiments images were resized to the input size of each network, i.e. 224×224 or 227×227.

CNN Architecture	#frames	Memory (MB)
ResNet-50 [19]	35	12201
GoogLeNet [40]	160	12081
VGG-16 [37]	175	12025
SqueezeNet v1.1 [22]	320	12109
AlexNet [30]	1080	12111
VGG-F [7]	1340	12131
CaffeNet [23]	**1450**	**12104**

Table 1. Most Common Architectures and the maximum number of frames we can load to them on a single GPU.

As can be seen in Table 1, the dummy SubUNet using CaffeNet was able to support batches containing significantly longer sequences. Therefore, CaffeNet is used as the spatial encoding layer for all remaining experiments. To be able to train variable length input and output sequences in a single batch and to avoid the memory allocation overhead we resized all frames to 227×227 and padded all input sequences to 300 frames.

All of our networks were trained using the Adam Optimization Algorithm [25] with a learning rate of 10^{-4} and the default parameters: $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}$.

4. Hand SubUNet: End-to-End Hand Shape Recognition and Alignment

To demonstrate the power of the proposed SubUNet approach we focus on the challenging task of Continuous Sign Language Recognition. One of the primary information carrying modalities in sign language is the hand shape. Therefore, as our first SubUNet, we train a network that learns to simultaneously recognize and time-align hand shape sequences from videos of cropped hand patches.

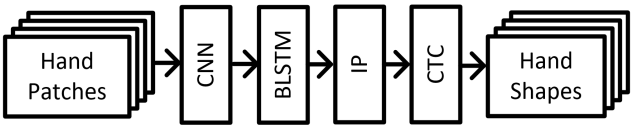

Figure 2. Hand SubUNet: End-to-end Hand Shape Recognition network from sequences.

Hand shape recognition is a challenging task in its own right, due to a hands' highly articulated nature. For instance, the same hand shapes (with the same linguistic meaning) but viewed from different directions, results in drastically different appearances in the image due to self occlusion. To be able to generalize across this variation, without over-fitting, requires vast amounts of training data.

We use the One-Million Hands [27] dataset for training the Hand SubUNet. The dataset consists of cropped hand images

[2]https://github.com/BVLC/caffe/pull/4681/
[3]https://github.com/neccam/SubUNets

collated from publicly available datasets, including Danish [29], New Zealand [31] and German (RWTH-PHOENIX-Weather-2014 [14] sign languages. It has over 1.2 million hand images, from which 1 million images were labelled with one of 60 hand shape classes. The dataset contains 23 different signers, which helps our network to generalize over different users as well as language. The statistics of the dataset can be seen in Table 2. The majority of the dataset comes from the Training set of RWTH-PHOENIX-Weather-2014, a continuous sign language dataset which will be used in our continuous sign language recognition experiments in Section 5.

	Danish	NZ	DGS	Total
duration [min]	98	192	532	882
#frames	145,720	288,593	799,006	1,233,319
#labelled frames	65,088	153,298	786,750	1,005,136
#sequences	2,149	4,155	5,672	11,976
#signs	2,149	4,155	65,227	69,382
#signers	6	8	9	23

Table 2. Statistics of the One Million Hands dataset which contains cropped hand patches from existing Danish, New Zealand ('NZ') and German('DGS' - RWTH-PHOENIX-Weather-2014) sign language datasets. See [27] for more details.

The One-Million Hands dataset provides frame-level annotation for these sequences. However, as we are focussing on the more challenging sequence-to-sequence problem, we remove repetitions of the frame-level annotations to form our target sequence of hand shapes.

For our network architecture, we used the first 7 layers (5 Convolution, 2 Fully Connected Layers) of the CaffeNet, followed by a single layer of BLSTM with 1024 units in each direction. As the size of our vocabulary for this SubUNet is 61 (60 hand shapes and the blank CTC label) we follow the BLSTM layer with an inner product layer of 61 units. At the end, a CTC Loss Layer is deployed to be able to learn both alignment and recognition in a sequence-to-sequence manner. A simplified visualization of the network can be seen in Figure 2, while Figure 1 illustrates the network after being unrolled in time.

The network was trained for 5000 iterations with a mini-batch size of 90 sequences, using the Adam Optimizer as described in Section 3.4. Optimization is terminated when the training loss has converged.

To evaluate the performance of our network we used the 3361 manually annotated hand images provided by [27], which are from the Development set of the RWTH-PHOENIX-Weather-2014 dataset. Again, because we are interested in the more challenging alignment & recognition problem, we run the system on the full (unseen) test sequences from which these images were taken. We then extract and evaluate the estimated hand shapes for the subset of frames which have ground truth.

As shown in Table 3, our Hand SubUNet surpasses the hand shape recognition performance of the state-of-the-art CNN-based method proposed by Koller et al. [27], by a margin of 18% Top-1 accuracy, which is a relative improvement of 30%. Koller et al. [27] iteratively realigned and retrained his network whereas the SubUNet architecture automatically overcomes the frame alignment issue. These experiments show us that SubUNets are able to learn both the alignment and the recognition jointly from sequences in an end-to-end fashion, without requiring any other alignment procedure.

We will now demonstrate the power of SubUNet to the sequence-to-sequence learning problem by applying it to end-to-end, multi-channel, continuous sign language recognition.

	Top-1	Top-3	Top-5	Top-10
Koller et al. [27]	62.8	–	85.6	–
Hand SubUNet	**80.3**	90.6	**93.9**	96.9

Table 3. Hand SubUNet's hand shape recognition results on the One-Million Hands dataset.

5. Sign SubUNets: End-to-End Continuous Sign Language Recognition

Compared to their spoken counter parts, Sign Languages are multi-channel languages. Its users convey information using a combination of hand and face gestures, hand movements, upper body pose and facial expressions. The nature of sign languages, makes it an ideal target application for the SubUNets-based approach.

Due to the difficulty in collecting annotations, most of the sign language recognition datasets that have been developed, consist of isolated sign videos [42], [6]. Although these datasets are suitable for isolated recognition [5], they do not support the ultimate aim of sign language recognition research: the translation of sign language utterances to their spoken languages equivalents. Indeed, training a sign language recognition system using these isolated datasets is equivalent to training a machine translation systems using a dictionary alone. Such a system would be unable to learn the higher order sentence-level relationship between words or signs[4].

To be able to train a sentence-level sign language recognition system, we used the RWTH-PHOENIX-Weather-2014 dataset, a DGS (German Sign Language) dataset that consists of Weather Forecast Footage. The dataset contains both the full frames and the cropped hands of the signers. This multi-channel data is ideal to test our SubUNet network. For both information channels there are 6841 sequences containing a total of 77,321 words. The statistics of the dataset can be seen in Table 4.

To be able to assess the benefits of SubUNets for Continuous Sign Language Recognition, we conducted experiments using a variety architectures.

[4]It is important to note that sign languages do not contain a direct equivalent to sentences, the term is used here to clarify the concept to the reader and refers to a meaningful phrase which consists of a sequence of continuous signs.

	Train	Dev	Test
#frames	799,006	75,186	89,472
#sequences	5,672	540	629
#words	65,227	5,564	6,530
#vocabulary	1,231	461	497
#signers	9	9	9

Table 4. Summary of RWTH-PHOENIX-Weather-2014 dataset.

All of the sign language recognition networks were trained for 6000 iterations using the Adam Optimizer as described in Section 3.4 with a mini-batch size of 60 sequences. Its performance on the development set was evaluated at every epoch, which is 96 iterations. If the training loss has not converged after 60 epochs, we restart the training using the best performing iteration with a lower learning rate and train until the training loss convergence.

To evaluate the performance of a model, we fed the development and test set sequences through the network and extracted posterior probabilities for each word and the blank CTC label. These posteriors are then passed through TensorFlow's implementation of CTC beam decoder [1] with a beam width of 100 to obtain the final sequence predictions. To facilitate comparison with previous publications we measure word error rate (WER) as:

$$\text{WER} = \frac{\#\text{deletions} + \#\text{insertions} + \#\text{substitutions}}{\#\text{number of reference observations}} \quad (9)$$

5.1. Word SubUNet: End-to-End Continuous Sign Language Recognition Single Channel

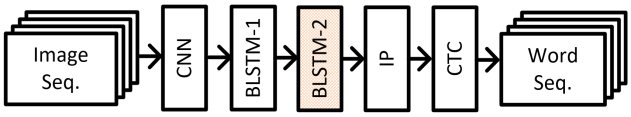

Figure 3. Word SubUNet: End-to-end Sentence-level Continuous Sign Language Recognition Network.

As our first sentence-level architecture we train SubUNets that learn the mapping between the given input sequence and the word sequences. As depicted in Figure 3, this network is similar to the proposed Hand SubUNet. However, as words have a more abstract relationship to the images than the observable hand shapes, we employ a deeper BLSTM structure (adding BLSTM-2) to help the network model the temporal relationships within the input sequence.

As can be seen in Table 5, having two layers in our Word SubUNets improves our sentence-level recognition performance. However, in order to combine multiple SubUNets for different information channels (in subsequent experiments), adding further layers is infeasible due to GPU memory limitations.

In theory, full frame sequences should provide all necessary channels of information for a sign. In other words hand shape are by definition contained in the full body frame and the network should be able to find and use this information. However, the problem is under-constrained, and it is unclear what information the network will actually use to predict word sequences. Due to the network's resolution, the most likely candidates are hand shape, arm motions and upper body pose. To see how much the network benefits from having the additional information in the full frame we train another SubUNet using the same network architecture and parameters but this time using only the cropped hands as the input sequences.

	Dev		Test	
Full Frames	del/ins	WER	del/ins	WER
Single Layer	19.9/5.2	44.5	18.9/5.6	43.8
Two Layers	20.6/3.2	**43.9**	19.8/3.2	**43.1**

Table 5. Evaluation of having a deeper network.

	Dev		Test	
Trained on	del/ins	WER	del/ins	WER
Hand Patches	24.3/2.8	45.8	23.4/2.5	44.5
Full Frames	20.6/3.2	**43.9**	19.8/3.2	**43.1**

Table 6. Evaluation of training Word SubUNets on different information channels.

As can be seen in Table 6, training a Word SubUNet with the hand patches worsens our performance by 2% WER. This means the network trained on the full frames does make use of the additional information contained in the full frames. However, it is still unclear how redundant the information is. Do the full frames contain all the information from the hand patches plus a small amount of novel information? Or is the additional context of motion in the full frame experiment compensating for the loss of hand shape information? To answer these questions, we propose combining networks that model hand shape (Hand Patches Word-SubUNets and Hand-SubUNets) with the Full Frame Word-SubUNet to see if the sentence-level recognition performance benefits from both sources of information.

5.2. Combining SubUNets: End-to-End Continuous Sign Language Recognition from Multiple Channels

So far we have trained three SubUNets: A *hand* SubUNet that predicts hand shape sequences from hand patches and two *word* SubUNets which perform sentence-level sign language recognition from either *full frame* or *hand patch* sequences. Although our experiments have demonstrated that a *word* level SubUNet is able to make use of the additional information from the Full Frames, it is unknown how novel this information is compared to the hand shape network. Therefore, we combine pre-trained networks, trained at the word level, for both Hand SubUNets and and Full Frame SubUNets to create a larger network that takes advantage of both sources of information.

Due to the asynchronous nature of the sign language modalities, we put the combined information of the Full Frame Word SubUNet and Hand SubUNet through an additional BLSTM layer. This models the temporal relationship between the modalities.

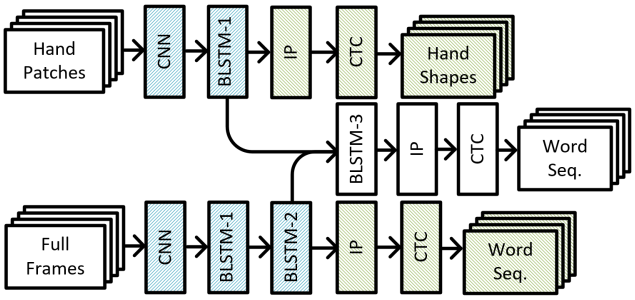

Figure 4. Combination of Hand SubUNets and Full Frame Word SubUNets. Blue and Green Blocks represent the weights that are going to be fixed and the weights that are going to omitted in the fixed setup, respectively.

As most datasets don't have annotations for both the hand shape and signs (making it impossible to jointly train all streams), we investigate the effects of fixing the weights for pre-trained SubUNets. By doing so we hope to determine how much the SubUNets benefits from tuning themselves to the new compound architecture. Therefore we train two networks. In the first experiment ("Fixed"), we pre-train the two SubUNets depicted in Figure 4. Combining the two networks at their final BLSTM layers into a 3rd BLSTM, IP and CTC layer and therefore maintaining 3 loss layers. In the figure, blue blocks are pre-trained and fixed, green blocks are removed, while white block are trained for the task. In the second variant ("Not Fixed"), all weights are trained using the gradients produced by all three loss layers.

	Dev		Test	
Combined SubUNet	del/ins	WER	del/ins	WER
Fixed	24.4/2.2	44.4	23.6/2.2	44.2
Not Fixed	19.6/2.7	**43.1**	18.7/2.9	**42.1**

Table 7. Evaluation of fixing SubUNets weights or allowing them to train end-to-end.

This experiment provides two very important insights into combining SubUNets. Firstly, as shown in Table 7, allowing the SubUNets to tune themselves to the new network structure by training end-to-end yields significantly improved results. Secondly, and more interestingly, the combination of the different SubUNet modalities outperforms all previous experiments using isolated SubUNets. This reinforces the idea that guided subunit learning is extremely valuable in sequence-to-sequence recognition.

For our final experiment, we evaluate how much the expert knowledge embedded within the SubUNets is contributing to the system. The inspiration behind this expert knowledge,

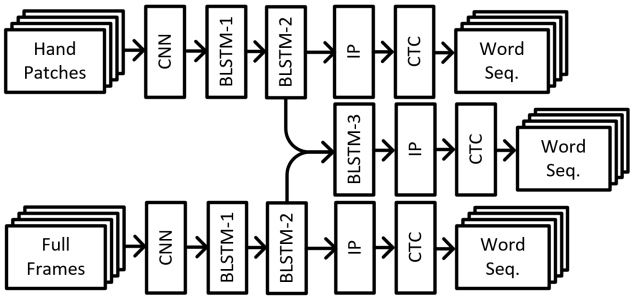

Figure 5. Combining Word SubUNets that model Full Frame and Hand Patches to sentence-level sign language.

comes from how humans teach and learn similar representations. For example, both linguists, and students learning sign, would classify the hand shape related to a sign as being a distinct but related entity to the motion of that that sign.

We investigate this using the best performing network from the previous section (the "Not Fixed" combination of Hand SubUNet and Full Frame Word SubUNet from Figure 4). As we trained this network, the additional supervisory information of the Hand SubUNet forces the hand patches stream to learn hand shapes explicitly, mimicking its human counterpart, which we named Expert SubUNets.

For comparison we instead leave the network free to train but replace the hand shape CTC with another Word CTC (as in Figure 5), which we named Generic SubUNets. In this case the network receives the same level of supervision, and one could argue that the supervision is more specific to the task at hand. The network is given the freedom to learn any intermediate representation it wishes in order to solve the overarching problem.

	Dev		Test	
	del/ins	WER	del/ins	WER
Generic SubUNets	27.1/1.6	43.0	26.8/1.5	42.6
Expert SubUNets	**19.6/2.7**	**43.1**	**18.7/2.9**	**42.1**

Table 8. Comparison of Generic and Expert SubUNet systems with other approaches.

However, as Table 8 shows, forcing networks to learn expert knowledge representations actually results in *better* performance on sentence-level sign language recognition. Although, both of the networks have a similar WER on the development set, the architecture that explicitly learns the intermediate hand shape representations performs better on the test set. Furthermore, the number of deletion and insertions is much more balanced for the network that mimics human learning. This implies that it is also performing better at the alignment task. Therefore, in light of these experiments we can conclude that training deep neural networks using SubUNets that explicitly model expert knowledge results in better constrained and more general solutions.

5.3. Decoding of networks trained with CTC loss

In this final section we explore the effects of different decoding and post-processing techniques during sequence-to-sequence prediction. Previously, the CTC outputs were decoded for prediction by performing a beam search on the sum of the probabilities over all possible alignment paths (as proposed by [16]). In other words it attempts to choose the best label for each frame, marginalised over all previous and future labellings. We refer to this approach as 'Full Sum' decoding. We contrast this against a greedy 'Viterbi' decoding which only considers the maximum path. Table 9 compares the two decoding strategies, showing that Full Sum decoding outperforms its counterpart by 0.5% points on dev and 0.6% points on the test set. However, this gain comes at a price of much higher computational complexity.

Decoding	Dev		Test	
	del/ins	WER	del/ins	WER
Viterbi	20.4/2.9	43.6	19.4/2.9	42.7
Full Sum	**19.6/2.7**	**43.1**	**18.7/2.9**	**42.1**

Table 9. Impact of the Full Sum and Viterbi decoding variants.

The significant impact of this change in post-processing raises an interesting question: Are there more advanced post-processing techniques that could further improve the performance of the system? We therefore apply an additional pre-learnt language model during decoding, similar to that proposed by [28].

Figure 6 shows the difference between the three tested decoding schemes. First the CTC topology, which binds a class posterior state to a tied blank state. Second, an intermediate topology referred to as LM, where the CTC style segments are joined with optional intermediate silence states that do not belong to the classes, but share the same distribution as the blank states. Finally, a HMM inspired topology is depicted, where two class states are bound together (sharing the same probability distribution) with optional tied silence states in between.

Table 10 summarises the decoding results employing the different topologies. We see that the HMM topology with a language model and the intermediate silence state outperforms the standard CTC topology by nearly 3% on development set. The table also shows that our proposed technique performs comparably to previous state-of-the-art research on this dataset, with the significant advantages that there is no need for a separate alignment step, the system can be trained end-to-end, and it extends easily to additional SubUNets.

6. Conclusion and Discussion

In this paper we have proposed SubUNets, a novel deep learning architecture that learns intermediate representations to guide the learning procedure. We have applied the proposed method to the challenging task of Continuous Sign Language Recognition.

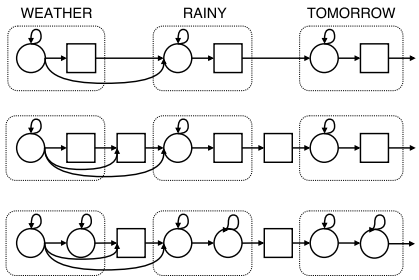

Figure 6. Showing three different model topologies used for decoding. Skip paths have only been illustrated on the first model segments. Round circles refer to single (class) states, whereas squares mean tied states (blank or silence). The top row shows the standard CTC topology. The middle row shows the CTC topology with optional silence insertions in between class symbols. The last row shows a HMM topology, where the same class distribution is shared across two states in a segment and optional silence can be inserted in between class symbols.

Model Structure	Dev		Test	
	del/ins	WER	del/ins	WER
CTC	19.6/2.7	43.1	18.7/2.9	42.1
LM	12.3/6.2	42.5	15.2/4.5	42.2
HMM-LM	**14.6/4.0**	**40.8**	**14.3/4.0**	**40.7**
[26]	23.6/4.0	57.3	23.1/4.4	55.6
[27]	16.3/4.6	47.1	15.2/4.6	45.1

Table 10. Evaluation of different decoding schemes and comparison with previous research.

As hands are one of the most informative channels of a sign we have trained a hand shape recognition network using the SubUNet architecture, that learns to predict hand shape sequences from a video. We trained and evaluated our hand shape recognizing SubUNet on the One Million Hands dataset [27] and reported state-of-the-art frame level accuracy (Top 1: 80.3%, Top 5: 93.9%), improving on previous research by around 30%.

Our experiments on Continuous Sign Language recognition show that having SubUNets that learn intermediate representations helps the network generalize better. Moreover we have thoroughly evaluated the effects of different decoding schemes and have seen the benefits of extra post processing, reporting competitive results to the state-of-the-art, without the need for an explicit segmentation of the signs.

As future work, it would be interesting to investigate hierarchical SubUNets, where each expert system is comprised of lower level expert systems.

Acknowledgement

This work was funded by the SNSF Sinergia project Scalable Multimodal Sign Language Technology for Sign Language Learning and Assessment (SMILE)" grant agreement number CRSII2_160811. We would also like to thank NVIDIA for their GPU grant.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. *arXiv:1603.04467*, 2016.

[2] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, et al. Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin. In *International Conference on Machine Learning (ICML)*, 2016.

[3] Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas. Lipnet: Sentence-level lipreading. *arXiv:1611.01599*, 2016.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations (ICLR)*, 2015.

[5] N. C. Camgöz, A. A. Kındıroğlu, and L. Akarun. Sign Language Recognition for Assisting the Deaf in Hospitals. In *International Workshop on Human Behavior Understanding*, 2016.

[6] N. C. Camgoz, A. A. Kindiroglu, S. Karabuklu, M. Kelepir, A. S. Ozsoy, and L. Akarun. BosphorusSign: A Turkish Sign Language Recognition Corpus in Health and Finance Domains. In *International Conference on Language Resources and Evaluation (LREC)*, 2016.

[7] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the Devil in the Details: Delving Deep into Convolutional Nets. In *British Machine Vision Conference (BMVC)*, 2014.

[8] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio. On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. *Syntax, Semantics and Structure in Statistical Translation*, 2014.

[9] J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman. Lip Reading Sentences in the Wild. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] H. Cooper and R. Bowden. Learning Signs from Subtitles: A Weakly Supervised Approach to Sign Language Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[11] H. Cooper, B. Holt, and R. Bowden. Sign Language Recognition. In *Visual Analysis of Humans*. 2011.

[12] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[13] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[14] J. Forster, C. Schmidt, O. Koller, M. Bellgardt, and H. Ney. Extensions of the Sign Language Recognition and Translation Corpus RWTH-PHOENIX-Weather. In *International Conference on Language Resources and Evaluation (LREC)*, 2014.

[15] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[16] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *International Conference on Machine Learning (ICML)*, 2006.

[17] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber. A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2009.

[18] A. Graves, A. Mohamed, and G. Hinton. Speech Recognition with Deep Recurrent Neural Networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[20] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 1997.

[21] D.-A. Huang, L. Fei-Fei, and J. C. Niebles. Connectionist Temporal Modeling for Weakly Supervised Action Labeling. In *European Conference on Computer Vision (ECCV)*, 2016.

[22] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer. SqueezeNet: AlexNet-level Accuracy with 50x Fewer Parameters and <1MB Model Size. *arXiv:1602.07360*, 2016.

[23] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. In *ACM International Conference on Multimedia*, 2014.

[24] N. Kalchbrenner and P. Blunsom. Recurrent Continuous Translation Models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2013.

[25] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2014.

[26] O. Koller, J. Forster, and H. Ney. Continuous Sign Language Recognition: Towards Large Vocabulary Statistical Recognition Systems Handling Multiple Signers. *Computer Vision and Image Understanding (CVIU)*, 2015.

[27] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[28] O. Koller, S. Zargaran, H. Ney, and R. Bowden. Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition. In *British Machine Vision Conference (BMVC)*, 2016.

[29] J. H. Kristoffersen, T. Troelsgard, A. S. Hardell, B. Hardell, J. B. Niemela, J. Sandholt, and M. Toft. Ordbog over Dansk Tegnsprog. 2008–2014.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*. 2012.

[31] D. McKee, R. McKee, S. P. Alexander, and L. Pivac. The Online Dictionary of New Zealand Sign Language. 2015.

[32] G. Neubig. Neural Machine Translation and Sequence-to-sequence Models: A Tutorial. *arXiv:1703.01619*, 2017.

[33] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training Recurrent Neural Networks. In *International Conference on Machine Learning (ICML)*, 2013.

[34] L. Pigou, A. Van Den Oord, S. Dieleman, M. Van Herreweghe, and J. Dambre. Beyond Temporal Pooling: Recurrence and Temporal Convolutions for Gesture Recognition in Video. *International Journal of Computer Vision (IJCV)*, 2015.

[35] R. Poppe. A Survey on Vision-based Human Action Recognition. *Image and Vision Computing*, 2010.

[36] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

[37] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *International Conference on Learning Representations (ICLR)*, 2015.

[38] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[39] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2017.

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[41] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning Spatiotemporal Features with 3D Convolutional Networks. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[42] H. Wang, X. Chai, X. Hong, G. Zhao, and X. Chen. Isolated Sign Language Recognition with Grassmann Covariance Matrices. *ACM Transactions on Accessible Computing*, 2016.

[43] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *International Conference on Machine Learning (ICML)*, 2015.