A Fully Convolutional Sequence Learning Approach for Cued Speech Recognition from Videos

Katerina Papadimitriou and Gerasimos Potamianos

Electrical and Computer Engineering Department, University of Thessaly, Volos 38221, Greece
aipapadimitriou@uth.gr , gpotam@ieee.org

***Abstract*—Cued Speech constitutes a sign-based communication variant for the speech and hearing impaired, which involves visual information from lip movements combined with hand positional and gestural cues. In this paper, we consider its automatic recognition in videos, introducing a deep sequence learning approach that consists of two separately trained components: an image learner based on convolutional neural networks (CNNs) and a fully convolutional encoder-decoder. Specifically, handshape and lip visual features extracted from a 3D-CNN feature learner, as well as hand position embeddings obtained by a 2D-CNN, are concatenated and fed to a time-depth separable (TDS) block structure, followed by a multi-step attention-based convolutional decoder for phoneme prediction. To our knowledge, this is the first work where recognition of cued speech is addressed using a common modeling approach based entirely on CNNs. The introduced model is evaluated on a French and a British English cued speech dataset in terms of phoneme error rate, and it is shown to significantly outperform alternative modeling approaches.**

***Index Terms*—Cued speech, 3D-CNN, TDS encoder, attention-based convolutional decoder**

I. Introduction

Speechreading constitutes a fundamental modality of speech perception among orally educated hearing-impaired people. However, the ambiguity of visual speech patterns renders speechreading inadequate in the absence of semantic content. For that reason, in 1967, Cornett [1] introduced Cued Speech (CS) by combining hand positional and gestural cues with mouthing patterns (see also Fig. 1). Thus, automatic recognition of CS necessitates the combined use of lipreading and sign language techniques [2], [3].

Lately, there has been increased interest in automatic CS recognition, due to the availability of publicly available corpora [4], [5] (see also Fig. 2). Early research has relied on artificial markings for addressing the problem of lip and hand segmentation [6], [7]. More recently [8], a novel scheme was proposed that abstains from the use of any visual artifices, employing instead the Kanade-Lucas-Tomasi lip feature tracker and a Gaussian mixture model (GMM)-based foreground extraction for hand region detection. Regarding speech modeling, the most dominant approaches map sequences of hand-crafted features to phonemes using hidden Markov models (HMMs) [6], [7]. More recently [8], [9], convolutional neural networks (CNNs) are employed to extract visual features from lip and hand regions, feeding them to an HMM-GMM classifier. A critical issue in CS is the asynchrony between hand and lip articulations, with most proposed approaches relying on audio-based segmentation [10]. For that purpose, a temporal segmentation scheme is used in [4], [5] to estimate the average hand preceding time for vowels and consonants, whereas in [9], a novel synchronization approach for multi-modal fusion is employed to align hand and lip features.

In this study, we focus on continuous CS recognition in videos with no artificial markings, regarding the CS recognition problem as an image-to-text translation task. Our approach contains three distinct pillars: feature extraction for the streams of interest (handshapes, lips, and hand position), multi-stream feature concatenation, and recognition. Thus, we introduce a visual multi-modal system based on a fully convolutional sequence model comprised of two main components that are trained separately: CNN-based stream feature learners and a fully convolutional encoder-decoder. The latter part composes of a time-depth separable (TDS) block structure [12] encoder, accompanied by a multi-step attention-based convolutional decoder [13], [14] with gated linear units [15] over the convolution output that are trained jointly for sequence prediction (see also Fig. 3). The model is complemented with an input feeding scheme, through which prior attentional vectors are concatenated with inputs at the following time step. Additionally, our approach abstains from using any explicit stream synchronization, applying instead direct fusion to the three feature flows, letting the model to learn such implicitly.

To the best of our knowledge, this paper constitutes the first attempt to combine a fully convolutional TDS encoder with a

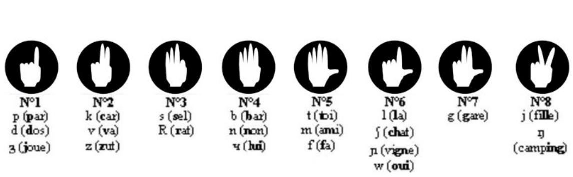


Fig. 1. French Cued Speech, showing hand positioning and shape (figure adapted from [11]).

This research work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant" (Project Number: 2456).

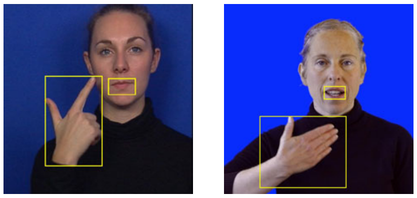

(a) (b)

Fig. 2. Example frames marked with rectangular boxes enclosing the detected handshape, as well as the mouth region from (a) the French cued speech dataset [8] and (b) the British English cued speech dataset [5].

multi-step attention-based convolutional decoder for automatic CS recognition. We evaluate the introduced approach on the French [8] and British English [5] CS datasets (see also Fig. 2). We compare our approach experimentally to four alternative sequence models under two additional feature learners, achieving significant absolute phoneme accuracy improvement of 9.38% in the French CS recognition task compared to previous approaches [8]. Notably also, to our knowledge, this represents the first attempt to address British English CS recognition.

The rest of the paper is organized as follows: Section II describes the basic pillars of the proposed model; Section III outlines the implementation details and the evaluated systems; Section IV presents the datasets and experiments; and Section V summarizes the paper.

II. Model

We consider a source sequence $x = (x_1, x_2, ..., x_m)$ expressed in m raw image frames that is processed generating a sequence of image features $z = (z_1, z_2, ..., z_m)$ and passes through an encoder-decoder module generating the predicted sequence $y = (y_1, y_2, .., y_n)$. As already discussed, the general architecture of the proposed system comprises two main phases (see Fig. 3): (i) a CNN-based image feature extractor, and (ii) a convolutional TDS encoder attended by an attention-based convolutional decoder for prediction, all detailed next.

A. 3D-CNN based feature learner

CNNs contain a series of convolutional layers complemented with non-linearity and pooling, followed by fully connected layers and an output layer. Here, we apply a pretrained 3D ResNet-34 [16] trained on the Kinetics dataset [17], to extract spatio-temporal features from a video clip containing adjacent frames (3 frames). The main difference between 3D ResNets and original ResNets is that the former employ 3D convolutions ($3 \times 3 \times 3$) and 3D pooling instead of 2D ones. All image frames are resized to 96×96 pixels, before being converted to clips and fed into the 3D ResNet-34 network. The network outputs 512-dimensional feature maps, by taking the output of the global average pooling layer.

B. Attention-based encoder-decoder

1) Time-depth separable convolutional encoder: Inspired by [12], for temporal modeling a TDS convolution block encoder is used. Such consists of a 2D convolution over time, followed by a fully-connected block. More precisely, the model initiates with a 2D convolutional layer, which is fed with an input of size $T \times w \times p$, with T being the number of time-steps, w the input width, and p the number of channels. The network uses $k \times 1$ convolutional kernels (kp^2 parameters) and downsampling with stride 2. The convolutional layer, which is complemented with a rectified linear unit (ReLU) non-linearity, is followed by a fully-connected layer composed of a sequence of two 1×1 convolutions with a stride of 2 complemented with ReLU non-linearity. The output of the convolutional block, before being fed to the fully-connected layer, is transformed into a shape of $T \times 1 \times wp$.

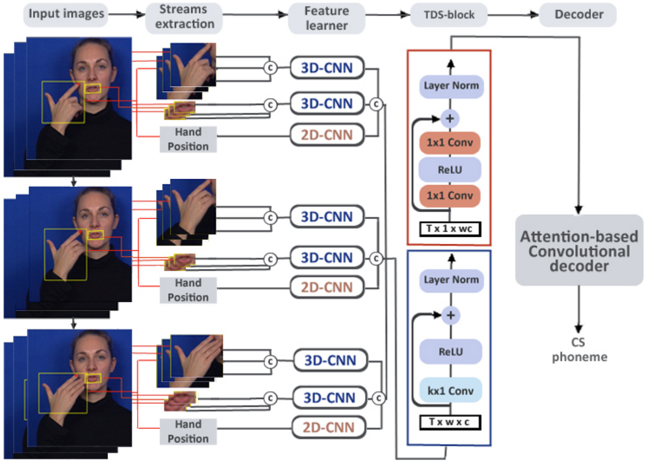


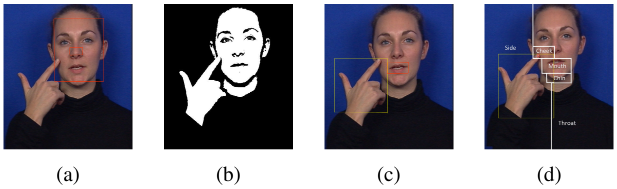
Fig. 3. Overview of the proposed model comprising hand- and lip-region extraction, as well as the hand position; a 3D-CNN/2D-CNN image feature learner (circled C denotes concatenation); a TDS block structure-based encoder; and an attention-based convolutional decoder for phoneme prediction.

To smoothly optimize and leverage the performance of the model, residual connections [18] and layer normalization over all dimensions are added after the convolutional and the fully-connected block. Since the output of each TDS layer is compressed in time, we increase the number of output channels of each convolutional layer by multiplying it with a factor equal to the input feature dimension divided by the number of channels of input features. Finally, dropout is applied after the ReLU non-linearity in each layer.

2) Multi-step attention-based convolutional decoder: As in [13], [14], each decoder layer composed of one-dimensional convolution is followed by a gated linear unit (GLU) [15] that operates as a gating tool for dealing with the convolution output $H = [AB] \in \mathbb{R}^{2D}$. For that purpose we use $u = A \otimes \sigma(B)$, where $u \in \mathbb{R}^D$ expresses which of A outputs associate with the current target element, and $\otimes$ denotes pointwise multiplication. For simplicity, we denote that the l-th decoder layer generates $d^l = (d^l_1, d^l_2, ..., d^l_n)$ hidden states.

Since for a one-layer decoder of kernel width k each generated hidden state d^1_j is related to k inputs, stacking multiple layers on top of each other results in states that are related to more inputs than previously. In more detail, each convolutional kernel $K \in \mathbb{R}^{2D \times kD}$ is fed with k concatenated input elements embedded in D dimensions ($Z \in \mathbb{R}^{k \times D}$) returning $H \in \mathbb{R}^{2D}$ with twice dimensionality than input elements, since layers process k outputs of the previous ones.

Additionally, the multi-step attention mechanism [13], [19]



(a) (b) (c) (d)

Fig. 4. Example of the preprocessing pipeline: (a) Input image marked with a rectangular box enclosing the detected face and nose region; (b) segmented skin region; (c) resulting image with the yellow rectangular box illustrating the moving object (hand) and the red box the lip region; (d) hand position detection through mouth and face location-based area division (white lines) and upper hand point coordinates deduction (red circle).

is an integral part of a fully convolutional decoder. More precisely, the attention weights a_{ij}^l, which are applied to h_i during decoding, are computed using a score function normalized by softmax. For that purpose, a variety of alignment functions have been proposed in the literature [20], [21]. Here, we apply a dot alignment function on the corresponding decoder layer d_j^l and encoder state h_i. The context vectors c_j^l are computed as the weighted sum of each encoder hidden state:

$$c_j^l = \sum_{i=1}^{m} a_{ij}^l h_i.$$

Subsequently, the context vectors are fed into the next decoder layer providing specific information for attention calculation. After last layer's context vector c_j^l computation and concatenation with the decoder hidden state, the attentional vector $\tilde{d}_t$ is generated:

$$\tilde{d}_t^l = \tanh(W_c[c_t^l; d_t^l]).$$

Finally, we complement the model with an input feeding mechanism [21], where we concatenate the attentional vectors $\tilde{d}_t^l$ with each layer decoder inputs at the following time step, converting the model to a fully connected deep neural network on both directions that deploys previous alignment information during the estimation of the new ones.

III. Implementation Details and Systems

A. *Preprocessing pipeline for CS recognition*

Our system is complemented with an image preprocessing pipeline, inspired by own prior work [22], for hand and lip region extraction, as well as hand position localization. In addition, the system is equipped with stream fusion of the three visual inputs (handshape, lips, and hand position).

1) Hand and lip detection: The primary step of the proposed system constitutes an image processing pipeline for hand and lip region extraction (see Fig. 4(a)-(c)). The pipeline initiates with nose area detection, which captures uniquely the skin color range, through the use of the Viola-Jones algorithm [23]. Subsequently for hand detection, the skin-tone information drives skin region segmentation in the YCbCr color space [24]. To treat overlap between the hand and face area, after skin-like pixel segmentation and since hands are moving objects in the field-of-view, we perform hand tracking by means of motion-based Kalman filtering [25]. In parallel, the lip area is extracted in each image frame through a cascade object detector using the Viola-Jones algorithm [23].

2) Hand position detection and representation: The preprocessing phase is complemented with hand configuration classification into 5 different positions (side, mouth, chin, cheek, throat) for French CS and 4 positions (side, mouth, chin, throat) for the British English CS case. Specifically, we perform hand edge detection through the Sobel operator [26], extracting the coordinates of the upper point of the hand region. Then, the input image is divided into several regions in accordance with the lips position, and a label is assigned based on the hand position (see Fig. 4(d)). Hand relative positions are learned via a five-layer 2D-CNN with three fully-connected layers resulting in a 64-dimensional feature vector.

3) Stream fusion: The features of each visual stream (handshape and lips) extracted from the 3D-CNN feature learner, as well as the position embeddings are concatenated generating a 1,088-dimensional feature vector (512 for handshape representation, 512 for the lip region, and 64 for the hand position embeddings) and subsequently fed to the proposed attention-based encoder-decoder for phoneme sequence prediction.

B. *System details*

We compare our approach experimentally to four alternative sequence models under two additional feature learners examining the effect of our architecture on CS recognition. All models were implemented in PyTorch [27], and their training was carried out using GPU acceleration.

1) Evaluated feature learners: For image feature extraction in our model we employ a 3D ResNet-34 with 3-frame video clips, trained through stochastic gradient descent with momentum at 0.9 with an initial learning rate of 0.1 (decayed by a factor of 0.001), performing 500 complete passes over the data. A mini-batch size of 256 images was employed.

Additionally, for comparison we use a Vanilla auto-encoder (AE) image feature learner based on MLP consisting of two hidden layers with fixed dimensionality at 100 on the encoder and the decoder, respectively. For weight initialization we performed the Xavier process [28]. AE network training was conducted using scaled conjugate gradient descent (SCGD) with an initial learning rate of 0.004 decreased by a factor of 0.8 and a mini-batch size of 64 images.

Moreover, we employ a 2D-CNN image feature learner based on a pretrained ResNet-18 network [29] (trained on the ImageNet database [30]) in order to extract feature maps by taking the output of the global average pooling layer. The network uses 3 × 3 convolutional kernels and downsampling with stride 2. The 2D-CNN transforms the image sequence into a sequence of 512-dimensional feature vectors employing the mean squared error loss function.

2) Sequence modeling schemes: The following are used:

TDS enc & attention-based CNN dec (TDS/CNN): The TDS encoder has one 10-channel and one 128-channel TDS blocks with kernel sizes 3×1 and 5×1, respectively. For the convolutional attention-based decoder, the size of hidden states is fixed at 128. The multi-step attention-based convolutional decoder comprises of a 6-layer decoder with kernel width 5. Training is carried out employing the Adagrad optimizer [31]

TABLE I
Performance comparison in PER (%) of the evaluated sequence models under all feature learners on both datasets.

Dataset		French CS					British English CS				
Features	Models	TDS/CNN	TDS/GRU	ARNN	ACNN	Transformer	TDS/CNN	TDS/GRU	ARNN	ACNN	Transformer
AE		42.03	42.51	50.06	44.94	54.17	52.81	59.21	61.04	56.07	67.22
2D-CNN		40.16	45.38	45.04	44.55	47.47	45.31	50.11	52.10	47.29	59.11
3D-CNN		**29.12**	31.28	33.35	31.20	37.47	**36.25**	37.09	43.57	36.84	39.77

with an initial learning rate of 0.3, decreased by a factor of 0.3. Dropout is added at a rate of 0.4. The beam search strategy [32] with beam width of 2 in decoding is applied. Finally, the mini-batch size is fixed to 128.

TDS enc & attention-based GRU dec (TDS/GRU): The model comprises of a 2-layer GRU decoder with 128 hidden units. Training is conducted employing the Adam optimizer [33] with an initial learning rate of 0.001 decreased by a factor of 3.0. Attention score calculation is carried out by the dot alignment function [21].

Attentional RNN enc-dec (ARNN): The model constitutes a 3-layer LSTM [34] encoder-decoder with 128 hidden units. Training is conducted employing the Adam optimizer [33] with an initial learning rate of 0.015 decreased by a factor of 2.0. Attention score calculation is carried out by the dot alignment function [21].

Attentional CNN enc-dec (ACNN): The model constitutes a 6-layer CNN encoder-decoder with kernel width 5 and 128 hidden units. Training is conducted employing the Adagrad optimizer [31] with an initial learning rate of 0.3 decreased by a factor of 0.3. Attention score calculation is carried out by the dot alignment function [21] and the model is complemented with an input feeding scheme.

Transformer enc-dec (Transformer): Sequences are fed to a 6-layer transformer with 8 heads for transformer self-attention and 2048-dimension hidden transformer feed-forward. Training is conducted employing the Adam optimizer [33] with an initial learning rate of 0.001 decreased by a factor of 2.0. Parameter initialization follows the Xavier process [28].

IV. Experiments

A. Datasets and experimental framework

The French CS dataset [8] contains 2 repetitions of 238 French sentences expressed by a professional CS interpreter consisting of about 11,770 phonemes totally. RGB video images including the interpreter's upper body are available at 50 fps and 720×576 pixel resolution. Note that French CS encapsulates 8 lip patterns, 8 handshapes, and 5 different hand positions, encoding a set of 34 phonetic classes, namely 14 vowels and 20 consonants.

The British English CS dataset [5] is recorded by a professional CS interpreter and contains 97 British English sentences. Color video images of the interpreter's upper body are available at 25 fps, with a spatial resolution of 720×1280. Note that British English CS encapsulates 4 hand positions for encoding the 12 monophthongs and 4 hand slips for encoding the 8 diphthongs, while 8 hand shapes are used to encode the 24 consonants.

Both datasets are randomly partitioned into 10 equal sized subsets, with 80% of the data being used for training, 10% for validation, and 10% for testing in each subset.

B. Results

All models are evaluated in phoneme error rate (PER) (%). As demonstrated in Table I, the proposed model yields the lowest PERs on both datasets achieving 29.12% for French CS (improving over the prior published results without synchronization techniques obtained in [8] by 9.38% PER, absolute) and 36.25% for British English CS. As it may be observed there is a significant difference between the PER performance on the two datasets. This is primarily due to the limited size of the British English CS dataset. Notably, the 3D-CNN based feature learner yields consistent improvement over all models for both datasets as compared to the other alternatives providing a more explicit discrimination of the extracted features. Finally, the attentional CNN encoder-decoder outperforms the other sequence models, but lags our model.

C. Model Variations

Table II reports PER results of a number of model variations regarding the feature learners, the number of TDS blocks, and the number of adjacent frames used by the feature learner. Specifically, we examined the performance of the proposed model under other baseline 3D-CNNs like ResNet-10 and ResNet-101, but we ended with worse PERs. It should be noted that a 2D-CNN image feature learner based on the Alexnet architecture provides lower performance by a significant margin of almost 9%. We also reduced the number of channels from $(10, 128)$ to $(10, 18)$ and $(10, 14)$ obtaining worse PERs by at least 3%. Moreover, we increased the number of TDS blocks from two to three without a meaningful improvement in performance. Additionally, the impact of using concatenated adjacent 2D-CNN context feature vectors instead

TABLE II
Comparative evaluation in PER (%) of a number of model variations with Nb being the number of TDS blocks and Nf the number of adjacent frames.

Model details			Cued speech datasets	
Feature learner	Nb	Nf	French	British
2D-CNN	2		38.52	44.27
2D-CNN Conc.	2	3	32.49	38.96
3D ResNet-10	2	3	30.85	37.41
3D ResNet-101	2	3	31.02	38.47
3D ResNet-34	3	3	29.76	37.34
3D ResNet-34	4	3	29.91	38.23
Automatic synchronization			30.96	38.39
Handshape & lips only			39.56	44.28

of 3D-CNN features on the model performance was evaluated. As it may be observed in Table II the proposed model with a 3D-CNN image feature learner turns out superior on both evaluation datasets.

As already mentioned, there exists asynchrony between hand and lip articulations with the hand generally preceding lips by approximately one syllable [4]. For that purpose, we evaluated the model performance using an automatic alignment based on the delay between the two streams, namely a 12-frame [11] and 15-frame delay for French CS and British English CS dataset, respectively. The proposed model with direct feature fusion yields lower PER on both evaluation sets. This is due to the better generalization ability of the proposed sequence learning approach. Finally, in order to demonstrate the importance of the hand position embeddings to CS, we evaluated our model performance by removing that third stream in our feature fusion module. That resulted in significant PER degradation, demonstrating the importance of the hand position in CS.

V. Conclusion

In this paper we propose a sequence learning model for effective CS recognition involving two principal phases, a 3D-CNN based feature learner followed by a fully convolutional TDS encoder and a multi-step attention-based convolutional decoder. We highlighted how the incorporation of the 3D ResNet-34 feature extractor improves the feature learning and, by extension, the fully convolutional sequence model performance. The performance comparative evaluation on two CS datasets demonstrated that the proposed model generalizes much better than other sequence learning architectures.

References

[1] R. O. Cornett, "Cued speech," *American Annals of the Deaf*, vol. 112, no. 1, pp. 3–13, 1967.

[2] G. A. Rao, K. Syamala, P. V. V. Kishore, and A. S. C. S. Sastry, "Deep convolutional neural networks for sign language recognition," in *Proc. Signal Process. and Commun. Engin. Systems*, 2018, pp. 194–197.

[3] V. Bheda and D. Radpour, "Using deep convolutional networks for gesture recognition in American Sign Language," *CoRR*, vol. abs/1710.06836, 2017.

[4] L. Liu, G. Feng, and D. Beautemps, "Automatic temporal segmentation of hand movements for hand positions recognition in French cued speech," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 3061–3065.

[5] L. Liu, J. Li, G. Feng, and X. Zhang, "Automatic Detection of the Temporal Segmentation of Hand Movements in British English Cued Speech," in *Proc. Interspeech*, 2019, pp. 2285–2289.

[6] P. Heracleous, D. Beautemps, and N. Aboutabit, "Cued speech automatic recognition in normal-hearing and deaf subjects," *Speech Communication*, vol. 52, no. 6, pp. 504–512, 2010.

[7] P. Heracleous, D. Beautemps, and N. Hagita, "Continuous phoneme recognition in cued speech for French," in *Proc. European Signal Processing Conference*, 2012, pp. 2090–2093.

[8] L. Liu, T. Hueber, G. Feng, and D. Beautemps, "Visual recognition of continuous cued speech using a tandem CNN-HMM approach," in *Proc. Interspeech*, 2018, pp. 2643–2647.

[9] L. Liu, G. Feng, D. Beautemps, and X. Zhang, "A new re-synchronization method based multi-modal fusion for automatic continuous cued speech recognition," *CoRR*, vol. abs/2001.00854, 2020.

[10] S. E. Tranter, K. Yu, G. Everinann, and P. C. Woodland, "Generating and evaluating segmentations for automatic speech recognition of conversational telephone speech," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 2004, pp. 753–756.

[11] V. Attina, D. Beautemps, M. Cathiard, and M. Odisio, "A pilot study of temporal organization in cued speech production of French syllables: rules for a cued speech synthesizer," *Speech Communication*, vol. 44, no. 1, pp. 197–214, 2004.

[12] A. Hannun, A. Lee, Q. Xu, and R. Collobert, "Sequence-to-sequence speech recognition with time-depth separable convolutions," *CoRR*, vol. abs/1904.02619, 2019.

[13] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. International Conference on Machine Learning*, 2017.

[14] K. Papadimitriou and G. Potamianos, "End-to-end convolutional sequence learning for ASL fingerspelling recognition," in *Proc. Interspeech*, 2019, pp. 2315–2319.

[15] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. International Conference on Machine Learning*, 2017, pp. 933–941.

[16] K. Hara, H. Kataoka, and Y. Satoh, "Learning spatio-temporal features with 3D residual networks for action recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3154–3160.

[17] J. Carreira and A. Zisserman, "Quo Vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4724–4733.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[19] S. Chollampatt and H. T. Ng, "A multilayer convolutional encoder-decoder neural network for grammatical error correction," in *Proc. AAAI Conference on Artificial Intelligence*, 2018.

[20] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, 2014, arXiv:abs/1409.0473v7.

[21] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015, pp. 1412–1421.

[22] K. Papadimitriou and G. Potamianos, "A hybrid approach to hand detection and type classification in upper-body videos," in *Proc. European Workshop on Visual Information Processing*, 2018.

[23] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001.

[24] K. B. Shaik, P. Ganesan, V. Kalist, B. Sathish, and J. M. M. Jenitha, "Comparative study of skin color detection and segmentation in HSV and YCbCr color space," *Procedia Computer Science*, vol. 57, pp. 41–48, 2015.

[25] J. Jeong, T. Yoon, and J. Park, "Kalman filter based multiple objects detection-tracking algorithm robust to occlusion," in *Proc. SICE Annual Conference*, 2014, pp. 941–946.

[26] I. Sobel and G. Feldman, "A 3×3 isotropic gradient operator for image processing," *Pattern Classification and Scene Analysis*, pp. 271–272, 1973.

[27] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Proc. NIPS-W*, 2017.

[28] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. International Conference on Artificial Intelligence and Statistics*, vol. 9, 2010, pp. 249–256.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[30] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[31] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[32] M. Freitag and Y. Al-Onaizan, "Beam search strategies for neural machine translation," in *Proc. Workshop on Neural Machine Translation*, 2017, pp. 56–60.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, 2014, arXiv:abs/1412.6980v9.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computing*, vol. 9, no. 8, pp. 1735–1780, 1997.