Phonologically-meaningful Subunits for Deep Learning-based Sign Language Recognition

Mark Borg[0000−0001−9167−5027] and Kenneth P. Camilleri[0000−0003−0436−6408]

Dept. of Systems and Control Engineering, University of Malta, Malta
{mark.j.borg,kenneth.camilleri}@um.edu.mt

Abstract. The large majority of sign language recognition systems based on deep learning adopt a *word model* approach. Here we present a system that works with subunits, rather than word models. We propose a pipelined approach to deep learning that uses a factorisation algorithm to derive hand motion features, embedded within a low-rank trajectory space. Recurrent neural networks are then trained on these embedded features for subunit recognition, followed by a second-stage neural network for sign recognition. Our evaluation shows that our proposed solution compares well in accuracy against the state of the art, providing added benefits of better interpretability and phonologically-meaningful subunits that can operate across different signers and sign languages.

Keywords: Sign language recognition, gesture recognition, deep learning, assistive technology, human computer interaction.

1 Introduction

It is estimated that over 5% (466 million people) of the global population are deaf. When deaf people are surrounded by hearing people, including their immediate families, with little to no knowledge of sign language, it creates a situation where the deaf struggle to establish effective communication. This creates communication barriers, that can lead to problems in early language acquisition, the development of language skills, and also affecting cognitive development in the long run [49, 55].

Automated sign language recognition (ASLR) is one tool, from a set of sign language related technologies [28, 18], that can provide some help in 'bridging the gap' between the deaf and the hearing world. ASLR converts a sign language utterance performed by a deaf person, into its textual representation or spoken language equivalent. The current state of the art in the field of ASLR is still limited in terms of accuracy and sign language support [15, 13, 60]. This is mainly due to the challenging nature of signing, both in terms of vision-based perception, as well as due to the complexities of the sign languages themselves. Compared to speech recognition, the current state of the art in terms of word error rate (WER) for ASLR is $\approx$26% [34], while for speech, the best WER stands at $\approx$4% [38] – this equates to roughly 1 in 4 mis-classified signs, versus 1 in 25 words

for speech [1]. So having a fully-fledged, robust, and generic sign language-to-text/speech conversion tool, is still at the moment not feasible. Notwithstanding this, advancements in ASLR are progressing at a steady and encouraging pace, especially in recent years with the advent of deep learning (DL) [60].

2 Related Work

From the literature we can identify two broad approaches to performing ASLR. The first approach consists of a traditional pipeline of computer vision components, followed by a sign recognition module. The vision components typically perform body-part detection and tracking, such as the signer's hands and face [9, 1, 58, 44, 12]. Depending upon the number and choice of modalities, a set of hand-crafted features are extracted from the video stream, and these are then fed to the recognition module, typically based on hidden Markov models (HMMs) [31, 48].

A common trend for these systems is the adoption of a *sub-word model* for sign recognition, in which signs are constructed from a concatenation of subunits (SUs), and recognition models, such as HMMs, trained for each SU [6, 9, 56, 5, 42, 48, 47, 20].

This traditional computer vision approach has seen extensive use in ASLR, with much effort dedicated to investigating different algorithms for body-part tracking, and feature selection and representation. The major challenges faced by systems adopting this approach are: dealing with the visual perception problems, as well as performing sign recognition with all the linguistic complexities that sign languages bring with them.

A more recent approach is that adopted by the DL-based systems, which have made strong inroads into the field of ASLR. DL-based systems are typically trained in an *end-to-end* fashion, with the input consisting of the raw image data [19, 10, 34, 17, 4, 46, 25]. The input is sometimes pre-processed to emphasise particular characteristics of the data (for example, frame differencing to highlight motion), or cropped to focus attention on the hand areas [45, 10, 30, 26]. In these systems, features are extracted automatically by the networks rather than engineered by hand.

DL-based systems tend to outperform traditional computer vision approaches: by working holistically over the entire image, and via automatic feature learning, they are able to capture spatiotemporal context that is too complex to engineer into traditional systems [43, 60]. But some drawbacks include the lack of interpretability of the model parameters ("black-box" nature of DL systems), as well as their general reliance on a *word model* approach to recognition instead of working with SUs.

[1] The value for speech is taken from a website which tracks the current state of the art in speech recognition on a number of standard benchmark datasets: `http://github.com/syhw/wer_are_we`. While the reported value for ASLR is obtained on one of the currently most challenging 'real-life' signing datasets available: `http://www-i6.informatik.rwth-aachen.de/~koller/RWTH-PHOENIX/`.

A few recent ASLR works [22, 41] have adopted a hybrid DL-based approach: they combine deep neural networks with traditional computer vision or sign recognition elements. These systems do away with an end-to-end architecture in favour of a more pipelined approach. For example DL methods are used to determine the signer's pose in terms of body keypoints (or skeleton models), and then these are used as input 'features' for upstream recognition components (which could consist of DL-based or traditional recognition algorithms) [41]. In this way, DL is leveraged to tackle the perception-related challenges, which is something it excels at, and at the same time, some form of control is maintained on the type of 'features' used by the system rather than relying completely on fully automated feature extraction of end-to-end systems. Only limited work has been done so far using this hybrid approach.

Like Gattupalli et al. [22] and Metaxas et al. [41], we adopt a pipelined DL approach, rather than opt for an end-to-end architecture. But in contrast to these two works, our proposed system can use DL for both the visual perception part and the sign recognition part of the system. A main reason for adopting a pipelined DL approach is to increase the interpretability of the system.

As described earlier, practically all DL-based systems work with word models rather than with SUs. To the best of our knowledge, the only exceptions are the works of Camgöz et al. [10], and Metaxas et al. [41].

Like them, we adopt a SU based approach. But unlike these works, which only generate an implicit set of SUs, our novel SUs are explicitly defined, and we show that they are also phonologically meaningful.

Adopting a SU approach offers a number of advantages: since the set of SUs is much smaller than the number of words, less training data is needed. This is of particular relevance to deep networks that require large training sets, without which they can easily *overfit*. Other advantages include better scaling to larger lexica, more robustness to out-of-vocabulary (OOV) signs, and that more complex sign language understanding (for example, decoding the layered meanings of inflected signs, or handling classifier constructions) would not be possible with word-level DL systems.

In the rest of this paper we describe our contributions: (1) novel use of a structure from motion (SfM) factorisation technique to derive hand motion features for use within our pipelined DL system. When these hand motion features are embedded within a trajectory space, we show that semantic meaning is preserved. (2) a novel choice of SUs for use within our DL-based system. We demonstrate that these SUs are phonologically meaningful.

3 Our Approach

In this section, we describe our proposed ASLR system, which is illustrated in Figures 1 and 2. We adopt a SU-based approach to recognition, unlike the majority of DL-based systems that work with word models. We also employ a non end-to-end learning approach, and instead utilise hand features embedded within a trajectory space, since we find that this endows our SUs with phonological meaning, making them both data driven as well as semantically meaningful.

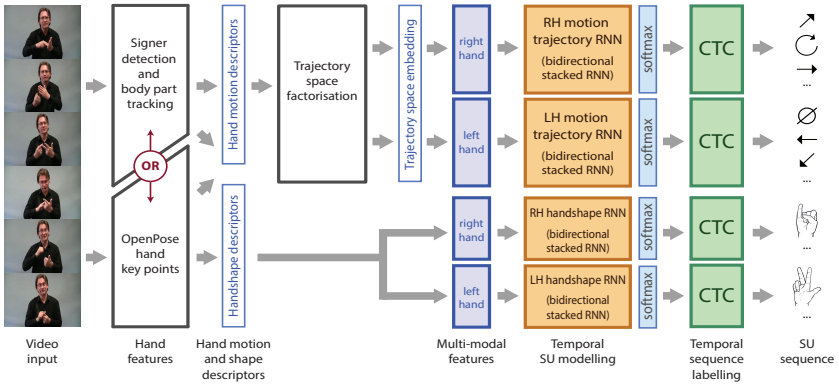


Fig. 1: Our SU RNN-based recognition system. As input, our system can utilise hand features obtained either via a traditional computer vision pipeline, or hand features obtained via a DL-based system. Regardless of the choice of input features, hand motion and handshape descriptors are extracted, and the hand motion descriptors are embedded in a trajectory space via trajectory-space factorisation [3]. These are then fed to the corresponding SU recurrent neural network (RNN). Finally, connectionist temporal classification (CTC) layers provide phonologically-meaningful SU labels.

Figure 1 depicts the four SU networks running in parallel and taking as input the chosen hand features and outputting sequences of SU symbols describing the motions and handshapes of the two hands of the signer. We adopt a parallel approach in order to better handle the multi-modal and concurrent characteristics prevalent in signing activity [21]. Once successfully trained, these four SU networks are then combined together with a second-level network for sign recognition, as shown in Figure 2.

To the best of our knowledge, the only other DL-based works that focus on SUs are those of Camgöz et al. [10], and Metaxas et al. [41]. Camgöz et al. [10] make use of two SU-based networks, trained in an end-to-end fashion. They use whole image frames and pre-cropped hand patches (fixed-size sub-images centred on the hands) respectively as input to their SU networks: the first network learns an implicit intermediate representation (the 'full frame SUs'), while the second network learns handshape SUs. Once trained, their two SU networks are combined together, and then the resulting system is trained for sign recognition.

Similar to Camgöz et al. [10], our system learns handshape SUs and hand motion SUs. But unlike their work, we do not perform end-to-end training of the network, instead using hand features embedded within a trajectory space as input. As a result our SUs carry phonological information, while those of Camgöz et al. [10] are data-driven (video sequence specific), intrinsic, and lack interpretability (particularly for the case of the full-frame network).

Our SUs resemble in principle Metaxas et al.'s [41] choice of linguistically-motivated input features. They estimate the 3D body pose using convolutional

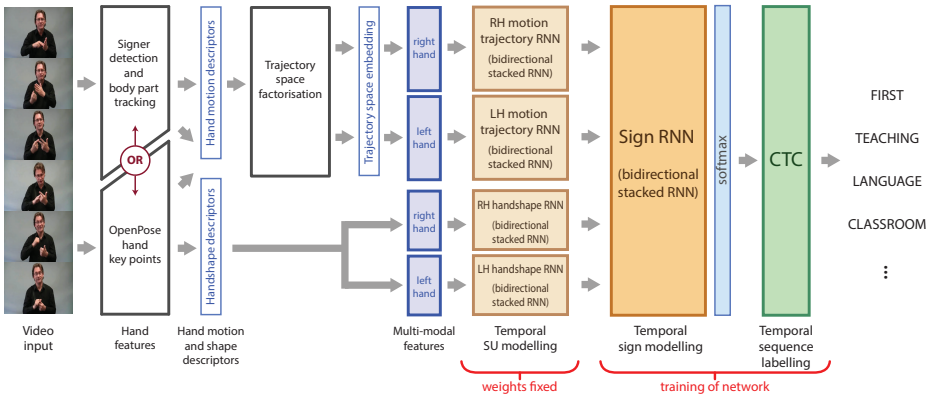


Fig. 2: Extending our SU RNN-based system (of Figure 1) for sign recognition. The output of the SU RNNs are concatenated together and fed to a second-level RNN for sign recognition.

pose machines, and from the pose they derive hand-crafted features such as an indicator of whether a sign is 1-handed or 2-handed, as well as 'contact' flags indicating if a hand is visually close to or touching a key body part (e.g. chin, shoulders, etc.). But then unlike our RNN-based system, Metaxas et al. [41] use a classical method for recognition, a conditional random field (CRF), and reserve the DL techniques solely for the extraction of the pose and pose-related features. And a limitation of their approach is that it works only for isolated sign recognition.

3.1 Hand Features

In our ASLR system, the input features can come from either a traditional computer vision pipeline, or from a DL-based system.

For the former case, we employ a face detector [59] for signer localisation and signing space determination. This is followed by hand region detection within the signing space area based on skin colour and motion cues, obtained using an adaptive skin colour model [57] and multi-frame differencing [27] respectively. Kanade-Lucas-Tomasi (KLT) features [50] are then extracted from the hand regions, grouped together based on motion similarity [14], and tracked in time via the use of the multiple hypothesis tracking (MHT) algorithm [7]. The resulting hand features thus consist of two sets of KLT feature points, one set for each hand of the signer, and their corresponding image-plane trajectories as they are tracked in time [8].

For the alternative DL-based approach, we utilise OpenPose [11] to get 20 hand keypoints for each of the signer's hands. These are then tracked in time to get the corresponding image-plane trajectories.

We consider this choice of input hand features, whether DL or non-DL based, as highlighting one versatile aspect of our non end-to-end learning approach.

3.2 Trajectory Space Factorisation

The image-plane trajectories of the hand features (whether KLT or OpenPose keypoints) exhibit complex motion patterns that are the result of various underlying factors entangled together: such factors can include camera motion, gross body movements of the signer, semantically-meaningful motions of the hands, as well as other non-meaningful hand movements caused by natural variations in articulation, inter-signer variations, dialects, disfluencies and noise.

To separate the real hand motions from the camera and whole body movements of the signer, we propose to use a non-rigid structure from motion (NRSfM) technique based on the *factorisation method* [54]. Furthermore, we interpret the motion patterns of the hands themselves during signing as constituting 'deformations' of the visible shape of the signer.

The 2D image plane trajectories of the features of hand h_i, over a temporal window of length Δt, are centred on the signer's torso, and arranged into the matrix $\mathbf{W} \in \mathbb{R}^{2\Delta t \times P}$, where P is the number of hand features. Then a trajectory space factorisation algorithm [3] is used to separate $\mathbf{W}$ into a product of 3 submatrices:

$$\mathbf{W} = \mathbf{R}\,\mathbf{S} = \mathbf{R}\,\mathbf{\Theta}\,\mathbf{A} \tag{1}$$

where matrix $\mathbf{R}$ describes the camera and the signer's whole body movements (rotations), while matrix $\mathbf{S}$ describes the 3D shape of the signer, in this case the varying 3D shape and orientation of the hands with respect to the centroid of the signer's torso over the time window.

As given in Equation 1, the deformable shape $\mathbf{S}$ can be represented as a weighted linear combination of a trajectory basis in a low-rank trajectory space, with $\mathbf{\Theta}$ being the trajectory basis, while $\mathbf{A}$ contains the weight coefficients. As suggested by Akhter et al. [3], we choose the discrete cosine transform (DCT) basis for $\mathbf{\Theta}$, because it is ideal for representing smooth motions and because of its energy compaction properties. Thus for a single hand feature i, we have:

$$S_i = \textstyle\sum_j^K a_{j,i}\,\theta_j, \quad \theta_j \in \mathbb{R}^{3\times\Delta t},\ a_{j,i} \in \mathbb{R} \tag{2}$$

where S_i is the 3D trajectory of the i^{th} hand feature, θ_j is the j^{th} DCT basis, K is the rank of the DCT basis, and $a_{j,i}$ is the corresponding weight coefficient of basis θ_j and hand feature i.

Traditionally factorisation algorithms run in batch mode, with matrix $\mathbf{W}$ set to the full duration of a video sequence. We adapt the trajectory space factorisation algorithm of Akhter et al. [3] to run in an online mode. We do this by adopting a sliding window approach and setting the window length Δt to 0.5 seconds, determined to being roughly the shortest duration for a single sign. For the sliding window at time t, any missing entries in matrix $\mathbf{W}(t)$ (for example, caused by tracking failures, hand occlusion, or missed hand keypoint detection), are filled via the column space fitting (CSF) matrix completion algorithm [53].

And since the shape and motion recovered by the factorisation method are unique up to a scale and a rotation [3], we align matrices $\mathbf{R}(t)$ with $\mathbf{R}(t-1)$ and $\mathbf{S}(t)$ with $\mathbf{S}(t-1)$ for each successive sliding window using the Procrustes superimposition method [2].

3.3 Trajectory Space Embedding of the Hand Motion Descriptors

Given the recovered motion patterns of the hand features of the signer via the factorisation method, we now use their embedding in the trajectory space as a basis for our choice of SUs. The embedding in trajectory space is given by the DCT coefficients $a_{j,i}$ of Equation 2 (or as grouped together in matrix $\mathbf{A}$ of Equation 1).

Further analysis of these DCT coefficients across different signers and sign languages reveal that similar motion patterns of the hands exhibit consistent and unique patterns in trajectory space leading us to conclude that this trajectory space embedding of the hand features preserves semantic meaning. This is also corroborated by 2D visualisations of the trajectory space coefficients obtained via the t-distributed stochastic neighbour embedding (t-SNE) algorithm [39], like the one shown in Figure 3.

We generate five number summary (FNS) statistics for the DCT coefficients of the hand features and use these hand motion descriptors as input vectors to our hand motion trajectory RNNs (see Figure 1).

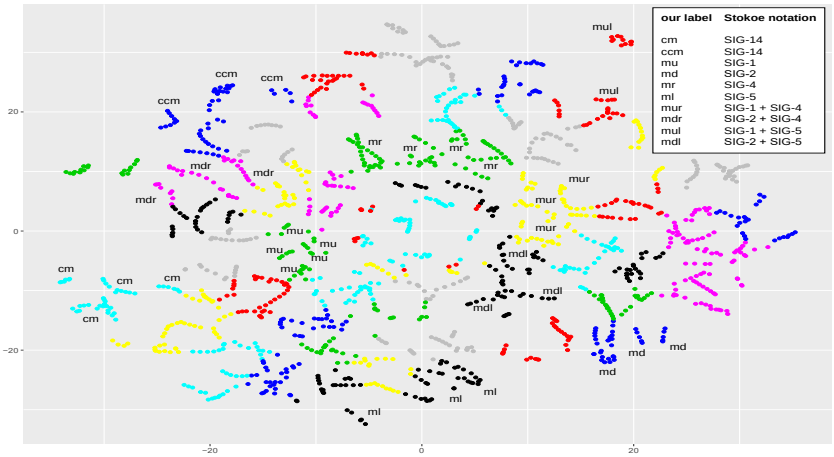


Fig. 3: 2D visualisation (t-SNE) of the trajectory space coefficients and their corresponding SU labels, performed across different signers and sign languages (German sign language (DGS) utterances from the RWTH-Phoenix Weather dataset [35] and Dutch sign language (NGT) utterances from the ECHO NGT dataset [16]). Transitional movements are not labelled. The SU labels follow the Stokoe phonological model [52].

3.4 Handshape Descriptors

We derive the 2D handshape descriptors directly from the hand features of §3.1 on a per-frame basis. The same set of descriptors are used for both the tra-

ditional computer vision pipeline as well as for the OpenPose module. These consist of the bounding box, best-fit ellipse, major axis orientation, aspect ratio, circularity, and convexity measures. For the OpenPose module, we also concatenate the normalised hand keypoints themselves to the handshape descriptors, since these represent salient hand features, in contrast to the arbitrariness of the KLT features of the traditional computer vision pipeline.

3.5 Subunit RNNs

The hand motion and handshape descriptors of the previous sections serve as the inputs to our SU-level RNNs. We use separate networks for the hands of the signer, and for the handshape and hand motion SUs.

We select gated RNNs for our networks, and evaluate the use of both long short-term memorys (LSTMs) and gated recurrent units (GRUs) in our experiments. We employ stacked bidirectional versions of these RNNs, combining the output of the constituent forward and backward layers via vector concatenation. Bidirectional RNNs prove to be better at modelling the temporal nature of our SUs. We use *dropout* within the RNNs, as this is an essential element for preventing overfitting, especially when training the networks with small to medium-sized datasets. We finish off our system by adding a CTC framework [24] for the temporal sequence learning of the SUs, as shown in Figure 1. And we adopt Stokoe's phonological model [52] for the SU class labels.

3.6 The Connectionist Temporal Classification (CTC) framework

We add a CTC framework to our system in order to handle the *temporal sequence labelling problem*: while an RNN performs framewise classification, the RNN's output has no clear one-to-one mapping with the target sequence (the SU sequence), the two sequences are often not of the same length, and the mapping itself is often ambiguous in nature [23, §2].

A number of ASLR works use a combination of RNNs and HMMs as an alternative to the CTC framework [37, 34, 36]: while an RNN provides the framewise class labels, the HMM learns the long-range mappings between the framewise labels and the target labels, since its states can by design "absorb" multiple inputs. But since this method relies on a two-step approach that is iterated several times, it can be sensitive to starting conditions, with accuracy prone to oscillating between iterations.

In contrast, the CTC framework combines the temporal sequence labelling problem directly with the RNN's recognition problem, by defining a CTC loss function that can be incorporated directly within the training mechanism of the RNN, and back-propagated through the network via backpropagation through time (BPTT). In this way, the RNN receives an error for misaligned sequences, even when it predicts the correct labels, and it can learn how to correct for it.

The CTC framework has been used successfully by a number of ASLR works [17, 10, 46], and we follow their lead and employ CTC within our networks.

3.7 Sign-level RNN

So far, our proposed ASLR system consists of four separate RNNs, trained to recognise hand motion and handshape SUs. In order to perform sign recognition, we now combine the networks into one system as shown in Figure 2. This approach is similar in idea to the one employed by Camgöz et al. [10].

We remove the CTC and softmax layers of the trained SU RNNs, freeze their weights, and add a new bidirectional stacked RNN that takes as its input the following feature vector:

$$\mathbf{f}_{\text{input}} = \mathbf{f}_{\text{rh\_motion\_su}} \oplus \mathbf{f}_{\text{lh\_motion\_su}} \oplus \mathbf{f}_{\text{rh\_handshape\_su}} \oplus \mathbf{f}_{\text{lh\_handshape\_su}} \quad (3)$$

where $\oplus$ represents the concatenation operation of vectors. This new RNN together with an associated CTC framework is used to perform sign recognition, producing a sequence of sign glosses as output.

3.8 Training Strategy for our ASLR System

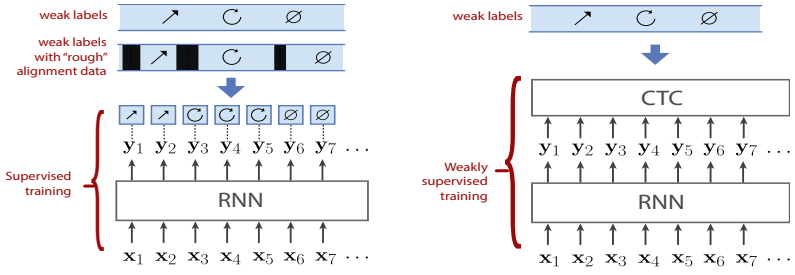
When training our system, we face the challenge that framewise groundtruth labels at SU level is lacking, a common occurrence in the field of ASLR. In the absence of proper groundtruth, we adopt a *weakly supervised* learning strategy for our RNNs, utilising the limited data available to guide the training process.

We utilise simple SU classifiers based on gradient boosting machines (GBMs) [8] that can be trained quickly and with a fraction of the data needed for deep networks. These serve as a source of weakly labelled data, thus providing us with the SU labels and their temporal order. Equipped with such data, weakly supervised training of our networks can proceed via the use of the CTC framework, which allows them to learn to recognise SUs, while at the same time learning to align the input data with the weak labels.

A number of works [33, 34, 17] employ weakly supervised learning for sign recognition. Of these works, the training method of Cui et al. [17] is the most similar to ours. They adopt a three-stage training strategy for their convolutional neural network (CNN)-LSTM based system: first performing end-to-end training of the whole system using weak labels, followed by fine-tuning of the CNN feature extractor on its own, and finally fine-tuning of the LSTM via CTC. Since Cui et al. [17] have only access to an ordered list of signs as groundtruth, with no alignment information, they initialise their training using a "flat start" approach – the input data is partitioned into equal-sized segments corresponding to the list of groundtruth signs.

We adopt a similar multi-stage training strategy. But in contrast to Cui et al. [17], we do not initialise our training with a "flat start" approach. Since we also have access to "rough" segmentation from the simple SU classifiers, we use this as alignment data to initialise the training.

Figure 4 illustrates our training strategy: we first perform supervised learning of the RNN layers (without the CTC layer), using the weak framewise labels generated by the simple SU classifiers. We use early stopping with a high threshold during this training stage, so that the RNN is close to but does not converge



(a) Stage 1 - supervised training (b) Stage 2 - weak supervision

Fig. 4: Our two-stage training strategy. (A) During the first stage, alignment data is used to train the system in supervised mode. The alignment data is subject to segmentation errors (boundary uncertainty indicated by the broad dashed regions), but framewise labels can still be derived from it. (B) Weakly supervised training is performed in stage 2 to refine the RNN weights.

fully. We observe that this stage helps the system to achieve a better initialisation of the layer weights, when compared to a "flat start" approach.

We then add back the CTC layer and continue with weakly supervised training using the SU labels and their temporal order (but with no alignment data). During this stage, the CTC framework helps to fine-tune the weights, improving upon the SU recognition and alignments obtained in the first stage. Training in this stage continues till the network is fully converged.

4 Experiments

We now describe the experiments performed to evaluate our proposed ASLR system. Figure 5 shows the implemented network that we use. The RNN gate type, number of layers, and hidden units are determined via ablation studies, the results of which are not included here due to space limitations. For the CTC part of the network in Figure 5, the CTC loss and analysis layers are only used during training and evaluation of the validation loss, and ignored otherwise. The CTC decoding layer is configured to use beam search decoding, with the beam width set to 100.

We initialise the RNN model weights using *He normal* initialisation [29]. Training for both the strongly supervised and weakly supervised stages operate with roughly the same configuration: we use *mini-batch stochastic gradient descent (SGD)* with *Adam* as the optimisation algorithm [32]. For configuring the mini-batch size and the learning rate, we employ a training schedule inspired by the findings and recommendations of Masters and Luschi [40], and Smith et al. [51]. Training of the RNNs proceeds for 500 epochs, with *early stopping* if there is no further improvement in the loss function over 10 consecutive epochs. The loss function employed depends on whether we are doing weakly supervised

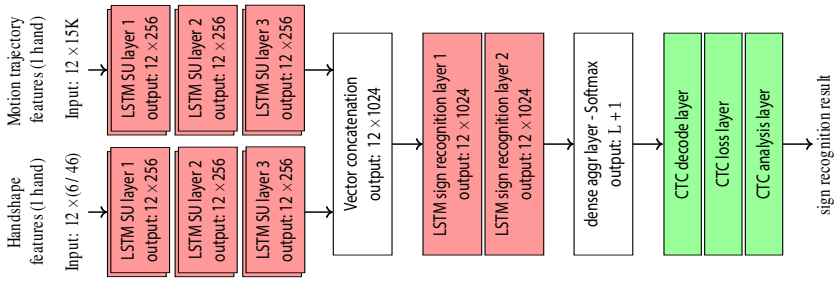


Fig. 5: Implementation of our RNN sign recognition system. In the above figure, K is the number of trajectory bases, while L represents the vocabulary size. The timesteps parameter of the RNNs is set to 12; thus the input to the network is $T \times V = 12 \times V$, where V is the size of the input feature vector.

training (CTC loss is selected), or whether doing strongly supervised training (cross-entropy loss selected).

The datasets available for training the networks at both SU level and sign level, are not balanced in terms of class instances. This is especially true for the handshape SUs and for signs of the RWTH-Phoenix Weather dataset [35]. For example, the ratio of the rarest handshape to the most frequent handshape is of around 0.0007. Because of this strong imbalance, we opted not rely on data augmentation or class re-balancing techniques. Instead, we apply weight corrections to the gradient updates of the SGD algorithm, to compensate for class imbalance – this mechanism adjusts the contribution of each update depending on the how frequent training examples of that particular class are.

5 Results

This section presents the results for the sign recognition experiments, starting with the results of an investigation into the contributions of the hand motion and handshape modalities, for the setup given in Figure 2. The RWTH-Phoenix Weather dataset [35] is used in these experiments, because its large lexicon provides for a realistic evaluation (1230 unique signs), it is multi-signer (9 signers), and has become a standard benchmarking dataset. WER and word accuracy are used as evaluation metrics.

Table 1: Sign recognition results for different modalities.

Modality	WER ↓	Accuracy ↑	Accuracy difference
RH + LH motion trajectory SU RNNs only	0.446	55.4 %	−16.5 %
RH + LH handshape SU RNNs only	0.703	29.7 %	−42.2 %
All four SU RNNs	**0.281**	**71.9** %	–

Table 2: Sign recognition results with different hand features as input.

Input hand features	Ins	Del	Sub	WER ↓	Accuracy ↑
Computer vision pipeline (KLT features)	439	387	1763	0.398	60.2 %
DL-based module (OpenPose keypoints)	221	655	953	**0.281**	**71.9** %

Table 1 shows that both hand motion and handshape modalities are important for sign recognition, with the hand motion SU RNNs contributing the most, compared to handshape RNNs – leaving out the hand motion RNNs, results in a reduction of 42.2 % in accuracy, while removing the handshape RNNs, reduces accuracy by a smaller margin of 16.5 %. While a majority of signs can be discriminated solely based on the gross hand motion trajectory, there are a number of signs (*minimal pairs*) whose appearances differ only by their handshapes. Hence while handshape on its own is a poor discriminator, in conjunction with hand motion, it allows the sign-level RNN to achieve a better recognition rate than with hand motion alone.

Table 2 gives the results when using different types of hand features as input.

Figure 6 presents some qualitative results. Sign alignments are included for illustration purposes only, and are not evaluated since we lack groundtruth – these alignments are extracted directly from the softmax output layer (see Figure 5). Sign evaluation is performed via the CTC analysis layer and reported in terms of insertions, deletions and substitutions.

Each example of Figure 6 (A) to (E) is performed by a different signer. For example in (C), the sign '`REGEN`' is mistaken for a different sign. This is mainly due to sign variations that do not appear within the training set, e.g., hands moving out of sync with each other in this particular case. Figure 7 shows more examples for sign '`REGEN`' which are classified correctly, and others which are misclassified. The canonical sign consists of the two hands moving in unison downwards once or multiple times, and exhibiting handshape `DEZ-5`. Sign variations include: one-handed versions, out-of-sync movements, shortened movements, and varying hand orientations. Despite all these variations, the RNN does a relatively good job at learning the variations, with some exceptions – WER for this sign is 0.85 (25 substitutions, 18 deletions, 0 insertions, out of a total of 290 instances in the test fold).

We now compare our sign recognition results against the state of the art – this is given in Table 3. The RWTH-Phoenix Weather dataset comes with the training, validation, and test folds pre-defined, thus ensuring a fair comparison.

We can observe that our system outperforms many of the recent DL-based systems, except for the work of Koller et al. [34]. The difference in accuracy between their work and ours is of 1.3 %. And the difference between our work and the rest of the systems is of 10.2 %.

While Koller et al.'s system [34] does better than ours, it is trained in an end-to-end fashion. It takes the full video frame as input, and uses a CNN for

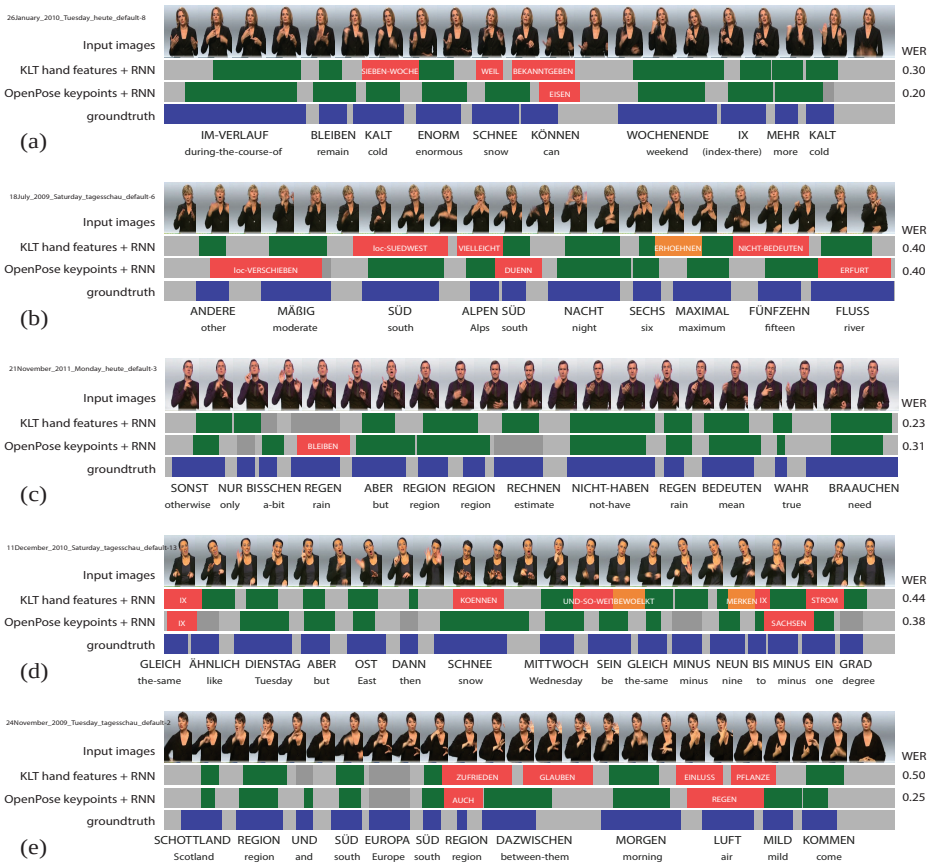


Fig. 6: Qualitative sign recognition results. Groundtruth is depicted in blue, substitution errors in red, insertion errors in orange, and deletions in dark gray.

Fig. 7: Examples of correct matches (outlined in green) and mismatches (in red) for DGS sign "REGEN" (English, 'rain'). Each pair of images depicts the start and end frame of the sign.

Table 3: Comparison of our proposed system (highlighted in blue) against the state of the art (PHOENIX Weather dataset).

Method	Ins %	Del %	Sub %	WER ↓ (test fold)	Accuracy ↑
CNN + HMM + expectation maximisation (EM) (Deep hand) [33]	0.04	0.19	0.22	0.451	54.9 %
CNN + LSTM + CTC (SubUNets) [10]	0.04	0.15	0.22	0.407	59.3 %
CNN + HMM (Deep Sign) [37]	0.05	0.13	0.21	0.388	61.2 %
CNN + LSTM + CTC [17]	0.08	0.12	0.19	0.387	61.3 %
Hierarchical attention networks [30]	–	–	–	0.383	61.7 %
Our DL-based system	**0.03 (221)** †	**0.10 (655)**	**0.15 (953)**	**0.281**	**71.9** %
CNN + RNN method (Re-Sign) [34]	–	–	–	**0.268**	**73.2** %

† Numbers in brackets specify the numbers of insertions, deletions, and substitutions. We convert these to percentage values for the purpose of comparison with other works.

automatic feature extraction. In contrast, our system makes use of trajectory space SUs that are phonologically meaningful, and our RNN-based system is trained on these SUs.

We thus argue that our approach provides a number of benefits, including that of better interpretability, albeit at a relatively small cost in accuracy reduction.

6 Conclusions

In this paper we have described a two-stage DL-based ASLR system. Our system takes as input hand features derived from either a traditional computer vision pipeline or a from hand keypoints obtained via OpenPose. A trajectory space factorisation method is then applied to extract the hand motions and these are embedded within a low-rank trajectory space. We demonstrated how this trajectory space embedding preserves semantic meaning, allowing us to base our choice of SUs on descriptors derived from the embedding coefficients. These descriptors are then fed to our SU RNNs for training, utilising a CTC framework to handle the temporal sequence labelling problem. Once the SU RNNs are trained, a second-level RNN is added for sign recognition.

We performed a number of investigations, first to choose between various design options for the RNNs, and then to evaluate the sign recognition accuracy of our system. We compared the results with the state of the art, where we found that our system surpasses ($\geq 10.2\,\%$) many of the recent works that employ DL in ASLR. Only one recent work performs marginally better ($1.3\,\%$) than our proposed system. But we argue that our system offers other benefits, such as phonologically-meaningful SUs and better interpretability.

Future work will look at extending the SU RNNs to handle other modalities, including non-manual signals such as facial expressions and mouthings. We will also investigate how our choice of phonologically meaningful SUs operate across different signers and different sign languages.

References

1. von Agris, U., Knorr, M., Kraiss, K.: The significance of facial features for automatic sign language recognition. In: Proc. 8th Int. Conf. on Automatic Face & Gesture Recognition (FG). IEEE (2008)
2. Akhter, I., Sheikh, Y., Khan, S., Kanade, T.: Nonrigid Structure from Motion in Trajectory Space. In: Koller, D., et al. (eds.) Advances in Neural Information Processing Systems (NIPS), p. 41. Curran Associates Inc. (2009)
3. Akhter, I., Sheikh, Y., Khan, S., Kanade, T.: Trajectory Space: A Dual Representation for Nonrigid Structure from Motion. IEEE TPAMI **33**(7), 1442–1456 (2011)
4. Avola, D., Bernardi, M., Cinque, L., Foresti, G.L., Massaroni, C.: Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures. IEEE Trans. on Multimedia (2018)
5. Awad, G., Han, J., Sutherland, A.: Novel boosting framework for subunit-based sign language recognition. In: Proc. ICIP. pp. 2729–2732. IEEE (2009)
6. Bauer, B., Kraiss, K.F.: Towards an automatic sign language recognition system using subunits. In: Wachsmuth, I., Sowa, T. (eds.) Int. Gesture Workshop on Gesture and Sign Languages in Human-Computer Interaction (GW). vol. LNAI 2298, pp. 64–75. Springer (2002)
7. Blackman, S.S.: Multiple Hypothesis Tracking For Multiple Target Tracking. IEEE Aerospace and Electronics Systems Magazine **19**(1), 5–18 (2004)
8. Borg, M., Camilleri, K.P.: Towards a Transcription System of Sign Language Video Resources via Motion Trajectory Factorisation. In: Proc. of the 2017 ACM Symposium on Document Engineering. pp. 163–172. DocEng'17, ACM, New York, NY, USA (2017). https://doi.org/10.1145/3103010.3103020
9. Bowden, R., Windridge, D., Kadir, T., Zisserman, A., Brady, M.: A linguistic feature vector for the visual interpretation of sign language. In: Proc. ECCV. pp. 390–401. Springer (2004)
10. Camgöz, N.C., Hadfield, S., Koller, O., Bowden, R.: SubUNets: end-to-end hand shape and continuous sign language recognition. In: Proc. ICCV. IEEE (Oct 2017)
11. Cao, Z., Hidalgo, G., Simon, T., Wei, S.E., Sheikh, Y.: OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In: arXiv preprint 1812.08008 (2018)
12. Charles, J., Pfister, T., Magee, D., Hogg, D., Zisserman, A.: Upper body pose estimation with temporal sequential forests. In: Proc. BMVC (2014)
13. Cheok, M.J., Omar, Z., Hisham Jaward, M.: A review of hand gesture and sign language recognition techniques. Int. Journal of Machine Learning and Cybernetics **10** (2017). https://doi.org/10.1007/s13042-017-0705-5
14. Choi, S., Kim, T., Yu, W.: Performance evaluation of RANSAC family. In: Proc. BMVC (2009)
15. Cooper, H., Holt, B., Bowden, R.: Sign Language Recognition. In: Moeslund, T.B., et al. (eds.) Visual Analysis of Humans - Looking at People, pp. 539–562. No. 231135, Springer (2011)
16. Crasborn, O., et al.: ECHO Data Set for Sign Language of the Netherlands (NGT) (2004)
17. Cui, R., Liu, H., Zhang, C.: Recurrent Convolutional Neural Networks for Continuous Sign Language Recognition by Staged Optimization. In: Proc. CVPR. pp. 1610–1618. IEEE (Jul 2017). https://doi.org/10.1109/CVPR.2017.175

18. Efthimiou, E., Fotinea, S.E., Hanke, T., Glauert, J., Bowden, R., Braffort, A., Collet, C., Maragos, P., Lefebvre-Albaret, F.: Sign Language technologies and resources of the Dicta-Sign project. In: Proc. Int. Conf. on Language Resources and Evaluation (LREC), RPSL Workshop. ELRA (2012)
19. Fang, B., Co, J., Zhang, M.: DeepASL: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation. In: Proc. 15th ACM Conf. on Embedded Network Sensor Systems (SenSys). ACM (2017). https://doi.org/10.1145/3131672.3131693
20. Farag, I., Brock, H.: Learning motion disfluencies for automatic sign language segmentation. In: Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). pp. 7360–7364 (May 2019). https://doi.org/10.1109/ICASSP.2019.8683523
21. Fenlon, J., Cormier, K., Brentari, D.: The Phonology of Sign languages, pp. 453–475. Routledge (2017). https://doi.org/10.4324/9781315675428
22. Gattupalli, S., Ghaderi, A., Athitsos, V.: Evaluation of Deep Learning Based Pose Estimation for Sign Language Recognition. In: Proc. 9th Int. Conf. PErvasive Technologies Related to Assistive Environments (PETRA). ACM (2016)
23. Graves, A.: Supervised Sequence Labelling with Recurrent Neural Networks, Studies in Computational Intelligence, vol. 385. Springer-Verlag (2012). https://doi.org/10.1007/978-3-642-24797-2
24. Graves, A., Fernández, S., Gomez, F.: Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: Proc. Int. Conf. on Machine Learning (ICML). pp. 369–376 (2006)
25. Guo, D., Tang, S., Wang, M.: Connectionist Temporal Modeling of Video and Language: a Joint Model for Translation and Sign Labeling. In: Proc. 28th Int. Joint Conf. on Artificial Intelligence, IJCAI. pp. 751–757 (2019)
26. Guo, D., Zhou, W., Li, H., Wang, M.: Hierarchical LSTM for sign language translation. In: Proc. 32nd AAAI Conf. on Artificial Intelligence. pp. 6845–6852 (2018)
27. Guo, J., Wang, J., Bai, R., Zhang, Y., Li, Y.: A new moving object detection method based on frame-difference and background subtraction. IOP Conference Series: Materials Science and Engineering **242**(1), 012115 (2017)
28. Hanson, V.L.: Computing technologies for deaf and hard of hearing users. In: Sears, A., Jacko, J.A. (eds.) Human-Computer Interaction: Designing for Diverse Users and Domains, chap. 8, pp. 885–893. Taylor & Francis Group (2009). https://doi.org/10.1201/9781420088885
29. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. In: Proc. ICCV. pp. 1026–1034 (2015). https://doi.org/10.1109/ICCV.2015.123
30. Huang, J., Zhou, W., Zhang, Q., Li, H., Li, W.: Video-Based Sign Language Recognition Without Temporal Segmentation. In: 32nd Conf. on Artificial Intelligence (AAAI). pp. 2257–2264. AAAI (2018)
31. Kelly, D., McDonald, J., Markham, C.: Recognition of Spatiotemporal Gestures in Sign Language Using Gesture Threshold HMMs. Machine Learning for Vision-Based Motion Analysis **Advances i**, 307–348 (2011)
32. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. In: ICLR 2015. p. 13 (2015)
33. Koller, O., Ney, H., Bowden, R.: Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data is Continuous and Weakly Labelled. In: Proc. CVPR. pp. 3793–3802. IEEE (Jun 2016). https://doi.org/10.1109/CVPR.2016.412
34. Koller, O., Zargaran, S., Ney, H.: Re-Sign: Re-Aligned End-to-End Sequence Modelling with Deep Recurrent CNN-HMMs. In: Proc. CVPR. pp. 3416–3424. IEEE (Jul 2017). https://doi.org/10.1109/CVPR.2017.364

35. Koller, O., Forster, J., Ney, H.: Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. Computer Vision and Image Understanding **141**, 108–125 (2015)
36. Koller, O., Zargaran, S., Hermann, N., Bowden, R.: Deep Sign: Enabling Robust Statistical Continuous Sign Language Recognition via Hybrid CNN-HMMs. Int. J. of Computer Vision **126**(12), 1311–1325 (2018)
37. Koller, O., Zargaran, S., Ney, H., Bowden, R.: Deep Sign: Hybrid CNN-HMM for Continuous Sign Language Recognition. In: Proc. BMVC (2016)
38. Lüscher, C., Beck, E., Irie, K., Kitza, M., Michel, W., Zeyer, A., Schlüter, R., Ney, H.: RWTH ASR Systems for LibriSpeech: Hybrid vs Attention. In: Proc. Interspeech 2019. pp. 231–235 (2019). https://doi.org/10.21437/Interspeech.2019-1780
39. van der Maaten, L., Hinton, G.: Visualizing High-Dimensional Data using t-SNE. Journal of Machine Learning Research **9**, 2579–2605 (2008)
40. Masters, D., Luschi, C.: Revisiting Small Batch Training for Deep Neural Networks. CoRR (2018)
41. Metaxas, D., Dilsizian, M., Neidle, C.: Linguistically-driven framework for computationally efficient and scalable sign recognition. In: Calzolari, N., et al. (eds.) Proc. 11th Int. Conf. on Language Resources and Evaluation (LREC). ELRA (2018)
42. Oszust, M., Wysocki, M.: Modelling and recognition of signed expressions using subunits obtained by data–driven approach. In: Ramsay, A., Agre, G. (eds.) Artificial Intelligence: Methodology, Systems, and Applications: 15th Int. Conf., AIMSA, Proc., pp. 315–324. Springer (2012)
43. Panzner, M., Cimiano, P.: Comparing hidden markov models and long short term memory neural networks for learning action representations. In: Pardalos, P.M., et al. (eds.) Machine Learning, Optimization, and Big Data. pp. 94–105. Springer, Cham (2016)
44. Pfister, T., Charles, J., Everingham, M., Zisserman, A.: Automatic and efficient long term arm and hand tracking for continuous sign language tv broadcasts. In: Proc. BMVC (2012)
45. Pigou, L., Herreweghe, M.V., Dambre, J.: Gesture and sign language recognition with temporal residual networks. In: Proc. ICCV Workshops. pp. 3086–3093 (Oct 2017). https://doi.org/10.1109/ICCVW.2017.365
46. Pu, J., Zhou, W., Li, H.: Dilated convolutional network with iterative optimization for continuous sign language recognition. In: Proc. 27th Int. Joint Conf. on Artificial Intelligence (IJCAI-18). pp. 885–891 (2018)
47. Pu, J., Zhou, W., Zhang, J., Li, H.: Sign language recognition based on trajectory modeling with HMMs. In: Tian, Q., et al. (eds.) MultiMedia Modeling. pp. 686–697. Springer, Cham (2016)
48. Sako, S., Kitamura, T.: Subunit Modeling for Japanese Sign Language Recognition Based on Phonetically Depend Multi-stream Hidden Markov Models. In: Stephanidis, C., Antona, M. (eds.) Universal Access in Human-Computer Interaction (UAHCI). Design Methods, Tools, and Interaction Techniques for eInclusion, pp. 548–555. Springer (2013)
49. Schirmer, B.R.: Psychological, social, and educational dimensions of deafness. Allyn & Bacon (2001)
50. Shi, J., Tomasi, C.: Good features to track. In: Proc. CVPR. pp. 593–600 (1994)
51. Smith, S.L., Kindermans, P.J., Le, Q.V.: Don't Decay the Learning Rate, Increase the Batch Size. In: Int. Conf. on Learning Representations (2018)
52. Stokoe, W.C.: Sign Language Structure. Annual Review of Anthropology **9**(1), 365–390 (1980). https://doi.org/10.1146/annurev.an.09.100180.002053

53. Sun, Z.L., Fang, Y., Shang, L., Zhu, X.G.: A missing data estimation approach for small size image sequence. In: 5th Int. Conf. on Intelligent Control and Information Processing. pp. 479–481. IEEE (Aug 2014)
54. Tomasi, C., Kanade, T.: Shape and motion from image streams under orthography: A factorization method. Int. J. Comput. Vision **9**(2), 137–154 (1992)
55. Van Staden, A., Badenhorst, G., Ridge, E.: The benefits of sign language for deaf learners with language challenges. Per Linguam **25**(1), 44–60 (2009)
56. Vogler, C., Goldenstein, S.: Toward computational understanding of sign language. In: Technology and Disability. vol. 20, pp. 109–119. IOS Press (2008)
57. Wimmer, M., Radig, B.: Adaptive skin color classificator. In: Proc. 1st ICGST Int. Conf. on Graphics, Vision and Image Processing (GVIP). pp. 324–327 (2005)
58. Yang, R., Sarkar, S., Loeding, B.: Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming. IEEE TPAMI **32**(3), 462–477 (2010)
59. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters **23**(10), 1499–1503 (Oct 2016)
60. Zheng, L., Liang, B., Jiang, A.: Recent Advances of Deep Learning for Sign Language Recognition. In: Int. Conf. on Digital Image Computing: Techniques and Applications (DICTA) (Nov 2017)