applied sciences

MDPI

Article

Neural Sign Language Translation Based on Human Keypoint Estimation

Sang-Ki Ko *,†, Chang Jo Kim †, Hyedong Jung and Choongsang Cho

Korea Electronics Technology Institute, Seongnam 13488, Korea

* Correspondence: sangkiko@keti.re.kr; Tel.: +82-31-739-7534

† Two authors contributed equally to this work.

Received: 29 March 2019; Accepted: 24 June 2019; Published: 1 July 2019

Abstract: We propose a sign language translation system based on human keypoint estimation. It is well-known that many problems in the field of computer vision require a massive dataset to train deep neural network models. The situation is even worse when it comes to the sign language translation problem as it is far more difficult to collect high-quality training data. In this paper, we introduce the KETI (Korea Electronics Technology Institute) sign language dataset, which consists of 14,672 videos of high resolution and quality. Considering the fact that each country has a different and unique sign language, the KETI sign language dataset can be the starting point for further research on the Korean sign language translation. Using the KETI sign language dataset, we develop a neural network model for translating sign videos into natural language sentences by utilizing the human keypoints extracted from the face, hands, and body parts. The obtained human keypoint vector is normalized by the mean and standard deviation of the keypoints and used as input to our translation model based on the sequence-to-sequence architecture. As a result, we show that our approach is robust even when the size of the training data is not sufficient. Our translation model achieved 93.28% (55.28%, respectively) translation accuracy on the validation set (test set, respectively) for 105 sentences that can be used in emergency situations. We compared several types of our neural sign translation models based on different attention mechanisms in terms of classical metrics for measuring the translation performance.

Keywords: sign language translation; human keypoint detection; deep learning; sequence-to-sequence model

1. Introduction

The absence of the ability to hear sounds is a huge obstacle to smooth and natural communication for the hearing-impaired people in a predominantly hearing world. In many social situations, the hearing-impaired people necessarily need help from professional sign language interpreters to communicate with the hearing people even when they have to reveal their very private and sensitive information. Moreover, the hearing-impaired people are more vulnerable in various emergency situations due to the communication barriers due to the absence of the hearing ability. As a consequence, the hearing-impaired people easily become isolated and withdrawn from society. This leads us to investigate the possibility of developing an artificial intelligence technology that understands and communicates with the hearing-impaired people.

However, sign language recognition or translation is a very challenging problem since the task involves an interpretation between visual and linguistic information. The visual information consists of several parts such as body movement and facial expression of a signer [1,2]. Interpreting the collection of the visual information as natural language sentences is also one of the tough challenges to realize the sign language translation problem.

To process a sequence, there have been several interesting variants of recurrent neural networks (RNNs) proposed including long short-term memory (LSTM) [3] and gated recurrent units (GRUs) [4]. These architectures have been successfully employed to resolve many problems involving the process of sequential data such as machine translation and image captioning [5–8]. Moreover, many researchers working on the field of image and video understanding have raised the level that seemed infeasible even a few years ago by learning their neural networks with a massive amount of training data. Recently, many neural network models based on convolutional neural network (CNNs) exhibited excellent performances in various visual tasks such as image classification [9,10], object detection [11,12], semantic segmentation [13,14], and action recognition [15,16].

Understanding sign languages requires a high level of spatial and temporal understanding and, therefore, is regarded as very difficult with the current level of computer vision and machine learning technology [1,17–22]. It should be noted that sign languages are different from hand (finger) languages as the hand languages only represent each letter in an alphabet with the shape of a single hand [23] while the linguistic meaning of each sign is determined by subtle difference of shape and movement of body, hands, and sometimes by facial expression of the signer [2]. More importantly, the main difficulty comes from the lack of dataset for training neural networks. Many sign languages represent different words and sentences of spoken languages with temporal sequences of gestures comprising continuous pose of hands and facial expressions. This implies that there are uncountably many combinations of the cases even to describe a single human intention with the sign language.

Hence, we restrict ourselves to the task of translating sign language in various emergency situations. We construct the first Korean sign language dataset collected from fourteen professional signers who are actually hearing-impaired people and named it the KETI sign language dataset. The KETI sign language dataset consists of 14,672 high-resolution videos that recorded the Korean signs corresponding to 419 words and 105 sentences related to various emergency situations. Using the KETI sign language dataset, we present our sign language translation model based on the well-known off-the-shelf human keypoint detector and the sequence-to-sequence translation model. To the best of our knowledge, this paper is the first to exploit the human keypoints for the sign language translation problem. Due to the inherent complexity of the dataset and the problem, we present an effective normalization technique for the extracted human keypoints to be used in the sign language translation. We implement the proposed ideas and conduct various experiments to verify the performance of the ideas with the test dataset.

The main contributions of this paper are highlighted as follows:

1. We introduce the first large-scale Korean sign language dataset for sign language translation.
2. We propose a sign language translation system based on the 2D coordinates of human keypoints estimated from sign videos.
3. We present an effective normalization technique for preprocessing the 2D coordinates of human keypoints.
4. We verify the proposed idea by conducting various experiments with the sign language dataset.

2. Related Work

There have been many approaches to recognize hand languages that are used to describe letters of the alphabet with a single hand. It is relatively easier than recognizing sign languages as each letter of the alphabet simply corresponds to a unique hand shape. In [17], the authors utilized depth cameras (Microsoft's Kinect) and the random forest algorithm to recognize the English alphabet with 92% recognition accuracy. A pose estimation method of the upper body represented by seven key points was proposed for recognizing the American Sign Language (ASL) [18]. We also note t an approach by Kim et al. [24] to recognize the Korean hand language by analyzing latent features of hand images.

In general, researchers rely on the movements and shapes of both hands to recognize sign languages. Starner et al. [22] developed a real-time system based on Hidden Markov model (HMM) to recognize sentence-level ASL. They demonstrated two experimental results: they used solidly

colored gloves to make tracking of hands easier in the first experiment and the second experiment was conducted without gloves. They claimed that the word accuracy of glove-based system is 99.2% but the accuracy drops to 84.7% if they do not use gloves. It should be noted that those accuracy can be reached because they exploited the grammar for reviewing the errors of the recognition. The word accuracy without grammar and gloves is 74.5%.

On the other hand, there have been approaches to automatically learning signs from weakly annotated data such as TV broadcasts by using subtitles provided simultaneously with the signs [25–27]. Following this direction, Forster et al. released the RWTH-PHOENIX-Weather 2012 [1] and its extended version RWTH-PHOENIX-Weather 2014 [28] that consist of weather forecasts recorded from German public TV and manually annotated using glosses and natural language sentences where time boundaries have been marked on the gloss and the sentence level. Based on the RWTH-PHOENIX-Weather corpus, Koller et al. [20] presented a statistical approach performing large vocabulary continuous sign language recognition across different signers. They developed a continuous sign language recognition system that utilizes multiple information streams including the hand shape, orientation and position, the upper body pose, and face expressions such as mouthing, eyebrows and eye gaze.

There have been many attempts to recognize and translate sign language using deep learning (DL). Oberweger et al. [29] introduced and evaluated several architectures for CNNs to predict the 3D joint locations of a hand given a depth map. Kishore et al. [19] developed a sign language recognition system that is robust in different video backgrounds by extracting signers using boundary and prior shape information. Then, the feature vector is constructed from the segmented signer and used as input to artificial neural network. An end-to-end sequence modeling using CNN-BiLSTM architecture usually used for gesture recognition was proposed for large vocabulary sign language recognition with RWTH-PHOENIX-Weather 2014 [21].

At the same time, one of the most interesting breakthroughs in neural machine translation or even in the entire DL was introduced under the name of "sequence-to-sequence (seq2seq)" [7]. The seq2seq model relies on a common framework called an encoder–decoder model with RNN cells such as LSTMs or GRUs. The seq2seq model proved its effectiveness in many sequence generation tasks by achieving almost human-level performance [7]. Despite its effectiveness, the seq2seq model still has some drawbacks such as the input sequences of varying lengths being represented in fixed-size vectors and the vanishing gradient due to the long-term dependency between distant parts.

Camgoz et al. [23] formalized the sign language translation problem based on the pre-existing framework of neural machine translation with word and spatial embeddings for target sequences and sign videos, respectively. They proposed to utilize the seq2seq models to learn how to translate the spatiotemproal representation of signs into the spoken or written language. Recently, researchers developed a simple sign language recognition system based on bidirectional GRUs which just classifies a given sign language video into one of the classes that are predetermined [30].

3. KETI Sign Language Dataset

The KETI dataset was constructed to understand the Korean sign language of hearing-impaired people in various emergency situations. Indeed, in many social situations, the hearing-impaired people necessarily need help from professional sign language interpreters to communicate with the hearing people even when they have to reveal their very private and sensitive information. Moreover, the hearing-impaired people are more vulnerable in various emergency situations due to the communication barriers due to the absence of the hearing ability. Therefore, we carefully examined the cases of relatively general conversations about emergency situations and chose 105 sentences and 419 words that could be used in various emergency situations.

The KETI sign language dataset consists of 14,672 full high definition (HD) videos that were recorded at 30 frames per second and from two camera angles: front and side. We recorded 524 different signs derived from the aforementioned process and performed by fourteen different hearing-impaired

signers to reflect the individual differences for the same sign. For each sign, we first record a "guide video" of an "expert" signer to remove the possible ambiguity of signs. After watching the guide video, the fourteen hearing-impaired signers recorded each of the 524 signs. As a result, each signer recorded a total of 1048 videos for the dataset. For the training and validation sets, we chose ten signers from fourteen signers and chose nine sign videos for each sign for the training set. The remaining sign videos were assigned to the validation set. The test set consists of sign videos of four signers who do not appear in any video in the training set or the validation set. Several statistics of the dataset are given in Table 1 and an example frame from the dataset is presented in Figure 1. We also present ten example frames that were extracted from sign videos of ten different signers in Figure 2.

Table 1. Statistics of KETI sign language dataset.

Metric	Training	Dev	Test
Number of sign videos	9432	1048	4192
Duration [hours]	20.05	2.24	5.70
Number of frames	2,165,682	241,432	615,486
Number of signers	10	10	4
Number of camera angles		2	

In particular, we annotated each of the 105 signs that correspond to the useful sentences in emergencies mentioned above with five different natural language sentences in Korean. Moreover, we annotated all sign videos with the corresponding sequences of *glosses* [31], where a gloss is a unique word that corresponds to a unit sign and used to transcribe sign language. For instance, a sign implying "I am burned." can be annotated with the following sequence of glosses: ("FIRE", "SCAR"). Similarly, a sentence "A house is on fire." is annotated by ("HOUSE", "FIRE"). Apparently, glosses are more appropriate to annotate a sign because it is possible to be expressed in various natural sentences or words with the same meaning. For this reason, we annotated all signs with the glosses with the help of Korean sign language experts. Table 2 exhibits ten examples from 105 data examples in total.

Table 2. Ten examples of our sign language annotations. We annotated each sign with five natural language sentences in Korean and a unique sign gloss. We only provide two sentences in the table due to space limitations.

ID	Korean Sentence	English Sentence	Sign Gloss
1	화상을 입었어요.	I got burned.	FIRE SCAR
2	폭탄이 터졌어요.	The bomb went off.	BOMB
3	친구가 숨을 쉬지 않아요.	My friend is not breathing.	FRIEND BREATHE CANT
4	집이 흔들려요.	The house is shaking.	HOUSE SHAKE
5	집에 불이 났어요.	The house is on fire.	HOUSE FIRE
6	가스가 새고 있어요.	Gas is leaking.	GAS BROKEN FLOW
7	112에 신고해주세요.	Please call 112.	112 REPORT PLEASE
8	도와주세요.	Help me.	HELP PLEASE
9	너무 아파요.	It hurts so much.	SICK
10	무릎 인대를 다친 것 같아요.	I hurt my knee ligament.	KNEE LIGAMENT SCAR

For the communication with hearing-impaired people in the situations, the KETI dataset was used to develop an artificial intelligence-based sign language recognizer or translator. All videos were recorded in a blue screen studio to minimize any undesired influence and learn how to recognize or translate the signs with an insufficient amount of data.

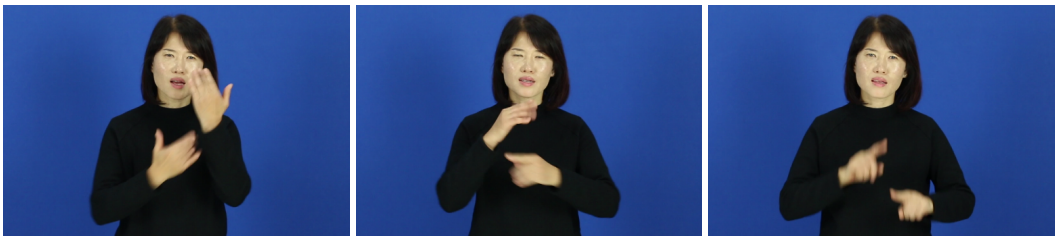

Figure 1. Example frames from our sign language dataset. The frames are extracted in a temporal order from the video for the sentence "I am burned".

Figure 2. Ten example frames from from our sign language dataset. Each frame was extracted from a sign video of a distinct signer.

4. Our Approach

We propose a sign recognition system based on the human keypoints that are estimated by pre-existing libraries such as OpenPose [32–34]. In Figure 3, we provide example results of human keypoint detection by the OpenPose for ten example frames presented in Figure 2. Here, we develop our system based on OpenPose, an open source toolkit for real-time multi-person keypoint detection. OpenPose can estimate in total 137 keypoints where 25 keypoints are from body pose, 21 keypoints are from each hand, and 70 keypoints from a face. The primary reason for choosing OpenPose as a feature extractor for sign language recognition is that it is robust to many types of variations.

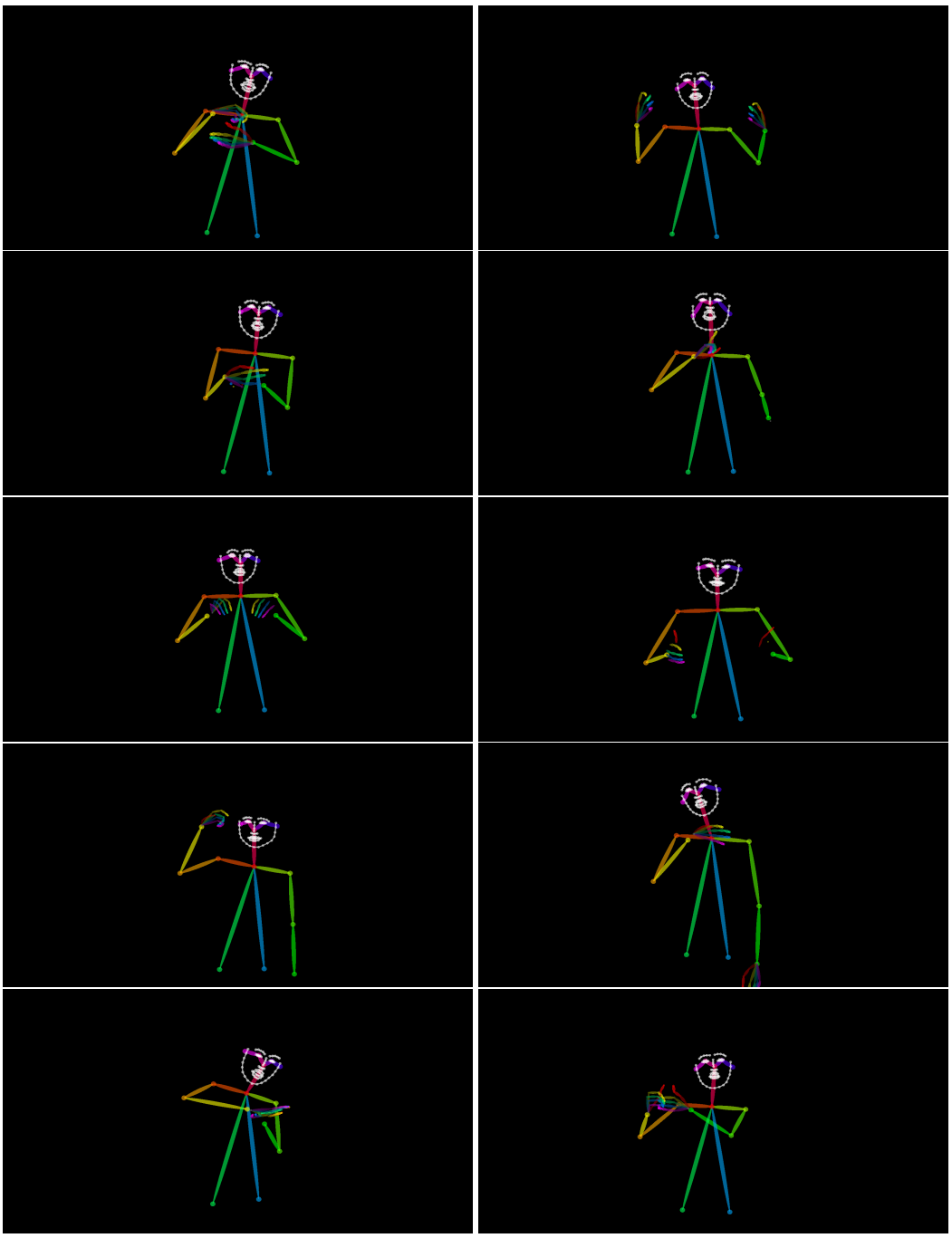

Figure 3. Keypoints extracted from example frames in Figure 2.

We use the estimated coordinates of 124 keypoints of a signer to understand the sign language of the signer, where 12 keypoints are from human body, 21 keypoints are from each hand, and 70 keypoints are from the face. Note that the number of keypoints from human body is 25 but we select 12 keypoints that correspond to upper body parts. The chosen indices and the names of the parts are as follows: 0 (nose), 1 (neck), 2 (right shoulder), 3 (right elbow), 4 (right wrist), 5 (left shoulder), 6 (left elbow), 7 (left wrist), 15 (right eye), 16 (left eye), 17 (right ear), and 18 (left ear).

4.1. Human Keypoint Detection by OpenPose

First, our recognition system is expected to be robust in different cluttered backgrounds as it only detects the human body. Second, the system based on the human keypoint detection works well regardless of signer since the variance of extracted keypoints are negligible. Moreover, we apply the feature normalization technique to further reduce the variance, which is dependent on signer. Third, our system can enjoy the benefits of the improvement on the keypoint detection system which has a great potential in the future because of its versatility. For instance, the human keypoint detection system can be used for recognizing different human behaviors and actions given that the relevant dataset is secured. Lastly, the use of high level features is necessary when the scale of the dataset is not large enough. In the case of sign language dataset, it is more difficult to collect than the other dataset as many professional signers should be utilized for recording sign language videos of high quality. The overall architecture of the proposed system is depicted in Figure 4.

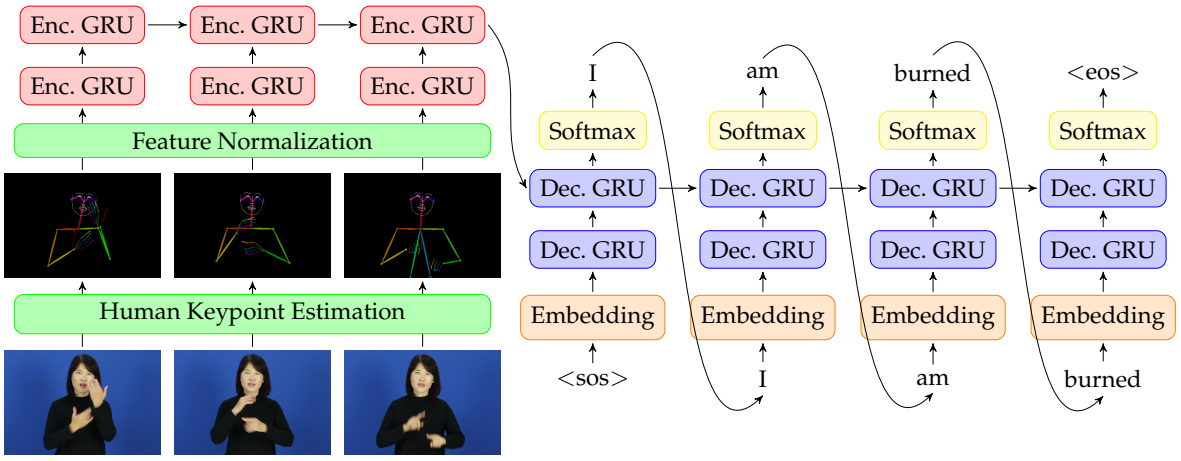

Figure 4. An overall architecture of our approach that translates a sign language video into a natural language sentence using sequence to sequence model based on GRU cells.

4.2. Feature Vector Normalization

There have been many successful attempts to employ various types of normalization methods in order to achieve the stability and speed-up of the training process [35–37]. One of the main difficulties in sign language translation with the small dataset is the large visual variance as the same sign can look very different depending on the signer. Even if we utilize the feature vector which is obtained by estimating the keypoints of human body, the absolute positions of the keypoints or the scale of the body parts in the frame can be very different. For this reason, we apply a special normalization method called the *object 2D normalization* that suits our purpose well.

After extracting high-level human keypoints, we normalize the feature vector using the mean and standard deviation of the vector to reduce the variance of the data. Let us denote a 2D feature vector by $V = (v_1, v_2, \ldots, v_n) \in \mathbb{N}^{n \times 2}$ that consists of n elements where each element $v_i \in \mathbb{N}^2$, $1 \leq i \leq n$ stands for a single keypoint of human part. Each element $v_i = (v_i^x, v_i^y)$ consists of two integers v_i^x and

v_i^y that imply the x- and y-coordinates of the keypoint v_i in the video frame, respectively. From the given feature vector V, we can extract the two feature vectors as follows:

$$V_x = (v_1^x, v_2^x, \ldots, v_n^x) \text{ and } V_y = (v_1^y, v_2^y, \ldots, v_n^y).$$

Simply speaking, we collect the x- and y-coordinates of keypoints separately while keeping the order. Then, we normalize the x-coordinate vector V_x as follows:

$$V_x^* = \frac{V_x - \bar{V_x}}{\sigma(V_x)},$$

where $\bar{V_x}$ is the mean of V_x and $\sigma(V_x)$ is the standard deviation of V_x. Note that V_y^* is calculated analogously. Finally, it remains to concatenate the two normalized vectors to form the final feature vector $V^* = [V_x^*; V_y^*] \in \mathbb{N}^{2n}$, which is used as the input vector of our neural network.

It should be noted that we assume that the keypoints of lower body parts are not necessary for sign language recognition. Therefore, we only use 124 keypoints from the 137 keypoints detected by OpenPose since six keypoints of human pose correspond to lower body parts such as both feet, knees and pelvises, as shown in Figure 2. We randomly sample 10–50 keyframes from each sign video. Hence, the dimension of input feature vector is $248 \times |V|$, where $|V| \in \{10, 20, 30, 40, 50\}$.

4.3. Frame Skip Sampling for Data Augmentation

The main difficulty of training neural networks with small datasets is that the trained models do not generalize well with data from the validation and the test sets. As the size of dataset is even smaller than the usual cases in our problem, we utilize the *random frame skip sampling* that is commonly used to process video data such as video classification [38] for augmenting training data. The effectiveness if data augmentation has been proved in many tasks including image classification [39]. Here, we randomly extract multiple representative features of a video.

Given a sign video $S = (f_1, f_2, \ldots, f_l)$ that contains l frames from f_1 to f_l, we randomly select a fixed number of frames, say n. Then, we first compute the average length of gaps between frames as follows:

$$z = \left\lfloor \frac{l}{n-1} \right\rfloor.$$

We first extract a sequence of frames with indices from the following sequence $Y = (y, y+z, y+2z \ldots, y+(n-1)z) \in \mathbb{N}^n$, where $y = \lfloor \frac{l-z(n-1)}{2} \rfloor$ and call it a *baseline sequence*. Then, we generate a random integer sequence $R = (r_1, r_2, \ldots, r_n) \in [1, z]^n$ and compute the sum of the random sequence and the baseline sequence. Note that the value of the last index is clipped to the value in the range of $[1, l]$. We start from the baseline sequence instead of choosing any random sequence of length l to avoid generating random sequences of frames that are possibly not containing "key" moments of signs.

4.4. Attention-Based Encoder–Decoder Network

The encoder–decoder framework based on RNN architectures such as LSTMs or GRUs is gaining its popularity for neural machine translation [7,40–42] as it successfully replaces the statistical machine translation methods.

Given an input sentence $\mathbf{x} = (x_1, x_2, \ldots, x_{T_x})$, an encoder RNN plays its role as follows:

$$h_t = \text{RNN}(x_t, h_{t-1})$$

where $h_t \in \mathbb{R}^n$ is a hidden state at time t. After processing the whole input sentence, the encoder generates a fixed-size context vector that represents the sequence as follows:

$$c = q(h_1, h_2, \ldots, h_{T_x}),$$

For instance, the RNN is an LSTM cell and q simply returns the last hidden state h_{T_x} in one of the original sequence to sequence papers by Sutskever et al. [7].

Now, suppose that $\mathbf{y} = (y_1, y_2, \ldots, y_{T_y})$ is an output sentence that corresponds to the input sentence $\mathbf{x}$ in training set. Then, the decoder RNN is trained to predict the next word conditioned on all the previously predicted words and the context vector from the encoder RNN. In other words, the decoder computes a probability of the translation $\mathbf{y}$ by decomposing the joint probability into the ordered conditional probabilities as follows:

$$p(\mathbf{y}) = \prod_{i=1}^{T_y} p(y_i | \{y_1, y_2, \ldots, y_{i-1}\}, c).$$

Now, our RNN decoder computes each conditional probability as follows:

$$p(y_i | y_1, y_2, \ldots, y_{i-1}, c) = \text{softmax}(g(s_i)),$$

where s_i is the hidden state of decoder RNN at time i and g is a linear transformation that outputs a vocabulary-sized vector. Note that the hidden state s_i is computed by

$$s_i = \text{RNN}(y_{i-1}, s_{i-1}, c),$$

where y_{i-1} is the previously predicted word, s_{i-1} is the last hidden state of decoder RNN, and c is the context vector computed from encoder RNN.

Bahdanau attention. Bahdanau et al. [40] conjectured that the fixed-length context vector c is a bottleneck in improving the performance of the translation model and proposed to compute the context vector by automatically searching for relevant parts from the hidden states of encoder. Indeed, this "attention" mechanism has proven really useful in various tasks including but not limited to machine translation. They proposed a new model that defines each conditional probability at time i depending on a dynamically computed context vector c_i as follows:

$$p(y_i | y_1, y_2, \ldots, y_{i-1}, \mathbf{x}) = \text{softmax}(g(s_i)),$$

where s_i is the hidden state of the decoder RNN at time i which is computed by

$$s_i = \text{RNN}(y_{i-1}, s_{i-1}, c_i).$$

The context vector c_i is computed as a weighted sum of the hidden states from encoder:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j,$$

where

$$\alpha_{ij} = \frac{\exp(\text{score}(s_{i-1}, h_j))}{\sum_{k=1}^{T_x} \exp(\text{score}(s_{i-1}, h_k))}.$$

Here, the function "score" is called an *alignment function* that computes how well the two hidden states from the encoder and the decoder, respectively, match. For example, $\text{score}(s_i, h_j)$, where s_i is the hidden state of the encoder at time i and h_j is the hidden state of the decoder at time j implies the probability of aligning the part of the input sentence around position i and the part of the output sentence around position j.

Luong attention. Later, Luong et al. [41] examined a novel attention mechanism which is very similar to the attention mechanism by Bahdananu et al. but different in some details. First, only the hidden states

of the top RNN layers in both the encoder and decoder are used instead of using the concatenation of the forward and backward hidden states of the bi-directional encoder and the hidden states of the uni-directional non-stacking decoder. Second, the computation path is simplified by computing the attention matrix after computing the hidden state of the decoder at current time step. They also proposed the following three scoring functions to compute the degree of alignment between the hidden states as follows:

$$\text{score}(h_t, h_s) = \begin{cases} h_t^\intercal h_s, & \text{(Dot)} \\ h_t^\intercal W h_s, & \text{(General)} \\ V^\intercal \tanh(W[h_t; h_s]), & \text{(Concat.)} \end{cases}$$

where V and W are learned weights. Note that the third one based on the concatenation was originally proposed by Bahdanau et al. [40].

Multi-head attention (Transformer). While the previous encoder–decoder architectures are based on RNN cells, Vaswani et al. [42] proposed a completely new network architecture, which is based solely on attention mechanisms without any recurrence and convolutions. The most important characteristic of the Transformer is the *multi-head attention*, which is used in three different ways as follows:

1. Encoder–decoder attention: Each position in the decoder can attend over all positions in the input sequence.
2. Encoder self-attention: Each position in the encoder can attend over all positions in the previous layer of the encoder.
3. Decoder self-attention: Each position in the decoder can attend over all positions in the decoder up to and that position.

Moreover, as the Transformer uses neither recurrence nor convolution, the model requires some information about the order of the sequence. To cope with this problem, the Transformer uses *positional encoding*, which contains the information about the relative or absolute position of the words in the sequence using sine and cosine functions.

5. Experimental Results

We implemented our networks using PyTorch [43], which is an open source machine learning library for Python. The Adam optimizer [44] was used to train the network weights and biases for 50 epochs with an initial learning rate 0.001. During the training, we changed the learning rate every 20 epochs by the exponential decay scheduler with discount factor 0.5. We also used the dropout regularization with a probability of 0.8 and the gradient clipping with a threshold 5. Note that the dropout regularization is necessarily high as the size and the variation of the dataset is small compared to other datasets specialized for deep learning training. For the sequence-to-sequence models including the vanilla seq2seq model and two attention-based models, the dimension of hidden states was 256. For the Transformer model, we used the dimension for input and output (d_{model} in [42]) of 256. The other hyper-parameters used for the Transformer were the same as in the original model including the scheduled Adam optimizer in their own setting. Moreover, the batch size was 128, the augmentation factor was 100, the number of chosen frames was 50, and the object 2D normalization was used unless otherwise specified.

As our dataset is annotated in Korean, which is an agglutinative language, the morphological analysis on the annotated sentences should be performed because the size of dictionary can be arbitrarily large if we split sentences into words simply by white-spaces in such languages. For this reason, we used the Kkma part-of-speech (POS) tagger in the KoNLPy package, which is a Python package developed for natural language processing of the Korean language to tokenize the sentences into the POS level [45].

To evaluate the performance of our translation model, we basically calculated "accuracy", which means the ratio of correctly translated words and sentences. It should be mentioned that the

accuracy was calculated only for comparing sentence-level annotation and gloss-level annotation in Table 3 since it is not suitable to compare two cases by accuracy since we had five ground truth sentences for each sing video. In this experiment, we trained our model with a single ground truth sentence or a single sequence of glosses. Besides, we also utilized four types of metrics that are commonly used for measuring the performance of machine translation models: BLEU [46], ROUGE-L [47], METEOR [48], and CIDEr [49] scores.

Table 3. Comparison of sign language translation performance on different types of annotations. Note that bold numbers represent the best performance results.

Annotation	Validation Set				Test Set			
	Accuracy	ROUGE-L	METEOR	BLEU	Accuracy	ROUGE-L	METEOR	BLEU
Sentence-level	82.07	94.42	67.35	90.57	45.56	**66.10**	**41.09**	**57.37**
Gloss-level	**93.28**	**96.03**	**71.04**	**93.85**	**55.28**	63.53	38.10	52.63

Sentence-level vs. Gloss-level training. As in [23], we conducted an experiment to compare the translation performance depending on the type of annotations. Because each sign corresponds to a unique sequence of glosses as well as to multiple natural language sentences, it is easily predictable that the gloss-level translation shows better performance. Indeed, we can confirm the anticipation from the summary of results provided in Table 3.

This also leads us to the future work for translating sequences of glosses into natural language sentences. We expect that the sign language translation can be a more feasible task by separating the task of annotating sign videos with natural language sentences by two sub-tasks where we annotate sign videos with glosses and annotate each sequence of glosses with natural language sentences.

Comparison with CNN-based approaches. In Table 4, we compare our approach to the classical methods based on CNN features extracted from well-known architectures such as ResNet [50] and VGGNet [51]. Since the size of sign video frames (1920×1080) is different from the size of input of CNN models (224×224), we first cropped the central area of frames in 1080×1080 and resize the frames to 224×224.

Table 4. Performance comparison with translation models based on CNN-based feature extraction techniques. Note that the augmentation factor in this experiment was all set to 50.

Feature Type	Validation Set				Test Set			
	ROUGE-L	METEOR	BLEU	CIDEr	ROUGE-L	METEOR	BLEU	CIDEr
VGGNet-16 [51]	66.85	41.92	56.75	2.369	**44.75**	**25.79**	**27.88**	**1.016**
VGGNet-19 [51]	61.77	38.72	50.95	2.060	42.75	24.81	24.27	0.839
ResNet-50 [50]	62.26	38.79	51.99	2.124	38.76	21.85	19.45	0.664
ResNet-101 [50]	66.28	41.81	56.26	2.368	40.10	22.68	21.86	0.772
ResNet-152 [50]	**74.10**	**48.03**	**66.73**	**2.841**	38.44	22.71	20.78	0.753
OpenPose [32–34]	96.92	72.14	96.11	4.380	73.95	46.66	64.79	2.832

The experimental results show that ResNet-152 exhibits the best translation performance on the validation set and the VGGNet-16 demonstrates the best performance on the test set. In general, the performance difference on the validation set is not large but it is apparent that the VGGNet models are much better in generalizing to the test set compared to the ResNet models.

Expectably, the translation models using the CNN extracted features show significantly worse translation performance than the models using the human keypoint features. It is well-known that CNN-based architectures such as ResNet and VGGNet have a huge number of trainable parameters (e.g., VGGNet-19 and ResNet-152 have over 143M and 60M parameters, respectively.) so that they easily fall into the overfitting problem due to the lack of a sufficient number of training examples.

Moreover, the CNN-based models have a weakness for recognizing signs of previously unseen signers as they are weaker than our model in dealing with subtle variances of images.

It is still interesting to know whether the combination of any CNN-based features and human keypoint features works better than when we solely rely on the human keypoint features. As the size of sign language dataset grows, we expect that the CNN-based models improve their performances and generalize much better.

Effect of feature normalization methods. To evaluate the effect of the feature normalization method on the keypoints estimated by OpenPose, we compared the following five cases: (1) no normalization; (2) feature normalization; (3) object normalization; (4) two-dimensional (2D) normalization; and (5) object 2D normalization. In the first case, we did not perform any normalization step on the keypoint feature generated by concatenating the coordinate values of all keypoints. In the feature normalization, we created a keypoint feature as in (1) and normalized the feature with the mean and standard deviation of the whole feature. In the object normalization, we normalized the keypoint features obtained from two hands, body, and face, respectively, and concatenated them to generate a feature that represents the frame. We also considered the case of 2D normalization in which we normalized the x- and y-coordinates separately. Lastly, the object 2D normalization is the normalization method that we propose in the paper.

Table 5 summarizes the result of our experiments. The table does not contain the results of the case without any normalization as it turns out that the proposed object 2D normalization method is superior to the other normalization methods we considered. Especially, when we trained our neural network with the keypoint feature vector which was obtained by simply concatenating the x- and y-coordinates of keypoints without any normalization, the validation loss never decreased. While any kind of normalization seemed to work positively, it is quite interesting to see that there is an additional boost in translation performance when the object-wise normalization and the 2D normalization were used together.

Table 5. Effect of different feature normalization methods on the translation performance. The results were obtained on the test set.

Method	ROUGE-L	METEOR	BLEU	CIDEr
Feature Normalization	66.28	40.94	56.91	2.401
2D Normalization	72.05	44.98	62.69	2.706
Object Normalization	64.16	38.84	53.83	2.235
Object 2D Normalization	**73.61**	**46.52**	**65.26**	**2.794**

Effect of attention mechanisms. Here, we compared four types of encoder–decoder architectures that are specialized in various machine translation tasks. Table 6 demonstrates the clear contrast between the attention-based model by Luong et al. [41] and the Transformer [42]. While the model of Luong et al. shows better performance than the Transformer on the validation set that contains more similar data to the training set, the Transformer generalizes much better to the test set, which consists of sign videos of an independent signer.

Table 6. Performance comparison of sign language translation on different types of attention mechanisms.

Attention Type	Validation Set				Test Set			
	ROUGE-L	METEOR	BLEU	CIDEr	ROUGE-L	METEOR	BLEU	CIDEr
Vanilla seq2seq [7]	90.03	62.66	87.79	3.838	62.93	38.03	50.80	2.129
Bahdanau et al. [40]	94.72	67.44	94.03	4.264	71.45	44.06	63.38	2.616
Luong et al. [41]	**96.63**	**72.24**	**95.86**	**4.322**	**73.61**	46.52	65.26	2.794
Transformer [42]	94.14	69.27	92.90	4.227	73.18	**47.03**	**66.58**	**2.857**

Effect of augmentation factor. We examined the effect of data augmentation by random frame skip sampling and summarize the experimental results in Table 7. We call the number of training samples randomly sampled from a single sign video the *augmentation factor*. Since the number of sign videos in the training set is 9432, the total number of training samples after the data augmentation is 943,200 when the augmentation factor is 100.

It should be noted that we do not include the result when we do not augment data by random frame sampling because the validation loss did not decrease at all due to severe overfitting. The result shows that the optimal augmentation factor is 50 for the validation and test set. This implies that a larger augmentation factor does not always lead to improvement in performance and even in generalization capability.

Table 7. Effects of data augmentation by random frame sampling on sign language translation performance.

Augmentation Factor	Validation Set				Test Set			
	ROUGE-L	METEOR	BLEU	CIDEr	ROUGE-L	METEOR	BLEU	CIDEr
100	96.63	**72.24**	95.86	4.322	73.61	46.52	**65.26**	2.794
50	**96.92**	72.14	**96.11**	**4.380**	**73.95**	**46.66**	64.79	**2.832**
10	95.69	70.14	94.46	4.227	71.40	45.10	62.95	2.662

Effect of the number of sampled frames. It is useful to know the optimal number of frames if we plan to develop a real-time sign language translation system because we can reduce the computational cost of the inference engine by efficiently skipping unnecessary frames. Table 8 shows how the number of sampled frames affects the translation performance. As the sequence-to-sequence model works for any variable-length input sequences, we do not necessarily fix the number of sampled frames. However, it is useful to know the optimal number of frames as the translation performance of the sequence-to-sequence models tends to decline with longer input sequences due to the vanishing gradient problem [52]. Our experimental result shows that the optimal number of frames for the best translation performance is 40 for the validation set and 50 for the test set.

Table 8. Effects of the number of sampled frames on sign language translation performance.

Number of Frames	Validation Set				Test Set			
	ROUGE-L	METEOR	BLEU	CIDEr	ROUGE-L	METEOR	BLEU	CIDEr
50	**96.63**	72.24	95.86	4.322	**73.61**	**46.52**	**65.26**	**2.794**
40	96.52	**72.36**	**95.96**	**4.327**	72.71	46.18	64.35	2.757
30	95.88	70.60	94.87	4.281	73.35	46.46	64.48	2.778
20	94.38	68.40	92.98	4.181	72.37	45.37	62.19	2.693
10	83.26	55.81	78.43	3.427	65.89	40.65	54.01	2.308

Effect of batch size. Recently, it is increasingly accepted that training with a small batch often generalizes better to the test set than training with large batch [53,54]. However, our experimental results provided in Table 9 show the opposite phenomenon. We suspect that this is due to the scale of the original dataset because the large batch is known to be useful to prevent overfitting to some extent.

Table 9. Effects of the batch size on sign language translation performance.

Batch Size	Validation Set				Test Set			
	ROUGE-L	METEOR	BLEU	CIDEr	ROUGE-L	METEOR	BLEU	CIDEr
128	96.63	72.24	95.86	4.322	**73.61**	**46.52**	**65.26**	**2.794**
64	**96.94**	**73.20**	**96.35**	**4.332**	72.93	46.06	63.91	2.725
32	95.65	70.68	94.57	4.231	71.94	45.30	62.71	2.673
16	93.74	67.58	92.74	4.118	70.63	43.86	61.94	2.571

Generalizations to real world data. As we can see in Figure 2, the sign videos in our dataset were recorded in a clear background. This led us to investigate the performance of our model when the background area of sign videos was cluttered. Moreover, we also checked how well our system generalizes to the real world situations by testing the system against more realistic sign video examples that were collected in a much less constrained setting. We collected 30 additional sign videos of five signs from six novice signers in relatively cluttered background to test the generalization ability of our system. Note that we selected relatively easier five signs among the 105 signs since it is very difficult to follow complicated signs for the novice singers who never learned signs before. Figure 5 contains example frames from the additional sign videos. As a result, our system achieved 89.06 (ROUGE-L), 60.64 (METEOR), 77.08 (BLEU) and 2.860 (CIDEr) in the four metrics. The experimental result shows that our system generalizes well to the real world conditions such as cluttered background area and clumsy signs from non-experts.

Figure 5. Example frames of additional test set videos recorded with cluttered background and non-expert signers.

Ablation Study

We also studied the effect of the use of keypoint information from two hands, body, and face. The experimental results summarized in Table 10 imply that the keypoint information from both hands is the most important among all the keypoint information from hands, face, and body.

Table 10. Ablation study on the contributions of keypoints from body, face, and hands. The results are obtained on the test set.

Method	ROUGE-L	METEOR	BLEU	CIDEr
Body	68.82	42.68	59.96	2.554
Hand	72.03	45.19	62.90	2.681
Body, Face	60.08	36.66	48.90	2.051
Hand, Face	69.43	43.26	59.92	2.538
Hand, Body	**74.02**	**46.84**	**65.83**	**2.831**
Hand, Body, Face	73.61	46.52	65.26	2.794

Interestingly, the experimental result tells us that the keypoint information from face does not help to improve the performance in general. The performance even dropped when we added face keypoints in all cases. We suspect that the reason is partly due to the imbalanced number of keypoints from different parts. Recall that the number of keypoints from face is 70 and this is much larger than the number of the other keypoints.

While the keypoints from both hands are definitely the most important features to understand signs, it is worth noting that the 12 keypoints from body boosted the performance. Actually, we losy the information about relative positions of parts from each other as we normalized the coordinates of each part separately. For instance, there was no way to infer the relative positions of two hands with the normalized feature vectors from both hands. However, it was possible to know the relative position from the keypoints of body as there also existed keypoints corresponding to the hands.

6. Conclusions

In this work, we have introduced a new sign language dataset, which is manually annotated in Korean spoken language sentences and proposed a neural sign language translation model based on the sequence-to-sequence translation models. It is well-known that the lack of large sign language dataset significantly hinders the full utilization of neural network based algorithms for the task of sign language translation that are already proven very useful in many tasks. Moreover, it is really challenging to collect a sufficient amount of sign language data as we need help from sign language experts. For this reason, we claim that it is inevitable to extract high-level features from sign language video with a sufficiently lower dimension. We are able to successfully train a novel sign language translation system based on the human keypoints that are estimated by a famous open source project called OpenPose [32–34] developed by Hidalgo et al.

In the future, we aim at improving our sign language translation system by exploiting various data augmentation techniques using the spatial properties of videos. We also expect that the performance of the proposed system can be improved if the performance of the human keypoint detection is improved. For instance, there have been various approaches in human keypoint detection such as Mask R-CNN [55] and AlphaPose [56] that exhibit even better performances than OpenPose in terms of accuracy. It is also possible to apply landmark detection methods [57–59] for better performance of keypoint detection from human body, face and both hands. We plan to implement different keypoint detection methods for sign language translation and compare the performances of the methods. It is also important to expand the KETI sign language dataset to sufficiently larger scale by recording videos of more signers in different environments.

Author Contributions: Conceptualization, H.J. and C.C.; methodology, S.-K.K.; software, S.-K.K. and C.J.K.; validation, S.-K.K. and C.J.K.; writing—original draft preparation, S.-K.K.; writing—review and editing, S.-K.K. and C.C.; project administration, H.J.; and funding acquisition, H.J.

Funding: This work was supported by the IT R&D program of MSIT/IITP [2017-0-00255, Autonomous digital companion framework and application].

Acknowledgments: We thank the anonymous referees for a careful reading of an earlier version of the paper and for many valuable suggestions that have improved the presentation. We also sincerely thank our colleagues from Korea Nazarene University who provided insight and expertise in Korean sign language that greatly assisted our research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Forster, J.; Schmidt, C.; Hoyoux, T.; Koller, O.; Zelle, U.; Piater, J.; Ney, H. RWTH-PHOENIX-Weather: A Large Vocabulary Sign Language Recognition and Translation Corpus. In Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey, 23–25 May 2012.

2. Von Agris, U.; Knorr, M.; Kraiss, K.F. The Significance of Facial Features for Automatic Sign Language Recognition. In Proceedings of the 8th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2008), Amsterdam, The Netherlands, 17–19 September 2008; pp. 1–6.
3. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
4. Cho, K.; van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
5. Dai, B.; Lin, D. Contrastive Learning for Image Captioning. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 898–907.
6. Liu, C.; Mao, J.; Sha, F.; Yuille, A.L. Attention Correctness in Neural Image Captioning. In Proceedings of the 31th AAAI Conference on Artificial Intelligence (AAAI 2017), San Francisco, CA, USA, 4–9 February 2017; pp. 4176–4182.
7. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014 (NIPS 2014), Montréal, Canada, 8–13 December 2014; pp. 3104–3112.
8. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhudinov, R.; Zemel, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of the 32nd International Conference on Machine Learning (ICML 2015), Lille, France, 6–11 July 2015; pp. 2048–2057.
9. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
10. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
11. Gao, M.; Yu, R.; Li, A.; Morariu, V.I.; Davis, L.S. Dynamic zoom-in network for fast object detection in large images. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 21–26.
12. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
13. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
14. Zhang, H.; Dana, K.; Shi, J.; Zhang, Z.; Wang, X.; Tyagi, A.; Agrawal, A. Context encoding for semantic segmentation. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7151–7160.
15. Donahue, J.; Anne Hendricks, L.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. Long-term Recurrent Convolutional Networks for Visual Recognition and Description. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), Boston, MA, USA, 7–12 June 2015; pp. 2625–2634.
16. Luvizon, D.C.; Picard, D.; Tabia, H. 2D/3D Pose Estimation and Action Recognition using Multitask Deep Learning. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–22 June 2018; pp. 5137–5146.
17. Dong, C.; Leu, M.C.; Yin, Z. American Sign Language alphabet recognition using Microsoft Kinect. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), Boston, MA, USA, 7–12 June 2015; pp. 44–52.
18. Gattupalli, S.; Ghaderi, A.; Athitsos, V. Evaluation of deep learning based pose estimation for sign language recognition. In Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments (PETRA 2016), Corfu, Greece, 29 June–1 July 2016; pp. 12:1–12:7.

19. Kishore, P.V.V.; Sastry, A.S.C.S.; Kartheek, A. Visual-verbal machine interpreter for sign language recognition under versatile video backgrounds. In Proceedings of the 1st International Conference on Networks Soft Computing (ICNSC 2014), Miami, FL, USA, 7–9 April 2014; pp. 135–140.
20. Koller, O.; Forster, J.; Ney, H. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Comput. Vis. Image Understand.* **2015**, *141*, 108–125. [CrossRef]
21. Koller, O.; Zargaran, S.; Ney, H. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 3416–3424.
22. Starner, T.; Pentland, A. Real-time American Sign Language recognition from video using hidden Markov models. In Proceedings of the International Symposium on Computer Vision (ISCV 1995), Coral Gables, FL, USA, 21–23 November 1995; pp. 265–270.
23. Cihan Camgoz, N.; Hadfield, S.; Koller, O.; Ney, H.; Bowden, R. Neural Sign Language Translation. In Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2018), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7784–7793.
24. Kim, T.; Kim, S. Sign language translation system using latent feature values of sign language images. In Proceedings of the 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2016), Xi'an, China, 19–22 August 2016; pp. 228–233.
25. Buehler, P.; Zisserman, A.; Everingham, M. Learning sign language by watching TV (using weakly aligned subtitles). In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami Beach, FL, USA, 20–25 June 2009; pp. 2961–2968.
26. Cooper, H.; Bowden, R. Learning signs from subtitles: A weakly supervised approach to sign language recognition. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami Beach, FL, USA, 20–25 June 2009; pp. 2568–2574.
27. Pfister, T.; Charles, J.; Zisserman, A. Large-scale Learning of Sign Language by Watching TV (Using Co-occurrences). In Proceedings of the 24th British Machine Vision Conference (BMVC 2013), Bristol, UK, 9–13 September 2013.
28. Forster, J.; Schmidt, C.; Koller, O.; Bellgardt, M.; Ney, H. Extensions of the Sign Language Recognition and Translation Corpus RWTH-PHOENIX-Weather. In Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, Iceland, 26–31 May 2014; pp. 1911–1916.
29. Oberweger, M.; Wohlhart, P.; Lepetit, V. Hands Deep in Deep Learning for Hand Pose Estimation. In Proceedings of the 20th Computer Vision Winter Workshop (CVWW 2015), Styria, Austria, 9–11 February 2015; pp. 1–10.
30. Ko, S.; Son, J.G.; Jung, H.D. Sign language recognition with recurrent neural network using human keypoint detection. In Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems (RACS 2018), Honolulu, HI, USA, 9–12 October 2018; pp. 326–328.
31. Liddell, S.K. *Grammar, Gesture, and Meaning in American Sign Language*; Cambridge University Press: Cambridge, UK, 2003.
32. Cao, Z.; Simon, T.; Wei, S.; Sheikh, Y. Realtime Multi-person 2D Pose Estimation Using Part Affinity Fields. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 1302–1310.
33. Simon, T.; Joo, H.; Matthews, I.A.; Sheikh, Y. Hand Keypoint Detection in Single Images Using Multiview Bootstrapping. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI, USA, 21–26 July 2017; pp. 4645–4653.
34. Wei, S.; Ramakrishna, V.; Kanade, T.; Sheikh, Y. Convolutional Pose Machines. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 27–30 June 2016; pp. 4724–4732.
35. Ba, L.J.; Kiros, R.; Hinton, G.E. Layer Normalization. *arXiv* **2016**, arXiv:1607.06450.
36. Ioffe, S.; Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In Proceedings of the 32nd International Conference on Machine Learning (ICML 2015), Lille, France, 6–11 July 2015; pp. 448–456.
37. Ulyanov, D.; Vedaldi, A.; Lempitsky, V.S. Instance Normalization: The Missing Ingredient for Fast Stylization. *arXiv* **2016**, arXiv:1607.08022.

38. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-Scale Video Classification with Convolutional Neural Networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), Columbus, OH, USA, 23–28 June 2014; pp. 1725–1732.
39. Perez, L.; Wang, J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv* **2017**, arXiv:1712.04621.
40. Bahdanau, D.; Cho, K.; Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
41. Luong, T.; Pham, H.; Manning, C.D. Effective Approaches to Attention-based Neural Machine Translation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), Lisbon, Portugal, 17–21 September 2015; pp. 1412–1421.
42. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017 (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
43. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic Differentiation in PyTorch. NIPS-W, 2017. Available online: https://openreview.net/pdf?id=BJJsrmfCZ (accessed on 30 June 2019).
44. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
45. Park, E.L.; Cho, S. KoNLPy: Korean natural language processing in Python. In Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology (HCLT 2014), Chuncheon, Korea, 11–14 October 2014; pp. 133–136.
46. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL 2002), Philadephia, PA, USA, 6–12 July 2002; pp. 311–318.
47. Lin, C.Y. ROUGE: A Package for Automatic Evaluation of summaries. In Proceedings of the ACL workshop on Text Summarization Branches Out, Barcelona, Spain, 25–26 July 2004; pp. 74–81.
48. Banerjee, S.; Lavie, A. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Ann Arbor, MI, USA, 29 June 2005; pp. 65–72.
49. Vedantam, R.; Zitnick, C.L.; Parikh, D. CIDEr: Consensus-based image description evaluation. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015), Boston, MA, USA, 7–12 June 2015; pp. 4566–4575.
50. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
51. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015.
52. Pascanu, R.; Mikolov, T.; Bengio, Y. On the difficulty of training recurrent neural networks. In Proceedings of the 30th International Conference on Machine Learning (ICML 2013), Atlanta, GA, USA, 16–21 June 2013; pp. 1310–1318.
53. Hoffer, E.; Hubara, I.; Soudry, D. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 1729–1739.
54. Smith, S.L.; Kindermans, P.; Ying, C.; Le, Q.V. Don't Decay the Learning Rate, Increase the Batch Size. In Proceedings of the 6th International Conference on Learning Representations (ICLR 2018), Vancouver, BC, Canada, 30 April–3 May 2018.
55. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.B. Mask R-CNN. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV 2017), Venice, Italy, 22–29 October 2017; pp. 2980–2988.

56. Fang, H.; Xie, S.; Tai, Y.; Lu, C. RMPE: Regional Multi-person Pose Estimation. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV 2017), Venice, Italy, 22–29 October 2017; pp. 2353–2362.
57. Duan, J.; Schlemper, J.; Bai, W.; Dawes, T.J.W.; Bello, G.; Doumou, G.; De Marvao, A.; O'Regan, D.P.; Rueckert, D. Deep Nested Level Sets: Fully Automated Segmentation of Cardiac MR Images in Patients with Pulmonary Hypertension. In Proceedings of the 22nd International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI 2018), Granada, Spain, 16–20 September 2018; pp. 595–603.
58. Duan, J.; Bello, G.; Schlemper, J.; Bai, W.; Dawes, T.; Biffi, C.; de Marvao, A.; Doumou, G.; O'Regan, D.; Rueckert, D. Automatic 3D bi-ventricular segmentation of cardiac images by a shape-refined multi-task deep learning approach. *IEEE Trans. Med. Imag.* **2019**. [CrossRef] [PubMed]
59. Wu, Y.; Ji, Q. Facial Landmark Detection: A Literature Survey. *Int. J. Comput. Vis.* **2019**, *127*, 115–142. [CrossRef]