Gesture and Sign Language Recognition with Temporal Residual Networks

Lionel Pigou, Mieke Van Herreweghe and Joni Dambre
Ghent University
{lionel.pigou, mieke.vanherreweghe, joni.dambre}@ugent.be

Abstract

Gesture and sign language recognition in a continuous video stream is a challenging task, especially with a large vocabulary. In this work, we approach this as a framewise classification problem. We tackle it using temporal convolutions and recent advances in the deep learning field like residual networks, batch normalization and exponential linear units (ELUs). The models are evaluated on three different datasets: the Dutch Sign Language Corpus (Corpus NGT), the Flemish Sign Language Corpus (Corpus VGT) and the ChaLearn LAP RGB-D Continuous Gesture Dataset (ConGD). We achieve a 73.5% top-10 accuracy for 100 signs with the Corpus NGT, 56.4% with the Corpus VGT and a mean Jaccard index of 0.316 with the ChaLearn LAP ConGD without the usage of depth maps.

1. Introduction

Sign language recognition (SLR) systems have many different use cases: corpus annotation, in hospitals, as a personal sign language learning assistant or translating daily conversations between signers and non-signers to name a few. Unfortunately, unconstrained SLR remains a big challenge. Sign language uses multiple communication channels in parallel with high visible intra-sign and low inter-sign variability compared to common classification tasks. In addition, publicly available annotated corpora are scarce and not intended for building classifiers in the first place.

A common approach in SLR is to get around the high dimensionality of image-based data by engineering features to detect joint trajectories [2], facial expressions [16] and hand shapes [19] as an intermediate step. Data gloves [20], colored gloves [29] or depth cameras [1] are often deployed in order to obtain a reasonable identification accuracy.

In recent years, deep neural networks achieve state-of-the-art performance in many research domains including image classification [26], speech recognition [9] and human pose estimation [21]. We start seeing its integration into the SLR field with the recognition of isolated signs using 3D convolutional neural networks (CNNs) [23] and continuous SLR using recurrent CNNs [6].

A task that is closely related to SLR is gesture recognition. Deep neural networks have proven to be successful for this problem, given a small vocabulary (20 gestures) [22, 30] and/or with a multi-modal approach [24, 18].

In this work, we investigate large vocabulary gesture recognition and SLR using deep neural networks with up-to-date architectures, regularization techniques and training methods. To achieve this, we approach the problem as a continuous framewise classification task, where the temporal locations of gestures and signs are not given during evaluation. The models are tested on the Dutch Sign Language Corpus (Corpus NGT) [4, 5], the Flemish Sign Language Corpus (Corpus VGT) [27] and the ChaLearn LAP RGB-D Continuous Gesture Dataset (ConGD) [28].

2. Methodology

Given a video file, we want to produce predictions for every frame. With a sliding window, a number of frames are fed into the model and output a prediction for either the middle frame (many-to-one) or all input frames (many-to-many). The input frames undergo some minimal preprocessing before feeding it to the model: the RGB channels are converted to gray-scale, resized to 128x128 pixels and the previous frame is subtracted from the current frame to remove static information (Figure 1).

The models are inherently CNNs [15] with recent im-

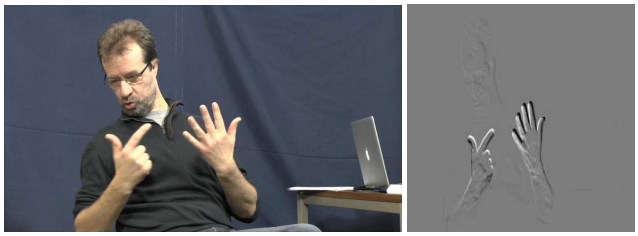

Figure 1. *Left:* Original RGB-data. *Right:* Model input. The RGB channels are converted to gray-scale, resized to 128x128 pixels and the previous frame is subtracted from the current frame to remove static information.

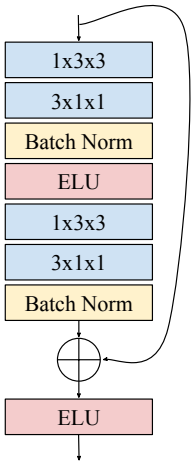

Figure 2. The residual building-block used in the deep neural networks for both models.

provements to facilitate the classification problem. CNNs are models that allow to learn a hierarchy of layered features instead of manually extracting them. They are among the most successful techniques in deep learning and have proven to be very successful at recognizing patterns in high dimensional data such as images, videos and audio. Our models also make use of temporal convolutions and recurrence to cope with the spatiotemporal nature of the data.

2.1. Residual Building-Block

The models in this paper use a residual network layout [10] consisting of so-called residual building blocks. Our adapted residual block is depicted in Figure 2.

The first two operations in the residual block are spatial convolutions with filter size 3x3 followed by temporal convolutions with filter size 3. This enables the extraction of hierarchies of motion features and thus the capturing of temporal information from the first layer on, instead of depending on higher layers to form spatiotemporal features. Performing three-dimensional convolutions is one approach to achieve this. However, this leads to a significant increase in the number of parameters in every layer, making this method more prone to overfitting. Therefore, we decide to factorize this operation into two-dimensional spatial convolutions and one-dimensional temporal convolutions. This leads to fewer parameters and optionally more nonlinearity if one decides to activate both operations. We opt to not include a bias or another nonlinearity in the spatial convolution step.

First, we compute spatial feature maps s_t for every frame x_t. A pixel at position (i, j) of the k-th feature map is determined as follows:

$$s_{tij}^{(k)} = \sum_{n=1}^{N} \left(W_{\text{spat}}^{(kn)} * x_t^{(n)} \right)_{ij} , \tag{1}$$

where N is the number of input channels and W_{spat} are trainable parameters. Finally, we convolve across the time dimension for every position (i, j) and add a bias $b^{(k)}$:

$$v_{tij}^{(k)} = b^{(k)} + \sum_{m=1}^{M} \left(W_{\text{temp}}^{(km)} * s_{ij}^{(m)} \right)_t , \tag{2}$$

where the variables W_{temp} and b are trainable parameters and M is the number of spatial feature maps.

The convolutions are followed by batch normalization [12]. This method will shift the internal values to a mean of zero and scale to a variance of one in every layer across the mini-batch. This will prevent the change of distribution of every layer during training, the so-called internal covariant shift problem. We found that training with batch normalization was crucial, because the network didn't converge without it.

The nonlinearity in the model is introduced by Exponential Linear Units (ELUs) [3]. This activation function speeds up training and achieves better regularization than Rectified Linear Units (ReLUs) [17] or Leaky Rectified Linear Units (LReLUs).

Following the original building block in [10], the previously described operations are stacked one more time, with the exception of the ELU. Right before the final activation, the input of the block is added. This addition is what makes the model a residual network. Residual networks allow to train deeper networks more easily, because there are shortcut connections (the aforementioned addition) to the input layers. This solves the degradation problem, where traditional networks see a decrease in performance when stacking too many layers.

2.2. Network Architecture

Two different architectures are employed for the SLR and the gesture recognition task. The SLR network has a many-to-one configuration (Figure 3) and the gesture recognition network has a many-to-many configuration (Figure 4). The reason for this difference is that we want to have better control over the training labels in the SLR case. The many-to-many configuration would try to model too much silent (or blank) annotations, while the gesture data does not have silent labels.

Both networks start with a three dimensional convolutional layer with filter size 7x7x7 and stride 1x2x2. This first layer allows us to use a higher spatial resolution (128x128) without increasing computation time. Replacing this layer with residual blocks would force us to use a

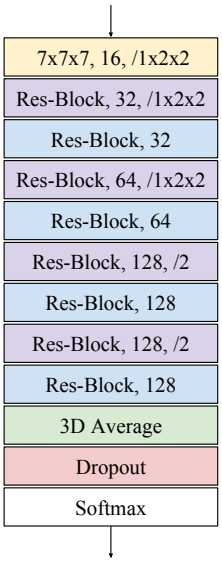

Figure 3. The deep residual neural network used for sign language recognition on the Corpus NGT [4, 5] and the Corpus VGT [27].

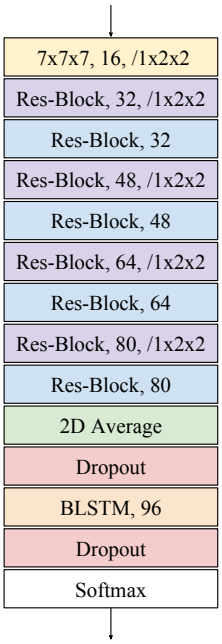

Figure 4. The deep residual neural network used for gesture recognition on ChaLearn ConGD [28].

small mini-batch size due to memory constraints and the computation time would increase twofold or more.

The first layer is followed by eight residual blocks, where we decrease the feature map dimensionality every odd layer. This results in seventeen convolutional layers in total. After the residual blocks, we take the average of every feature map. In the many-to-many case we only take the spatial average.

The SLR network ends with a dropout layer and a softmax layer. The gesture recognition network adds a bidirectional LSTM [11] (with peephole connections [8]), which enables us to process sequences in both temporal directions.

2.3. Model Training

We train our models in an end-to-end fashion, backpropagating through time (BTT) for the recurrent architecture. The network parameters are optimized by minimizing the cross-entropy loss function using mini-batch gradient descent with the *Adam* update rule [14]. Adam is an optimization algorithm based on adaptive estimates of lower-order moments of the gradients. We found that Adam works great in practice, especially when experimenting with very different layer types in the same model. Leaving the proposed hyper-parameters of Adam untouched, we observed improved training convergence in comparison to SGD with Nesterov momentum. All our models are trained the same way with early stopping, a mini-batch size of 24, a learning rate of 10^{-3} and an exponential learning rate decay. Before training, we initialize the weights with a random orthogonal initialization method [25].

Lastly, it is worth mentioning that we use data augmentation. Data augmentation has a significant impact on generalization. For all our trained models, we used the same augmentation parameters: $[-32, 32]$ pixel translations, $[-8, 8]$ rotation degrees, $[\frac{1}{1.5}, 1.5]$ image scaling factors and random horizontal flips. From each of these intervals, we sample a random value for each video fragment and apply the transformations online using the CPU.

3. Experiments

3.1. Sign Language Recognition

The two corpora used to explore SLR (Corpus VGT [27] and Corpus NGT [4, 5]) have similar camera setups and use very similar annotation rules with identical software (ELAN). Both corpora consist of Deaf signers that perform tasks such as retelling comic strips, discuss an event and debating on chosen topics. For each corpus, the 100 most frequently used signs are extracted together with their gloss. A gloss is the written form of a sign.

As Figure 5 shows, there is a class imbalance for both corpora. This means that accuracy measures will be highly skewed. For example, only predicting the most common sign (which is "ME") for every sample across the whole dataset already results in 30.9% and 11.2% accuracy for the Corpus NGT and the Corpus VGT respectively.

The SLR data is split into three sets (for each corpus): 70% training set, 20% test set and 10% validation set. The training set is used to optimize the neural networks, the val-

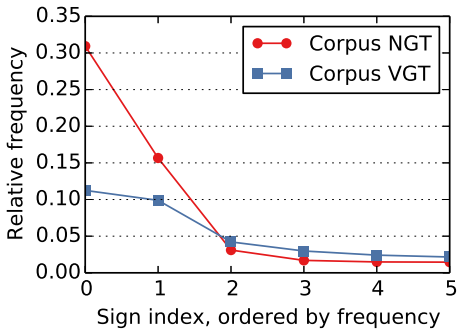

Figure 5. The relative frequency for the five most common signs in both corpora. The class imbalance is significant in both corpora, but is especially prevalent for the Corpus NGT [4, 5].

idation set is used for evaluation during training and the test set is used to evaluate the final models.

The model takes an input of 16 frames, sampled at 25 frames per second with a resolution of 128x128 pixels. The network makes predictions for the 8^{th} frame, as it has a many-to-one configuration. During training, random fragments of 16 frames are sampled. During evaluation a sliding window across the entire video file is employed. Only frames of known signs are considered for evaluation to eliminate the dependency on the amount of silences (which can be detected by motion vectors) and unknown signs.

3.1.1 Corpus NGT

The Corpus NGT [4, 5] (Figure 6) contains Deaf signers using Dutch Sign Language from the Netherlands. This project was executed by the sign language group at the Radboud University Nijmegen. Every narrative or discussion fragment forms a clip of its own, with more than 2000 clips. We extracted a total of 55224 video-gloss pairs from 78 different Deaf signers.

The top-N accuracy is a measure indicating the probability that the correct answer is within the model's N best guesses. The framewise top-N accuracies of the test set for the Corpus NGT are depicted in Figure 8. The model achieves a top-1, top-3, top-5 and top-10 accuracy of 39.9%, 57.9%, 64.4% and 73.3% respectively for 100 signs. This is especially interesting for automatic corpus annotation, where providing a list with the N best guesses is appropriate.

The confusion matrix shows the fraction of true positives for each class (each sign) on the diagonal. It also tells us which classes it gets confused with. To have a better insight into the model's performance, we show the confusion matrix in Figure 9. Not surprisingly, almost all classes get confused with frequently occurring ones. The network learned to bet on common glosses when it is unsure about a certain input, because more often than not it will get rewarded for that. Other misclassification is due to signs that are hard to distinguish from each other.

3.1.2 Corpus VGT

The Corpus VGT [27] (Figure 7) uses Flemish Sign Language. The project started in Juli 2012 and ended in November 2015 at Ghent University, in collaboration with the Linguistics Group VGT of KU Leuven Campus Antwerp, and promoted by Prof. Dr. Mieke Van Hereweghe (Ghent University) and Prof. Dr. Myriam Vermeerbergen (KU Leuven Campus Antwerp). The corpus contains 140 hours of video and a small fraction is annotated. After cleaning the data, we extracted a total of 12599 video-gloss pairs from 53 different Deaf signers.

To cope with the smaller amount of annotations for the Corpus VGT compared to the Corpus NGT, we transfer all the parameters from the Corpus NGT model and use them as initial weights. This is a form of *transfer learning* or *pretraining*, where the knowledge of one or more domains (in this case the Corpus NGT) is useful for other domains. Our motivation is that the learned features for both domains should be similar, except for the softmax classifier. All sign languages have similar visual features: they consist of hand, arm, face and body expressions. We hope to capture these generic building blocks in order to boost the performance for the Corpus VGT.

In Figure 10, the top-N accuracies are shown. It achieves a top-1, top-3, top-5 and top-10 accuracy of 18.2%, 32.3%,

Figure 6. A sample from the Corpus NGT (Radboud University Nijmegen) [4, 5], filmed from two viewpoints.

Figure 7. A sample from the Corpus VGT (Ghent University) [27], filmed from three viewpoints.

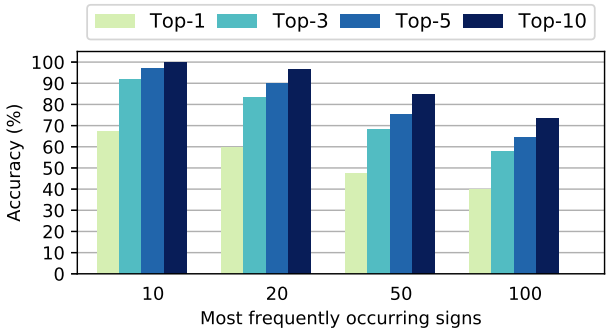

Figure 8. **Corpus NGT** [4, 5] top-N accuracies, indicating the probability of the correct answer being within the model's N best guesses.

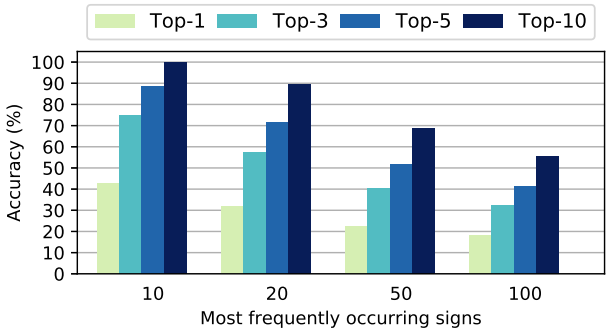

Figure 10. **Corpus VGT** [27] top-N accuracies, indicating the probability of the correct answer being within the model's N best guesses.

41.4% and 55.7% respectively for 100 signs. The resulting confusion matrix is shown in Figure 11. The errors are more spread out than the ones for the Corpus NGT, because the class imbalance is less prevalent.

3.2. ChaLearn LAP ConGD

The ChaLearn LAP RGB-D Continuous Gesture Dataset (ConGD) [28] is a large-scale gesture dataset and has been used for two rounds of classification challenges (2016 and 2017). The gestures come from multiple sources, including sign language, underwater signs, helicopter and traffic signals, pantomimes and symbolic gestures, Italian gestures, and body language (Figure 12) The database consists of 249 different gesture classes performed by 21 individuals. Each individual belongs to either the training, the validation or the test set. The videos are recorded with a Microsoft Kinect RGB-D camera. Each class occurs at least 200 times with 47933 gestures in 22535 videos files. Each video contains one or more gestures and are annotated with the start and end frames.

The challenge is approached in a similar fashion as SLR. We only consider the RGB channels and discard the depth map, as we want to contribute by using a model that does not need a depth sensor, although we realize we throw away a lot of useful information. The difference with the SLR is that the model takes an input of 32 frames, sampled at 10 frames per second. Furthermore, the network has a many-to-many configuration (Figure 4) with a bidirectional LSTM stacked on top of the residual network.

Lastly, a postprocessing modus-filter of size 39 is applied on the final framewise predictions. The modus of a series of

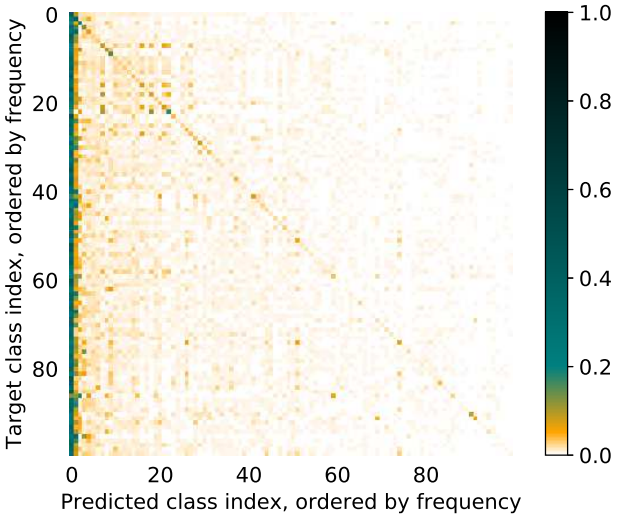

Figure 9. **Corpus NGT** [4, 5] confusion matrix indicating the classification performance of the deep neural network.

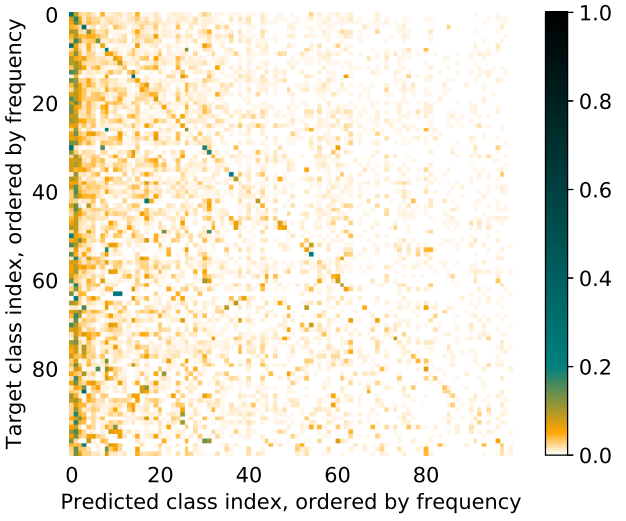

Figure 11. **Corpus VGT** [27] confusion matrix indicating the classification performance of the deep neural network.

Figure 12. A few samples from the ChaLearn LAP ConGD challenge [28].

integers is the most frequently occurring one. This smooths out the noisy predictions of the model. This method is based on the fact that annotations do not change more than once over a time-window of about 20 frames.

We follow the ChaLearn LAP 2017 Challenge score to measure the performance of our model. The score is based on the Jaccard index, which is defined as follows:

$$J_{s,n} = \frac{A_{s,n} \cap B_{s,n}}{A_{s,n} \cup B_{s,n}}. \tag{3}$$

The binary ground truth for gesture category n in sequence s is denoted as the binary vector $A_{s,n}$, whereas $B_{s,n}$ denotes the binary predictions. The Jaccard index $J_{s,n}$ can be seen as the overlap rate between $A_{s,n}$ and $B_{s,n}$. To obtain the final score, the mean Jaccard index among all categories

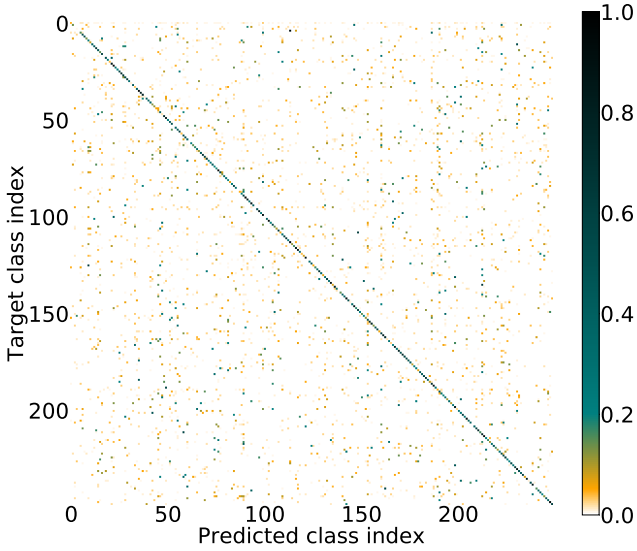

Figure 13. **ChaLearn ConGD** [28] confusion matrix indicating the classification performance of the deep neural network.

Round 2 (2017) [13]			
Rank	**Team**	**MJI Valid**	**MJI Test**
1	ICT_NHCI	0.5163	0.6103
2	AMRL	0.5957	0.5950
3	PaFiFA	0.3646	0.3744
4	Ours (RGB)	0.3190	0.3164
Round 1 (2016) [7]			
Rank	**Team**	**MJI**	**Method**
1	ICT_NHCI	0.2869	appearance model + RNN + RGB-D
2	TARDIS	0.2692	C3D + sliding window + RGB-D
3	AMRL	0.2655	QOM+CNN+depth
-	Baseline[28]	0.1464	MFSK

Table 1. ChaLearn LAP ConGD Challenge Round 1 [7] and 2 [13] final results. MJI: Mean Jaccard Index.

and sequences is computed:

$$J_{\text{mean}} = \frac{1}{N} \sum_{n=1}^{N} \sum_{s=1}^{S} \frac{J_{s,n}}{l_s}, \tag{4}$$

where $N = 249$ is the number of categories, S the number of sequences in the current set and l_s the number of gestures in sequence s.

Our model achieves a mean Jaccard index of 0.3164 on the test set. The comparison with other teams can be found in Table 1. The model is able to surpass all methods used in the first round without using depth information. The confusion matrix is depicted in Figure 13. Looking at the diagonal, we can see that there are quite a few similar gestures which are difficult to distinguish from one another, as well as classes with good accuracy.

4. Conclusion and Future Work

We showed in this paper that deep residual networks are capable of learning patterns in continuous gesture and sign language videos with virtually no preprocessing and with the use standard RGB cameras. Our models were evaluated on two different sign language corpora and the largest known gesture dataset. We observed a top-10 framewise accuracy of 73.3% with the Corpus NGT [4, 5] and 55.7% with the Corpus VGT [27]. We achieved a mean Jaccard index of 0.3164 with the ChaLearn LAP ConGD Challenge [28].

These results have a lot of room for improvement. We suspect a big increase in performance when using depth sensors. The disadvantage is that a lot of datasets or applications don't have depth maps available. Another accuracy

boost would be gained from unsupervised feature learning and/or pretrained weights from large image datasets. Also, improvements would be gained from the integration of a hand and arm tracking method. A last suggested addition would be to employ a language model in the SLR case, as nearby predicted glosses are often related.

5. Acknowledgments

We would like to thank NVIDIA Corporation for the donation of a GPU used for this research. The research leading to these results has received funding from the agency Flanders Innovation & Entrepreneurship (VLAIO).

References

[1] X. Chai, G. Li, Y. Lin, Z. Xu, Y. Tang, X. Chen, and M. Zhou. Sign language recognition and translation with kinect. In *IEEE Conf. on AFGR*, 2013.

[2] J. Charles, T. Pfister, M. Everingham, and A. Zisserman. Automatic and efficient human pose estimation for sign language videos. *International Journal of Computer Vision*, pages 1–21, 2013.

[3] D.-A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *International Conference on Learning Representations (ICLR)*, 2016.

[4] O. Crasborn, I. Zwitserlood, and J. Ros. The Corpus NGT. A digital open access corpus of movies and annotations of Sign Language of the Netherlands. *Centre for Language Studies, Radboud Universiteit Nijmegen. http://www.ru.nl/corpusngtukgp/*, 2008.

[5] O. A. Crasborn and I. Zwitserlood. The corpus ngt: an online corpus for professionals and laymen. In *Construction and Exploitation of Sign Language Corpora. 3rd Workshop on the Representation and Processing of Sign Languages (LREC)*, pages 44–49. ELDA, 2008.

[6] R. Cui, H. Liu, and C. Zhang. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. 2017.

[7] H. J. Escalante, V. Ponce-López, J. Wan, M. A. Riegler, B. Chen, A. Clapés, S. Escalera, I. Guyon, X. Baró, P. Halvorsen, et al. Chalearn joint contest on multimedia challenges beyond visual analysis: An overview. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 67–73. IEEE, 2016.

[8] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning precise timing with lstm recurrent networks. *The Journal of Machine Learning Research*, 3:115–143, 2003.

[9] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[12] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[13] W. Jun, S. Escalera, A. Gholamreza, H. J. Escalante, X. Baró, I. Guyon, M. Madadi, A. Juri, G. Jelena, L. Chi, and X. Yiliang. Results and analysis of chalearn lap multi-modal isolated and continuous gesture recognition, and real versus fake expressed emotions challenges. In *ICCV Workshops*, 2017.

[14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *ICLR 2015*, 2015.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998.

[16] J. Liu, B. Liu, S. Zhang, F. Yang, P. Yang, D. N. Metaxas, and C. Neidle. Non-manual grammatical marker recognition based on multi-scale, spatio-temporal analysis of head pose and facial expressions. *Image and Vision Computing*, 32(10):671–681, 2014.

[17] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[18] N. Neverova, C. Wolf, G. Taylor, and F. Nebout. Moddrop: adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(8):1692–1706, 2016.

[19] E.-J. Ong and R. Bowden. A boosted classifier tree for hand shape detection. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 889–894. IEEE, 2004.

[20] C. Oz and M. C. Leu. American sign language word recognition with a sensory glove using artificial neural networks. *Engineering Applications of Artificial Intelligence*, 24(7):1204–1213, 2011.

[21] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. *Asian Conference on Computer Vision (ACCV)*, 2014.

[22] 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*, Oktober 2016.

[23] L. Pigou, M. Van Herreweghe, and J. Dambre. Sign classification in sign language corpora with deep neural networks. *LREC Workshop on the Representation and Processing of Sign Languages*, May 2016.

[24] V. Pitsikalis, A. Katsamanis, S. Theodorakis, and P. Maragos. Multimodal gesture recognition via multiple hypotheses rescoring. In *Gesture Recognition*, pages 467–496. Springer, 2017.

[25] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[27] M. Van Herreweghe, M. Vermeerbergen, E. Demey, H. De Durpel, N. H., and S. Verstraete. Het Corpus VGT. Een digitaal open access corpus van videos and annotaties van Vlaamse Gebarentaal, ontwikkeld aan de Universiteit Gent ism KU Leuven. www.corpusvgt.be, 2015.

[28] J. Wan, Y. Zhao, S. Zhou, I. Guyon, S. Escalera, and S. Z. Li. Chalearn looking at people rgb-d isolated and continuous datasets for gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 56–64, 2016.

[29] R. Y. Wang and J. Popović. Real-time hand-tracking with a color glove. *ACM transactions on graphics (TOG)*, 28(3):63, 2009.

[30] D. Wu, L. Pigou, P.-J. Kindermans, N. Le, L. Shao, J. Dambre, and J.-M. Odobez. Deep dynamic neural networks for multimodal gesture segmentation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence: Multimodal Human Pose Recovery and Behavior Analysis SI*, 2016.