Enabling Real-time Sign Language Translation on Mobile Platforms with On-board Depth Cameras

HYEONJUNG PARK, School of Integrated Technology, Yonsei University, South Korea
YOUNGKI LEE, Department of Computer Science and Engineering, Seoul National University, South Korea
JEONGGIL KO*, School of Integrated Technology, Yonsei University, South Korea

In this work we present *SUGO*, a depth video-based system for translating sign language to text using a smartphone's front camera. While exploiting depth-only videos offer benefits such as being less privacy-invasive compared to using RGB videos, it introduces new challenges which include dealing with low video resolutions and the sensors' sensitiveness towards user motion. We overcome these challenges by diversifying our sign language video dataset to be robust to various usage scenarios via data augmentation and design a set of schemes to emphasize human gestures from the input images for effective sign detection. The inference engine of *SUGO* is based on a 3-dimensional convolutional neural network (3DCNN) to classify a sequence of video frames as a pre-trained word. Furthermore, the overall operations are designed to be light-weight so that sign language translation takes place in real-time using only the resources available on a smartphone, with no help from cloud servers nor external sensing components. Specifically, to train and test *SUGO*, we collect sign language data from 20 individuals for 50 Korean Sign Language words, summing up to a dataset of ~5,000 sign gestures and collect additional in-the-wild data to evaluate the performance of *SUGO* in real-world usage scenarios with different lighting conditions and daily activities. Comprehensively, our extensive evaluations show that *SUGO* can properly classify sign words with an accuracy of up to 91% and also suggest that the system is suitable (in terms of resource usage, latency, and environmental robustness) to enable a fully mobile solution for sign language translation.

CCS Concepts: • **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

Additional Key Words and Phrases: Sign language translation; Mobile applications and services; Depth image processing

ACM Reference Format:
HyeonJung Park, Youngki Lee, and JeongGil Ko. 2021. Enabling Real-time Sign Language Translation on Mobile Platforms with On-board Depth Cameras. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 5, 2, Article 77 (June 2021), 30 pages. https://doi.org/10.1145/3463498

1 INTRODUCTION

5% of the world population lives with hearing disabilities, and 0.4% are functionally deaf [1]. While people with mild hearing loss can still communicate verbally, with serious functional deafness, natural communication becomes difficult. Thus, many cultures have developed their form of sign language. The American Sign Language (ASL) is a widely used example together with the British, Australian, and New Zealand Sign Language (BANZSL).

*Corresponding Author: jeonggil.ko@yonsei.ac.kr

Authors' addresses: HyeonJung Park, School of Integrated Technology, Yonsei University, 85 Songdogwahak-Ro, Yeonsu-Gu, Incheon, South Korea; Youngki Lee, Department of Computer Science and Engineering, Seoul National University, 1 Gwanak-Ro, Gwanak-Gu, Seoul, South Korea; JeongGil Ko, School of Integrated Technology, Yonsei University, 85 Songdogwahak-Ro, Yeonsu-Gu, Incheon, South Korea.

2474-9567/2021/6-ART77 $15.00
https://doi.org/10.1145/3463498

There are many other locally used language examples, such as the Korean Sign Language (KSL) and Japanese Sign Language (JSL). These different sign languages result from varying alphabet patterns and cultural differences, but all share the common characteristic that words are visually expressed using the hand(s), facial expressions and upper body movements to assist message delivery [8]. Nevertheless, the communication gap between verbal communication and sign language is yet prominent.

In the mobile and ubiquitous computing community, there have been efforts to exploit sensing platforms for sign language translation [14, 34, 37, 39, 44, 57, 60]. These previous works use devices such as RGB cameras [5, 21, 27, 29, 33, 35, 46, 54, 55], motion sensors (e.g., Leap Motion) [14, 41], depth cameras/sensors (e.g., Kinect) [6, 10, 11, 16, 38, 48, 51], or electromyogram (EMG) sensors [53, 57] to capture user hand motions and combine sensing results with various machine learning models to infer the word being expressed. More recently, research has considered the contextual meanings of words and their syntaxial relationships to generate proper sentences from sign language motions [13, 14, 21]. However, it is not trivial to apply these technologies in everyday situations since they either require additional devices or infrastructure support. First, many systems adopt an extra sensing device for accurate hand motion tracking. For example, MyoSign uses EMG sensors [57] and DeepASL [14] requires a Leap Motion [50] depth sensor. Second, the use of infrastructure-embedded sensors (e.g., WiFi [34, 45, 56], mmWave radios [44]) require the users to perform hand gestures in pre-defined locations. In addition, while sign language consist of more than the hand gestures themselves [15], most previous works have only focused on the gestural characteristics (i.e., movements) of the hands.

In this work, we propose a new system *SUGO*, which uses the smartphone as the *only* hardware platform (with the server only taking the role of model training and distribution) for light-weight and accurate on-device sign language translation. Our key approach is in leveraging a depth-camera equipped in most state-of-the-art smartphones. Unlike some previous systems that exploit the smartphone's RGB camera [21, 46], the depth camera provides unique opportunities in three folds: (i) improving translation accuracy with new depth information, which is also more robust against light conditions, colors of clothing and surroundings, and occlusions, (ii) reducing the processing burden as the depth images include much smaller information while sufficient for sign language translation, and (iii) alleviating user-perceived privacy concerns related to using RGB videos in third-party apps. Furthermore, depth cameras are no longer "add-on" accessories carried around separately. For instance, since iPhone X, the iPhone has included a TrueDepth sensor, and the Pixel 4 provides depth capturing capabilities using multiple (stereo) cameras. By leveraging these benefits, *SUGO* performs all operations required for translation entirely on the smartphone, minimizing inference latency and reducing potential privacy invasions due to sharing continuous images over the wireless network.

Despite the advantages of using depth information, several challenges persist. First of all, there have been no prior studies to build a computational model to translate sign language solely based on the depth video information. A naive computational model would fail to accurately translate the sign language since the depth information is just a single channel grayscale image with significant noises. Such limited information can make the translation model vulnerable to various motion artifacts introduced across multiple sign language gestures. In addition, the characteristics of depth cameras embedded in today's smartphones are still understudied, and their performance under various light conditions remains unknown for custom applications. Finally, the end-to-end computation pipeline should run on resource-constrained mobile devices, requiring a light-weight computational model.

To address these challenges, *SUGO* first adopts a *3-dimensional convolutional neural network (3DCNN)* as its baseline inference model for feature extraction and word classification. The 3DCNN effectively extracts temporal patterns of a user's hand movements from depth image frames, allowing for accurate sign language translation [9, 23]. On top of this base model, we developed a suite of techniques to improve translation accuracy as well as the computational cost. Firstly, the raw images from the smartphone's depth camera are passed through a *human gesture emphasis module*. In this module, we restructure the pixel values to emphasize and re-normalize

depth representations of effective regions while nullifying the irrelevant background. Secondly, we perform *augmentation* on the training dataset by mimicking hand-trembling effects to make our inference model robust against hand motion-initiated artifacts. Finally, to assure that the 3DCNN can operate on resource-limited mobile devices, we *prune and fine-tune* the model to achieve a small memory footprint with low inference latency.

For a thorough evaluation of our system, we collect 50 different Korean Sign Language words from 20 people (including two trained experts), summing up to a dataset of 5,000 sign word gestures. We carefully select the 50 words that represent possible forms of hand and body gestures to ensure the contextual scalability of *SUGO* towards supporting more words with similar motion characteristics. Moreover, we collect an additional dataset to confirm that *SUGO* is robust under various practical usage conditions such as varying light conditions and different external noise factors (e.g., vibrations from user motions).

Overall, the word-level classification results of *SUGO* are as high as 91%, and such high classification accuracy holds even under various environmental conditions. Furthermore, our implementations of *SUGO* on embedded GPUs and the iPhone show that the classification latency can be kept low even when operating with the limited computing resources of a smartphone. Overall, our results suggest that *SUGO* shows the potential to be a practically applicable sign language translation system in real-world use cases.

Specifically, the contributions of this work are as follows:

- **[Mobile Depth Camera-based Sign Language Translation System]:** We introduce an on-device sign language translation system, enabled by a depth camera. Our study shows that depth videos collected from recent smartphones, combined with a deep learning model for efficiently processing the input videos can be an effective sensing modality for sign language translation.
- **[Accurate and Light-weight Sign Language Translation Techniques:]** We develop a suite of techniques (e.g., human gesture emphasis module, a light-weight 3DCNN-based translation model, and data augmentation) to utilize depth information for sign language translation effectively. Our system shows that the depth information is indeed useful to i) improve the accuracy of the sign language translation compared to using RGB datasets, and ii) show improved robustness under various lighting conditions (up to ~40% in dark environments and ~10% with typical indoor lighting compared to the RGB-based model). In this work, we focus on exploring the possibility of utilizing the depth information as the sole sensing modality, but it can be fused with other modalities to achieve an even higher translation accuracy, especially when powerful networking and computing resources are available.
- **[Large-scale Open Sign Language Dataset]:** In this work, we collect and present a large-scale dataset consisting of 5,000 KSL signs (50 unique words) and a multi-environment dataset that consists of 1,000 signs under different lighting and motion characteristics. When selecting the 50 words, we focus on the hand gestures' motion characteristics rather than the context to examine the scalability of depth video-based sign language translation systems. The Korean Sign Language dataset used in this work is publicly available at https://www.eis-lab.org/yonsei-ksl-dataset.

2 RELATED WORK

Sign language translation research combines findings from a number of different research domains, ranging from computer vision, IoT, and mobile computing. Based on such various technologies, the research community has proposed a number of systems to support sign language translation by combining a number of different sensors and computing platforms.

• **Glove-based Motion Capturing**: Given that sign language is mostly performed with a user's hand, previous work have proposed glove type sensors to capture a person's hand gestures in detail. Such approaches have a solid advantage over other indirect sensing modalities that fine-grained samples user hand shapes or movements can be captured. Mohandes [37] designed a two-handed sign recognition system for Unified Arabic Sign Language

Table 1. Comparison of *SUGO* with recently proposed related work in sign language translation. We do not provide a full list, but present most relevant examples for different sensing modalities.

Paper	Modality	Devices	# of Words	# of Subjects	On-device	Sensing Parts
EIS [60]	Epidermal -iontronic	EIS glove	35	8	X	Hand
DeepASL [14]	Skeleton	Leap motion	56	11	O	Hand/Arm
Signfi [34]	WiFi	WiFi APs	150	5	X	Head/Hand/Arm
MyoSign [57]	EMGs	Myo	100	15	O	Hand/Arm
SignSpeaker [19]	IMU	Smartwatch	129	16	X	Hand/Arm
mmASL [44]	60 GHz mm wave	Radio	50	15	X	Upper body / Head / Arm / Hand
DeepArSLR [5]	Color	RGB camera	23	3	X	Hand
Molchanov et al. [38]	Color+Depth	Kinect	25	20	X	Hand
Huang et al. [20]	Color+Depth +Skeleton	Kinect	25	9	X	Upper body / Head / Arm / Hand
SUGO	Depth	Smartphone	50	20	O	Upper body / Head / Arm / Hand

using CyberGloves, and a support vector machine (SVM) was used as a classifier. Mummadi et al. [39] designed a real-time recognition system for the French Sign Language alphabet using a custom-developed IMU sensor-based glove. In this work, the authors exploit a multi-layer perceptron (MLP) to classify between different alphabet letters. The EIS system proposed by Zhu et al. [60] translates and recognizes American Sign Language alphabets through a glove-type epidermal iontronic sensing based wearable device. While glove type wearables offer a detailed perspective on the hand gestures, they show two major limitations. First, the act of wearing the glove for sign language translation can be burdensome and second, they can *only* capture the hand gestures. Previous literature has shown that sign language is not only a language expressed using the hands, but also where the hand is positioned on the human body and body motions (combined with hand gestures) can reflect different meanings [52].

• **Band-type Sensing**: Unlike a glove type wearable, a sensing device in the form of an arm band worn on a wrist or the arm can capture sign language gestures using information by detecting the minute movements of the fingers and hand. MyoSign [57] integrates Myo EMG sensors [40] to capture different muscle movements for sign language classification. Combined with a Bi-LSTM, MyoSign achieves a classification accuracy of 93.7% for 100 different ASL words. More recently, SignSpeaker [19] has been proposed to capture hand gestures using a smartwatch platform for sign language translation. It captures IMU sensor signals from the users' smartwatch and analyzes the data using an LSTM model. These wearable sensors are less invasive compared to glove-type systems, but still share similar limitations, given the additional burden of carrying an additional sensor device and the limits of only being able to capture hand motions.

• **User Skeleton-based Sensing**: Another active direction of research is in exploiting user skeleton information for sign gesture recognition. Using devices such as the Microsoft Kinect [58] or Leap Motion [50], these systems capture skeletal movements of the user's hand and/or upper body to translate motions to sign language words. DeepASL [14] is a system that exploits skeleton data collected from a Leap Motion platform. It exploits a hierarchical bidirectional deep recurrent neural network for sign classification and shows 94% accuracy for 56 ASL words collected from 11 subjects. Naglot et al. [41] also used leap motion sensor-based hand skeleton data to create a multi-layer perceptron classifier for the ASL alphabet. While the reported results are promising, the

need for an external sensing platform still remains. Systems that use larger devices such as the Kinect, require a designated room in which the device is installed [10].

• **Exploiting RGB Videos**: Sign language recognition through RGB video has been studied for a long time in the field of computer vision, and has the advantage of being able to obtain datasets more easily compared to other sensing modalities. Yogopuspito et al. [55] and Makarov et al. [35] designed a lightweight system that classifies sign language alphabets on a smartphone. Liao et al. [33] designed a word-level sign language translation system using a 3DCNN architecture and LSTM models via handcrafted RGB video-based features, and Li et al. [31] compared the performance of various deep learning architectures with a large scale word-level dataset. Aly et al. [5] also designed a word-level sign language recognition system focusing on hand shape. Koller et al. [29] designed a sentence-level sign language translation system through the open source PHOENIX [28] dataset. Unfortunately, word-level classification models in previous work were not designed to operate fully on mobile platforms. The video dataset used in these works were mostly captured from PC-connected cameras and inference operations took place on a server-scale device only.

• **RGB+Depth Videos for Sign Language Translation**: Another drawback of exploiting RGB videos is that the content includes various information that can bother the classification process. As an example, color similarities with the hand and clothing and externally present patterns can interfere with the segmentation operations needed in RGB-based sign language translation. For this, some recent work have integrated depth information together with RGB videos. Aly et al. [6] designed a sign language alphabet translation system with RGB+depth videos and Molchanov et al. [38], Wang et al. [51], Huang et al. [20], and Guo et al [16] design word-level sign language translation systems with the same modality. In these works, a Kinect camera was used and complex algorithms such as Open Flow [36] were used to capture and match skeleton data with RGB videos. These works represent meaningful progress in sign language translation research, but the ultimate goal of such a system should to be fully mobile and independent, with no external sensing equipment.

• **Other Meaningful Efforts**: Besides the dominant trend of using visual information, another stream of recent research exploit other modalities for sign language translation, such as CSI capable WiFi or mmWave radios. Signfi [34] removes the need for carrying an additional sensor device and uses WiFi signals to capture hand gestures and classifies up to 150 ASL words. mmASL [44] utilizes signals from millimeter-wave radios to capture different upper body gestures for detecting and classifying 50 different words. While these systems neglect the need for a user-carried platform, they require special equipment to be installed in the target environment, which is another obstacle for practical use.

Apart from these systems, *SUGO* exploits only the depth video information, which is available on most recently released smartphones. Such new data modality collected from a smartphone's depth camera has been recently applied in designing novel application systems such as Hand gesture recognition [26], authentication [59], and mobile augmented reality (AR) [12]. Exploiting depth videos allow for the ubiquitous use of sign language translation independent of the environment, and without additional hand-carried sensors. To our knowledge, this is the first work to exploit only the smartphone's depth camera information for offering full on-device sign language translation. For sign classification, *SUGO* incorporates a 3DCNN-based deep learning model architecture, which has been shown recently to be extremely beneficial in gesture recognition systems due to their capability of processing image sequences on the timescale [31]. Specifically, as Table 1 shows, unlike many previous works, our work targets to operate solely using the smartphone's internal computing resources without any assistance from external computing platforms.

3 SYSTEM OVERVIEW

3.1 Design Goals

In formulating our system design goals, we held an interview session in the form of unstructured open discussions with four sign language users, a sign language interpreter, and one researcher from our team. The main focus of the interview was to extract user-requirements on a smartphone-based system for sign language translation. Upon starting the discussions we asked the sign language users about their experiences with previously accessed sign language translation systems and what they expect from a smartphone camera-based system. Three common issues raised from the interviewees were that (1) currently accessible research prototypes induce too much translation latency and show low accuracy for specific words such as compound words, (2) they did not want to carry additional devices for the purpose of increasing the accuracy, and (3) they felt reluctant in using video-based systems as it would reveal the environment they are located in to external parties - mostly caused from the distrust of service providers. The 2 hour long interview also lead to discussions on the diversity and motion complexity of sign language words, while discussing on the potential limitations that a smartphone-based system may possess. The initial interview with the four participants was followed by an additional interview with a sign language instructor, which we used to confirm and formalize the outcomes from the earlier open discussion format interview.

Based on the user requirements extracted from the interviews, we set the design goals for our proposed system as the following.

(1) **Depth Video Usage**: The system should use only the depth video data from the smartphone as its input. This is to minimize potential user-perceived privacy concerns from using RGB videos in third party apps and to reduce model complexity. Our discussions with sign language users suggest that they feel uncomfortable with capturing RGB images since it would expose their surroundings and faces to a third-party app.
(2) **On-device Operation**: We target to use the smartphone as the only device in the system. The system should not incorporate data from external sensing platforms and perform all computational operations on the smartphone device, rather than offloading the computation to remote servers.
(3) **Latency**: Sign language translation, should execute within minimal latency on resource-limited smartphones. To practically use the system in real-world scenarios, the latency introduced from the translation process should be kept low enough to not interfere with daily conversations.
(4) **Word Scalability**: Typical sign language translation systems focus on designing an inference model for a set of words. This is understandable, given that it is difficult to cover several thousands of words in a sign language vocabulary set. Nevertheless, the model should be general enough to interpret a comprehensive set of words with various gestures.

3.2 Exploiting Depth Videos for Sign Language Translation

While using depth-only videos offers several application-level and usage benefits compared to RGB or RGB+Depth-based systems [38, 43], on-board smartphone depth cameras have several fundamental limitations for sign language translation. Specifically, infrared (IR)-based depth sensors (e.g., Apple TrueDepth) can perform poorly under bright sunlight since the solar spectrum includes interfering IR wavelengths. As a preliminary study in Figure 1, we present an image of a person performing a gesture in a sunny environment and compare the the same gesture in an indoor environment. In bright environments, the edges (of the person and hand) in the image are less precisely expressed (in some cases even missing) when compared to the indoor environment, which can lead to difficulties in tracking subtle hand movements. Furthermore, low-cost time-of-flight (ToF) sensors, used by some recent smartphones such as LG G8 ThinQ, ill-perform under bright light conditions [25]. Stereo camera-based depth-sensing modules may perform well when bright, as they exploit the disparity between multiple RGB cameras. However, they are limited in dark environments as a clear multi-dimensional image is

(a) Outdoor sunny environment (b) Indoor office environment

Fig. 1. Depth images captured in indoor and outdoor environments using the iPhone X's TrueDepth camera from a single participant making the same sign gesture. Data collected in an sunny outdoor environment is shows less clear edges compared to the data collected in and indoor environment.

required. These practical issues suggest that a depth camera-based system should be designed (or should confirm their performance) for environments under various lighting conditions.

From Figure 1 (b), we can also notice that, despite being clearer than the outdoor case, even in indoor environments, the IR camera introduces noticeable noise compared to an RGB image. This is mainly a performance limitation of low-cost IR-depth cameras equipped on smartphones and suggests the need for a robust inference model that can tolerate such input noise.

Depth videos can also be more vulnerable to different types of motions compared to RGB videos. Specifically, suppose the camera or target object dynamically moves during a video sequence (i.e., a device's relative position to the target object changes). It can be challenging to accurately segment and track objects across multiple frames due to the lack of contextual information. However, for understanding sign language, continuously tracking small movements of the human body (e.g., changes to finger movements or body posture changes) is essential. Thus, to solely use depth videos under such dynamic conditions, additional image processing schemes are required to overcome these contextual limitations.

3.3 System Overview

Considering such issues in exploiting depth videos for sign language translation, we design our proposed system, *SUGO*, as the following. We present an overview of the *SUGO* operations in Figure 2. When *SUGO* is in operation (e.g., dotted lines in Figure 2), we first pass all input videos to the Human Gesture Emphasis module (Sec. 5.1) to remove the background pixels and emphasize the hand gestures, and feed them to the 3DCNN-based inference model (Sec. 5.2) after performing word segmentation (Sec. 5.5). All operations here take place *on the smartphone*, from video capture to sign classification.

When training the 3DCNN inference model (e.g., solid lines in Figure 2), we collect data samples and perform an additional phase of generating an augmented dataset, which includes data mimicking the motion noise that can be introduced in real-world usage scenarios (Sec. 5.4). Once the model is trained, we perform operations to lighten the model, including filter pruning and weight quantization, to reform the model to be suitable for resource-limited mobile platforms (Sec. 5.3). The following sections present details on the data collection process and the system design.

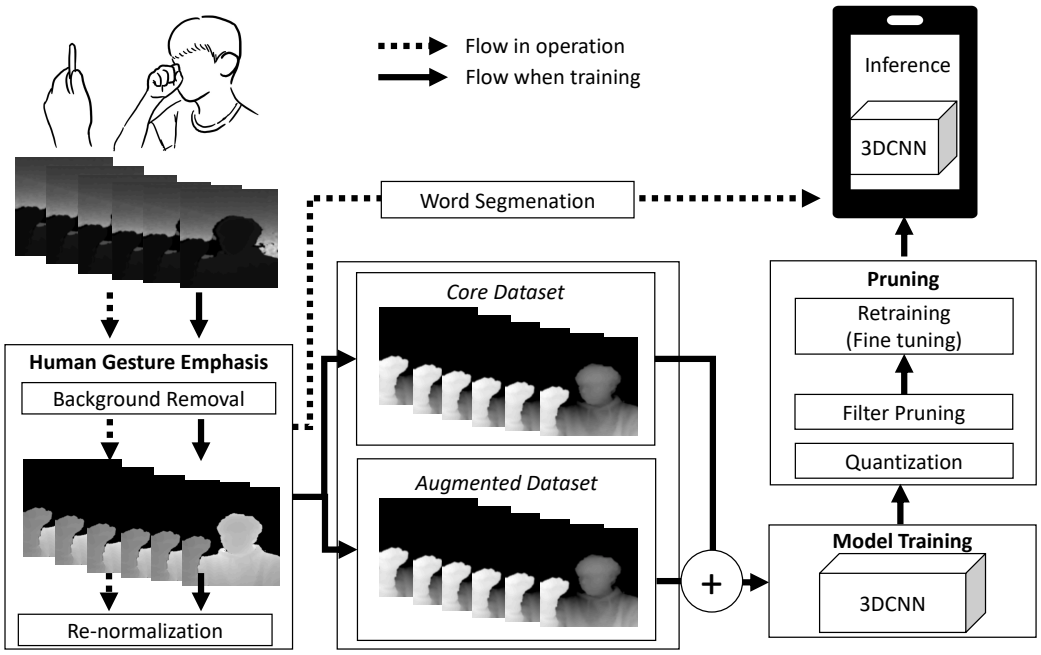


Fig. 2. System overview of *SUGO*. Dotted lines present the operational flow and the solid lines show the flow taken in the model training phase. All inference operations, from data collection to sign language translation, takes place on the user's smartphone. *SUGO* performs human gesture emphasis on the input images and the 3DCNN-based inference model is trained using the core dataset collected for 50 KSL words along with an augmented dataset for robustness towards motion artifacts that occur in real-world use cases.

4 DATASET

4.1 Word Selection

To train and test our inference model, we created a depth video dataset of 50 different words from the "daily life sign language vocabulary set" available in the Korean sign language (KSL) dictionary. Unlike many previous datasets containing words from widely used sentences/phrases (or with unknown selection criteria), our dataset differs in that we focus on the word gestures' *motion characteristics*. In other words, we design a dataset comprehensive enough to cover the various types of motions/gestures that KSL words (similar to many other sign languages) encompass. We choose the following criteria for dataset word selection based on previous literature offered by the National Institute of the Korean Language [52]. As a result, we construct a KSL dataset of 50 words that comprehend different criteria as below.

- **Hand gesture positions**: Some signs are performed near the face, and some are made near the body. In KSL, the location of the singer's hand can be located near the head, body, or both. In detail, the gestures performed near the head is sectored as the entire face, forehead, temple, eyes, ears, mouth, cheeks, nose and chin areas. Furthermore, gestures made around the body area is sectored as the whole body, the front and back of the neck, shoulders, chest, and abdomen. We therefore chose words with such categories inconsideration. For example, raising the right fist to the nose (i.e., face) indicates the word "right" (word #0), and placing the right palm in the center of the chest (i.e., body) means"me" (word #6). There are also

Table 2. 50 different KSL words included in the core dataset with respect to where the motions are performed.

Hand gesture positions	Words
Near the face	Like, Eat, Right, None, Mental, Can, Smile, Age, High, Boring, Wonder, Poverty
Near the body	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, Fun, Entertainment, Power, What, Where, Will, Income, Motorcycle, Hello, Game, Name, Best, Place, Person, Mind, Only, You, This, Action, Go out, Play, Me
Both	Coffee, Poor, Ability, Nice to meet, Ask, Shock

Table 3. 50 different KSL words with respect to the number of hands used for the gestures.

Hand Usage	Words
Single-hand	Like, Eat, Right, None, Me, 1, 2, 3, 4, 5, 6, 7, 8, 10, Mental, What, Where, Power, Can, Ability, Age, This, You, High, Boring, Name, Best, Ask, Wonder, Only, Place, Poverty
Two-hand	Go out, Coffee, Will, Play, Fun, Entertainment, Smile, Nice to meet, Game, Action, Income, Motorcycle, Hello, Shock, Person, Poor, Mind

Table 4. Word groups with motion/gesture similarity.

Group	Words	Group	Words
Group 1	Me, 7, 8, 9	Group 5	Motorcycle, Hello
Group 2	Name, Best	Group 6	This, You, Action
Group 3	Ask, Shock	Group 7	Fun, Entertainment
Group 4	Wonder, Only, Boring	Group 8	1, 10, What

Table 5. 50 KSL words classified with respect to their gesture/motion complexity.

Gesture complexity	Words
Simple	Like, Eat, Right, Me, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, Mental, What, Where, Will, Power, Can, Ability, Play, This, You, Motorcycle, Hello, Place
Complex	None, Go out, Coffee, Fun, Entertainment, Smile, Nice to meet, Game, Age, Action, Income, High, Boring, Name, Best, Ask, Shock, Wonder, Only, Poverty, Person, Poor, Mind

more complex words such as “ask” (word #41) that require the gesture to start at the head position (i.e., face) and move down to the chest (i.e., body). Table 2 summarizes the 50 words in our dataset in these three categories.

- **Single-hand and two-hand gestures**: Some sign gestures are expressed with a single hand, and some require both. We point out that a major characteristic of KSL is that there are many different two-handed sign language words with various cheremes mixing the dominant and non-dominant hand gestures. Our dataset shows a mixture of these two types of words. As a representative example, the gesture that means “ask” (word #41), described earlier, is a word that uses the right hand, in which the user points the index finger to the temple, and as the hand lowers down to the chest, all fingers open and the palm faces upward. A word with similar motion, but using both hands, “shock” performs the same operations as “ask” but as

Table 6. Compound words included in our dataset with their respective element words.

Compound word	Element words	Compound word	Element words
Coffee	Nose + Tea *(not in dataset)*	Nice to meet	Smile + Fun
Where	What + Place	Game	Entertainment + Fun
Ability	Can + Power	Poor	Poverty + Human

the right hand lowers to the chest, the left hand acts as the bottom floor and the gesture ends with the back of the right hand strongly hitting the palm of the left hand near the chest. We summarize how different words are signed in Table 3.

- **Motion similarity**: KSL consists of a total of 30 core hand shapes. Small gesture variations from these core gestures and the position in the head/body of where the gesture is performed leads to completely different meanings. Therefore, many words will have similarities and classifying them can be a challenging task. In our dataset, we select eight groups of words that are similar in terms of the motion. We intentionally selected them to validate the effectiveness of sign language translation systems in practical usage scenarios. For example, in KSL, putting your index finger on your chin while you bounce your middle finger with the thumb represents the term "only" (word #44). The same motion, but when done near the nose, expresses "wonder" (word #43), an entirely different meaning with similar motions. Furthermore, the same motion near the side of the head (at the temple) means "boring" (word #36). Such words suggest that systems focusing on hand gestures only [15] (without observing the upper body as a whole) may fail to scale to a diverse vocabulary set. Table 4 presents how we group such similar words.
- **Gesture complexity**: Given the diversity of sign language gestures, some gestures are simple (e.g., static finger representations), while some are more complex (e.g., fast and dynamic movements). KSL gestures include path movements and articulator-internal movements that modify hand shapes or directions by moving wrists or finger joints. Furthermore, there are many words that require the folding, bending or twisting of the fingers. Words expressed in this form are considered complex. We make sure that our dataset includes both types of gestures (simple and more complex). Among the 50 words in our dataset, we include 23 complex hand gesture sign words and 27 simple hand gesture sign words. In addition, we mix gestures that include finger occlusions, which are challenging for camera-based sign language translation systems. In Table 5 we classify the 50 words as either a simple or complex word.
- **Compound words**: Sign language (similar to verbal language) includes compound words, which are combinations of multiple words. Among words registered in the KSL Dictionary, there are 2,606 single-word words and 5,916 compound words [52]. Therefore, compound words occupy a very large portion of the entire word set. For example, in KSL, the word for "nice to meet" (word #28) is a mixture of "smile" (word #27) and "fun" (word #25) signs. In our dataset, we include six compound words. Table 6 presents different compound words included in our dataset with their respective element words.

We emphasize that the words in our dataset were carefully selected to evaluate the performance of *SUGO* for various sign language vocabulary's motion characteristics. By focusing on the motion characteristics, we try to confirm that *SUGO* can scale to new words that are currently not included in our dataset. High classification accuracy for limited types of gestures could not necessarily indicate the robustness against the words with different motion characteristics.

4.2 Data Collection Process

4.2.1 Core Dataset (Training and Testing). Our *core dataset* consists of data collected from 20 participants (12 male and 8 female; age range 13-52), in which all participants performed each of the 50 words for five times. All videos

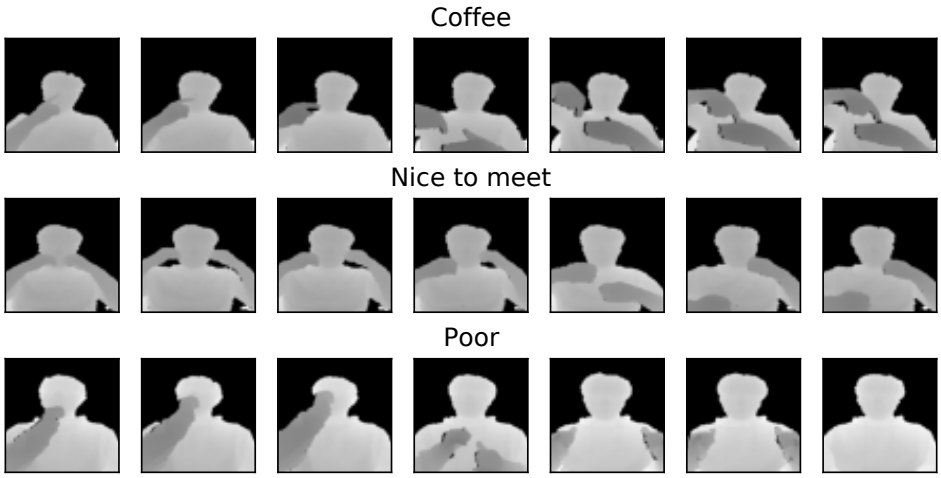


Fig. 3. Sample word sequences for three different sign gestures. A sign language translation inference model should extract separate features from each image and these features should be effective in understanding the context over multiple consecutive frames.

were captured at 8 frames per second. In total, our dataset consists of 5,000-word sequences encoded as ~450,000 depth video frames. Two of the 20 participants received professional KSL training, and the other 18 were assisted with videos offered from the web-based KSL dictionary from the National Institute of the Korean Language (http://sldict.korean.go.kr/). We provided enough time for the subjects to get used to each of the 50 words before recording. Data collection for each subject took approximately 1.5 hours in a research lab environment where we mount the smartphone on the desk with the front camera facing the participant's face and upper body. We used an iPhone X and recorded the video from its TrueDepth camera and the RGB camera. We used the RGB videos only for verification purposes to better understand the performance of *SUGO*. Specifically, they were used to understand ground truth human gestures for analyzing misclassified cases in our evaluation and in designing *SUGO*'s core deep learning model. The data collection process and all experimental procedures presented in this work have been approved by our Institutional Review Board (IRB).

4.2.2 Multi-environment Dataset (Testing). In addition to the core dataset, we gathered an additional *multi-environment dataset* of 10 different words that sum up to 2,000 word sequences (200 sequences per word; 48,000 frames in total). We collect the dataset under different environmental conditions (e.g., bright daylight, dark outdoors) or with added motion artifacts (e.g., hand-held walking and in-vehicle recordings). Specifically, for collecting data with different motion artifacts, we ask the participants to i) stand still and hand-hold the camera while making single hand gestures with the other, ii) perform the same single hand gestures while in moving vehicle's passenger seat with the smartphone hand-held, and iii) make the same single hand gestures while walking. We used this dataset as a validation set to confirm the performance of *SUGO* in practical usage scenarios. We will later discuss the classification results for this data in Section 6.2.

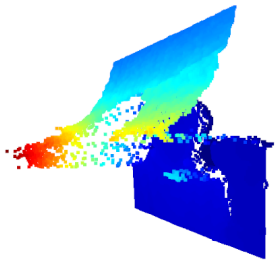
5 SUGO - SMARTPHONE DEPTH CAMERA-BASED SIGN LANGUAGE TRANSLATION

We now present details on the system design of *SUGO*. Specifically, to achieve the goals mentioned in Section 3, we develop three major software components: (1) a Human Gesture Emphasis module for re-normalizing an

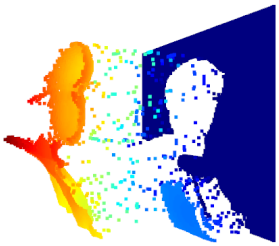
(a) 2D representation of depth image without human gesture emphasis.

(b) 2D representation of depth image with background removal applied.

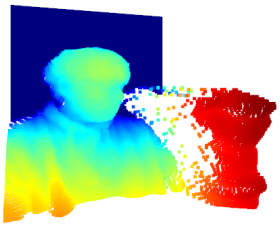
(c) 2D representation of depth image with background removal and re-normalization applied.



(d) 3D representation of depth image without human gesture emphasis.



(e) 3D representation of depth image with background removal applied.



(f) 3D representation of depth image with background removal and re-normalization applied.

Fig. 4. Impact of the Human Gesture Emphasis module (background removal and re-normalization) on input frames. The Human Gesture Emphasis module in *SUGO* successfully removes the background and emphasizes the hand and upper body gestures, which is important in overcoming the unique noise that low-cost depth camera-based videos introduce.

input depth video frame, (2) a 3DCNN module for sign language-text inference, and (3) a Word Segmentation module to assist word detection in multi-word embedded videos. Furthermore, *SUGO* includes techniques such as data augmentation, weight quantization, and model pruning to add robustness and lighten the 3DCNN model so that it meets the system-level requirements. The following sections present details on each of these components.

5.1 Human Gesture Emphasis Module

The human gesture emphasis module takes in raw depth videos from the smartphone's depth sensor and performs a set of operations to prepare the frames for an ideal format for the inference model. The goals here are to (1) remove background noise, and (2) emphasize the human gestures so that the core contents of the images can be effectively exploited for sign language translation.

When a depth frame is captured for sign-language translation, we can expect the person's face and upper body to consist most of the image, with some parts of the image including contents such as the environmental background. Depth data for such background regions are merely useful in sign language translation, and we are more interested in capturing detailed motions near the human body and face.

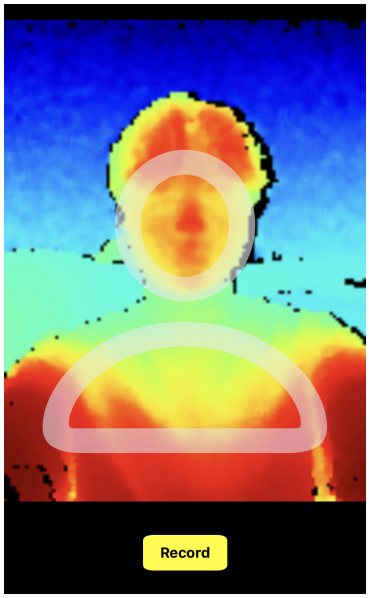


Fig. 5. Data collection app screenshot used for core dataset collection. The app provides guidelines on where the subject should position their face and chest. Similar visual guidelines are used for sign language inputs in the operational app.

To perform background removal, it is important to identify where the subject's face and upper body regions are located. We take a simple yet effective approach for this purpose. Specifically, as Figure 5 shows, we present visual guidelines on the smartphone app to assist the users in placing their face near the center of the image. Using this information the human gesture emphasis module takes the depth value from the image's center pixel and re-configures the depth. This results in modifying the maximum depth to be 15 cm deeper than the center point's depth (typically 80 cm from the camera) and the minimum depth near the camera module itself. Using 8-bit depth representations, each bit will correspond to approximately 3 mm differences in depth. Based on this information, the human gesture emphasis module performs background removal by setting all pixels farther away than the maximum range to 0.

Next, for emphasizing the human-related contents within the image, the human gesture emphasis module re-normalizes the parts of the video frame that contains potential human body parts. Specifically, in this phase, we try to disregard the depth values of the background, and make sure that the body-to-hand depth range covers the full depth spectrum. Given that the depth differences between the background and body is typically larger than the difference between the body and hand, such a re-normalizing approach allows the depth difference between the body and hand gestures to be much more emphasized than the body-to-background distance. For example, given a depth range of 0-1000, in the original image, the hand may be positioned at 150, the body at 300 and the background at 800. When trying to capture features based on depth differences, since the distance between the body and background stand our the most, the 150 difference between the body and hand, where in fact most of the sign language-related motions take place, may not be properly emphasized. Therefore, when re-normalizing we extend the depth range between the hand and body to a wider range, say 0-800. When done so, the difference between the background and body is de-emphasized and depth changes are mostly captured between the body and hand.

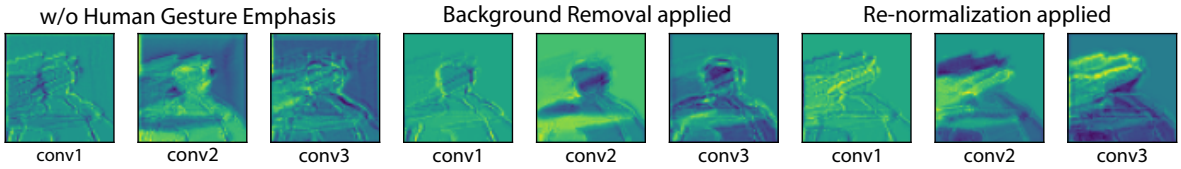


Fig. 6. Feature map plots of the first, second, third convolution layer of *SUGO*'s deep learning model. The Human Gesture Emphasis module emphasizes the input images so that more attention is given to the user's gestures.

As we exemplify in Figure 4 (a) and (d), in 2D and 3D respectively, an image before applying the human gesture emphasis module is noisy, and the features made from the user's hand cannot be seen clearly. As the image passes the two phases of the module (e.g., background removal and re-normalization), the body and hand features gain clarity (Figure 4 (b), (e) and (c), (f), respectively). Note that Figures 4 (d), (e), and (f) present the 3D illustration of the frame as we apply each of the components in the human gesture emphasis module.

The initial convolution layers in a CNN typically serve as an edge detector, and the whole point of *SUGO*'s human gesture emphasis module is to assist this edge detecting process. As an example, we show in Figure 6 we present the feature maps for three different convolution layers for different approaches. Note that the user in this video frame is moving the right hand towards the camera. We can see by comparing the figures that when the re-normalization is applied, the hand gesture is best recovered. Given that this re-normalization emphasizes the difference in depth for the regions between the body and hand, while nullifying the impact of the background, the convolution layers can detect a more crisp edge compared to the other cases.

We will later show in the evaluations that the use of the human gesture emphasis module is essential in achieving high classification accuracy when using depth images for sign language translation.

5.2 Sign Language Translation Model

Unlike many existing image classification applications, translating sign language gestures involves understanding the context for a *sequence* of frames. Given a frame sequence that consists of sign gestures, useful features need to be extracted from each video frame, and the sequence of these features should be used to classify a word represented by the gesture. Figure 3 shows sample frame sequences of depth video frames after the human gesture emphasis phase. A model should extract proper features from each of these frames and make sure that the features are useful in understanding the context over multiple consecutive frames. For this reason, we need an inference model that can process and extract both single image-based features and sequential features. Typically, a Convolutional Neural Network (CNN) architecture is used to extract features from images [4], and a Recurrent Neural Network (RNN) structure is applied when understanding sequential information [42]. Due to such characteristics, to understand video data, there have been some attempts of inputting features obtained through a CNN into an RNN model [16, 29, 46]. However, such an approach can be challenging given that, ideally, the features of a frame extracted by the CNN, should not focus on the per-frame characteristics, but output features that are effective for time-series analysis by the RNN. In other words, the quality of the per-frame features extracted by the CNN model can only be determined to be of good quality if the RNN makes accurate predictions. This requires careful engineering of the loss function that both the CNN an RNN can share, which is a engineering challenge by itself. As a result, for sign language translation, many studies focus on manually focusing on hand-picked features such as the skeletal features extracted from RGB(+D) videos or external sensing platforms [10, 11, 14, 57].

In *SUGO*, we take a different approach and exploit a 3 dimensional CNN (3DCNN) architecture for video processing. A 3DCNN-based deep learning model is similar to a typical 2D CNN, but it uses a 3D version

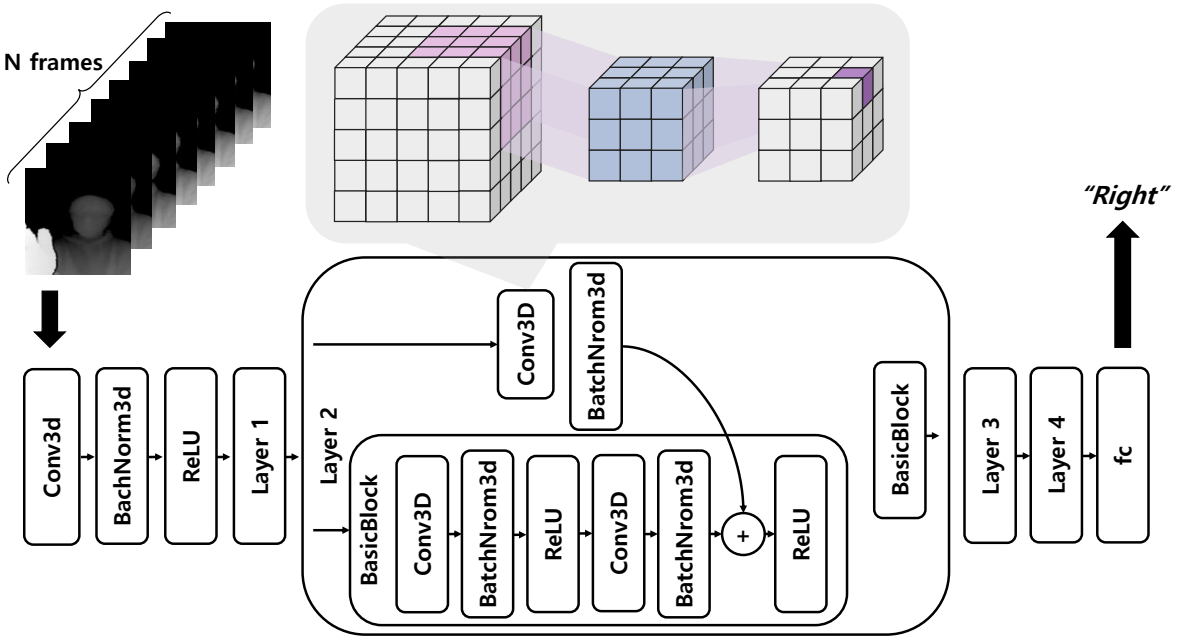


Fig. 7. Inference model architecture used in *SUGO*. *SUGO* exploits a 3DCNN network architecture as its base model given that the 3DCNN offers both per-image feature extraction and feature sequence-based classification via a single pass of the model. The illustration is the final network architecture of *SUGO*'s inference model, after the pruning process.

convolutional layer. Note that a 2D convolutional layer is effective in extracting features from a 2D image, whereas, in 3DCNNs, the third dimension can be used to learn features in the time domain (i.e., over a sequence of 2D images). Thus, 3DCNNs can be designed to allow the extraction per-frame features that are meaningful on a multi-frame (i.e., time) perspective. The selection of using a 3DCNN-based model was also influenced by recent work in human activity recognition and motion classification, which acknowledged the effectiveness of applying 3DCNN for similar applications [9, 31].

As the base inference model architecture for the 3DCNN, we use the ResNet-18 model, which is known to provide satisfactory feature extraction performance for human gestures and has been validated as an effective base model for 3D structures [17]. The ResNet3D model [17], the 3DCNN we use in *SUGO*, has the same structure as ResNet-18, except that it exploits 3D convolution and batch normalization operations. Using the ResNet-18, we pre-train this model with the Kinetics-400 dataset [24], which includes detailed categories for classifying various body parts, and re-train this model with our collected dataset. This re-training process can be considered as a simple form of transfer learning to exploit the pre-trained weights for the 3DCNN model using the Kinetics-400 dataset to suit our application's purposes [49].

5.3 Model Size and Computational Cost Reduction

The 3DCNN model holds the advantage of analyzing a single video frame and, at the same time, can extract sequential features over consecutive frames. However, 3DCNN models are known to be resource-demanding and large [30], which contradicts our goal of operating our system solely on resource-constraint mobile devices. For

this purpose, we perform pruning on the 3DCNN model. Specifically, in *SUGO*, we apply filter-pruning [32] to the original ResNet3D-based 3DCNN architecture. The filter-pruning is a technique that reduces the memory usage of a model and also the computational overhead by eliminating less important filters on the final result. Since filter-pruning eliminates filters, it is important to understand the relationships between different layers of the model. Specifically, given the skip connection characteristics of the ResNet-18 model [18], the filter pruning should consider the input and output shapes of each layer. Thus, while there are 18 layers in ResNet-18, only four of these layers can be aggressively pruned. Overall, we present the final model architecture after the pruning process in Figure 7. Compared to the original ResNet3D model [17], our pruning approach effectively reduces the size and complexity of the model so that it is suitable for on-device operations. Later using Sections 6.1.4 and 6.4, we will discuss the impact of pruning on the model size and accuracy with respect to different pruning parameters.

In addition to model pruning, to further reduce the inference operation complexity, we perform weight quantization on the inference model [22]. Specifically, we convert all weight parameters in the 3DCNN to be in half floating-point units (i.e., `float16`) instead of `float32` to expedite the inference process and lighten the model complexity.

5.4 Data Augmentation

We train the 3DCNN model using a subset of data from our core dataset introduced in Section 4.2. The core dataset serves well as a representative dataset, but it is limited in covering various real-world scenarios; for instance, it is collected using a smartphone mounted to a desk to eliminate user hand vibrations' effects. To make the model more robust against input noise from various forms of user motions, we add an augmented dataset to diversify the coverage of the core dataset with realistic motion noise. While it is ideal to physically collect a large amount of data empirically, it can be difficult due to practical limitations.

Specifically, given the challenges discussed in Section 6.2, we move the human contents of an image vertically and horizontally (randomly) by $0 - 10$ pixels to imitate small vibrations that hand-held recordings can introduce. At each learning iteration, we randomly set the position of the human contents on the image to create an augmented dataset for model training. At each training phase, each original data (a frame from the core dataset) is augmented with an additional frame with the random changes. Note that data augmentation is only used in the model training phase and not for *SUGO*'s performance validation. Later, we will show the impact of data augmentation on how it improves the system's performance when evaluated for videos collected in real-world settings.

5.5 Word Segmentation Module

When multiple words are sequentially embedded in a single input video, *SUGO* should split the video frames into subsets so that individual words can be processed separately. However, splitting the video into units of sign words is a challenging task. While sentence construction using different natural language processing (NLP) techniques is out of the scope of this work, there is still a need to confirm that *SUGO* holds the capability to detect the boundary between consecutive word-level signs for proper word-level classification.

For this, we implement a simple word segmentation module where we structure the input video sequence so that the input video is split using a fixed-sized sliding window. Figure 8 presents an overview of this module's operations. Given that most sign language gestures take less than 1-2 seconds [7], and *SUGO* takes as input depth videos with a frame rate of 8 frames per second (fps), we conservatively take the first 24 frames. Note that signs can be shorter than 24 frames, but our goal is to make sure that our processing unit can cover the entire gesture and nothing is missed. If the video is less than this size, we take the full video from f_0 to f_{23} as the initial input. Next, the sliding window moves by α, and the input sequence becomes $f_{0+\alpha}$ to $f_{23+\alpha}$, and this process recursively

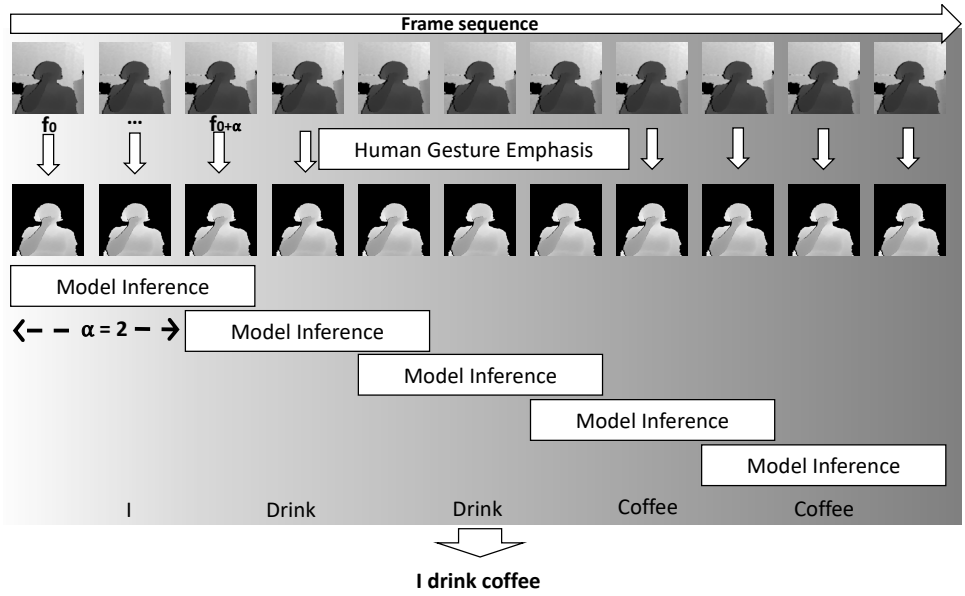


Fig. 8. Word segmentation in *SUGO*. For input sequences with multiple words embedded, *SUGO* performs inference operations based on a sliding window to capture possible words from the input data.

continues until the end of the video. By applying word segmentation, *SUGO* carefully examines at what sequence a gesture can be recognized as a (known) word and outputs the word with the highest possible match. Finally, we set a threshold of τ to filter out word estimations with low confidence since intermediate gestures can lead to classifying wrong words.

While this scheme effectively identifies multiple words in a single video sequence, this is a greedy approach. Therefore, the downside is that multiple rounds of 3DCNN operations are required to identify a single word from the video. For resource-limited platforms, this can lead to increased resource and energy usage. Moreover, the most critical threat of using such a scheme is that the latency of sign language translation can increase. If the inference latency of the 3DCNN model (δ) is high, since a chunk of the video is passed to the 3DCNN for every α frames, given an input video consisting of n words or fc_n frames, a latency of $\delta \times \frac{fc_n}{\alpha}$ is required to process the full word sequence. Nevertheless, a core requirement in *SUGO* is to support real-time daily conversations; thus, the translation latency should be kept minimal. Fortunately, by adjusting α, we can control this latency by skipping through frames, and as a positive side effect, reduce the computational overhead. We will later evaluate the accuracy of word classification on multiple word sequences in Section 6.5 and show the impact of α on the classification accuracy.

6 EVALUATION

To evaluate the performance of *SUGO*, we train the inference model using the PyTorch deep learning framework. Using the models, we perform experiments in different computing environments. Specifically, for accuracy-related experiments, we exploit a GPU server machine equipped with four NVIDIA RTX 2080ti graphics processing units (GPU). All other systems-related features are tested with four different resource-constraint platforms: i) the NVIDIA Jetson TX2 [2], ii) NVIDIA Jetson Nano [3], iii) Apple iPhone X and iv) the Apple iPhone 11 Pro. Details on these four device configurations will be presented in Section 6.4.

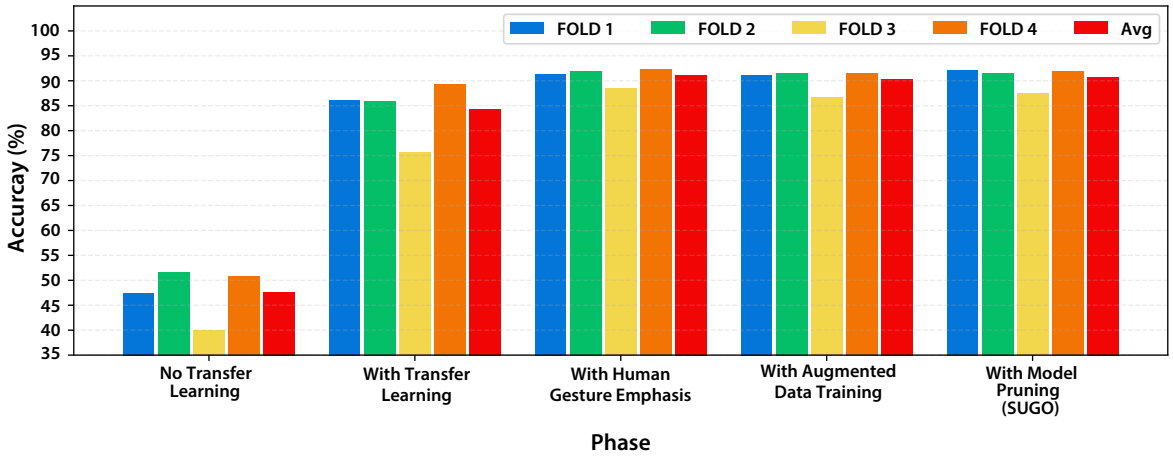


Fig. 9. Classification accuracy for different variations of *SUGO*'s inference model. Applying techniques such as transfer learning (using the Kinetics-400 dataset) and input video processing contributes to improving the classification latency. Furthermore, integrating augmented data to the training dataset and pruning the model does not harm the overall accuracy of the model. We will examine how these two schemes contribute to the robustness and light-weight characteristics of *SUGO* using following results. Note that the plots present changes to the performance in additive manner from left to right, in the order of presentation.

For model training, we select data from 15 of the 20 subjects from the core dataset and perform data augmentation for this training data, as presented in Section 5.4. The data for the remaining five subjects are *not* included in the training phase and isolated for testing only. Note that the data points in the testing dataset are excluded from the data augmentation phase as well. Finally, we note that the core dataset is used for both training and testing purposes unless we specify that a different dataset was used.

6.1 Word Classification Accuracy

We start our evaluations by examining the word classification accuracy from the depth videos captured from an iPhone X (i.e., the core dataset). We take a step-by-step approach in which we first examine the impact of transfer learning where we observe the performance of the original 3DCNN model trained using only the Kinetics-400 dataset [24] then add our training dataset to see how much improvement in performance we can achieve. Next, we examine the impact of human gesture emphasis and model pruning. For all experiments, we performed 4-fold cross-validation in which data from 15 of the 20 subjects were used for training. The other five are used as the testset, with each data being part of the testset only once.

6.1.1 Impact of Transfer Learning. To observe the impact of transfer learning, we evaluate two different models: (1) a raw ResNet3D model trained with our training dataset only, and (2) a ResNet3D model with pre-trained weights using the Kinetics-400 dataset [24] and re-trained with our training dataset. Note that the Kinetics-400 dataset consists of RGB images; thus, the weights are trained using three different channels (e.g., R, G, and B). Nevertheless, since our dataset of depth videos consist of only a single channel (i.e., depth), when initially training the weights using the Kinetics-400 dataset, we make sure that only the first layer of the ResNet3D model takes in three channels as input and the input is reduced to a single channel from the second layer onwards. In operation, we point out that when using the depth-only videos, there is only a single input channel. Note that we empirically

Table 7. Impact of different model pruning cases and weight quantization on the memory usage and model size. The filter pruning and weight quantization methods used in *SUGO* contributes to significantly reducing the model size, making it suitable for on-mobile device operations.

Test case	Model pruned ratio	Memory usage float32 (MB)	Memory usage float16 (MB)	Model size float32 (MB)	Model size float16 (MB)
0	original	473	237	127	64
1	(0.1, 0.1, 0.1, 0.3)	387	194	74	37
2	(0.2, 0.2, 0.2, 0.4)	335	168	56	28
3	(0.25, 0.275, 0.25, 0.45)	304	152	48	24

confirmed that using the original three channels by stacking the depth image did not show statistically different performance. To perform 4-folds validation, each of the two test models was trained four times separately with different training sets.

In the first two plots of Figure 9 (i.e., "Without Transfer Learning" and "With Transfer Learning"), we present the classification accuracy of these two test cases. Notice from these plots that applying transfer learning (i.e., initial weights configured using the Kinetics-400 dataset) nearly doubles the classification accuracy from an average of 47% to 85%. This suggests that applying transfer learning to pre-train weights in the inference model significantly contribute towards achieving a very high classification accuracy.

6.1.2 Impact of the Human Gesture Emphasis Module. We now examine the performance of *SUGO without* the human gesture emphasis to show its impact on the classification accuracy. By default, all inputs in *SUGO* pass through the human gesture emphasis module introduced in Section 5.1. However, we note that the "With Transfer Learning" plots in Figure 9 are results when this phase is neglected, and in the third plot of Figure 9 (i.e., "Without Human Gesture Emphasis"), we show the performance of *SUGO* when input depth videos are passed through the human gesture emphasis module. Results suggest that the human gesture emphasis module helps achieve classification accuracy gain of ~7% (from 85% to 91% - note: changes in the schemes in Figure 9 is additive in the order of presentation). The main reason behind this improvement is that the resulting frames from the human gesture emphasis module effectively separates background context from the foreground and emphasizes sign language motions with improved feature extraction.

6.1.3 Impact of Data Augmentation. While we will examine the detailed impact of different motions on *SUGO*'s classification performance in Section 6.2, in the fourth plot of Figure 9 ("With Augmented Data Training") we present the model's classification accuracy after adding the augmented data as part of the training dataset. For this we add augmented data for the training dataset (e.g., data from 15 subjects) as part of the training data. Note that the test dataset here is from the same core dataset, and we do not add augmented data to the test data. The results indicate that additional augmented training data only showed 0.3% degradation compared to the "With Human Gesture Emphasis" plots. In any case, the augmented data was added to make *SUGO* resilient towards motion-introduced noise, and we will show later that the use of augmented data for training helps achieve this goal in practical use case scenarios.

6.1.4 Impact of Model Pruning. Next, we apply filter pruning and weight quantization on *SUGO*'s inference model and examine the memory usage and model size in Table 7. Here, we can see that for different pruning cases (target pruning ratio presented for each of the four adjustable layers of the 3DCNN model) and weight quantization, the model size and the memory usage decreases noticeably, making the inference model suitable for operation on resource-constraint mobile platforms. For each different pruning and weight quantization test case in Table 7, we present the top-K classification accuracy in Figure 10. Notice that using `float16` reduces the

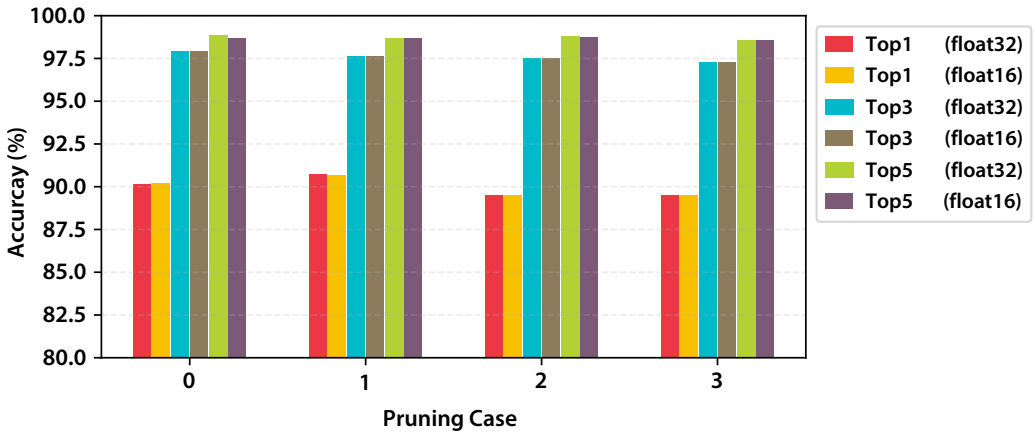


Fig. 10. Top-K accuracy for different pruning and weight quantization test cases. Model pruning and weight quantization does not affect the overall classification accuracy.

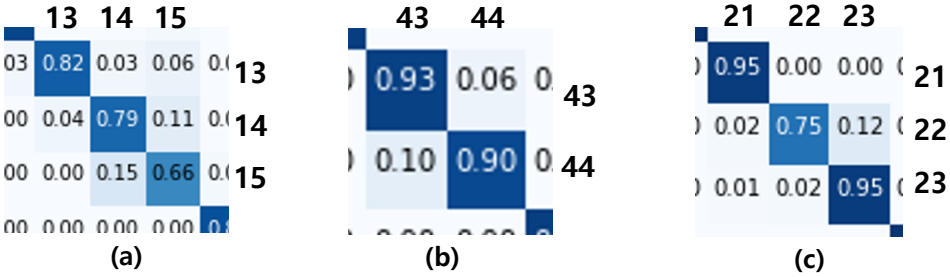


Fig. 11. Confusion matrix snippets for three points discussed in text (x-axis: predicted label, y-axis: ground true label). Detailed discussions are in Section 6.1.5.

model size and memory usage, but does not degrade the classification performance. Furthermore, filter pruning does not degrade the classification accuracy. Given the resource constraints of our target smartphone (iPhone X), we select to use test case 1 for the remainder of the evaluations, which offered almost identical performance compared to the non-pruned case, with nearly 20% reduction in memory usage and more than 40% reduction in model size. We point out that iOS does not yet fully support deep learning models with `float16` weights (details on Sec. 5.3). Thus, for the iPhone implementations, we keep `float32`.

In the final plots presented in Figure 9 ("With Model Pruning"), we present the 4-folds evaluation on the classification accuracy for comparisons with the previous model configurations. As the plots show, we see no significant performance degradation, suggesting that applying model pruning is effective in reducing model size while maintaining model accuracy.

6.1.5 Analyzing the Confusion Matrix. Figure 11 highlights three different regions on the final confusion matrix that show relatively lower classification performance. As aforementioned, our dataset was designed so that we intentionally include various challenges to the inference model. Specifically, Figure 11 (a) presents three words #13, #14, #15, which represent the sign gestures for the number 7, 8 and 9, respectively. These words commonly

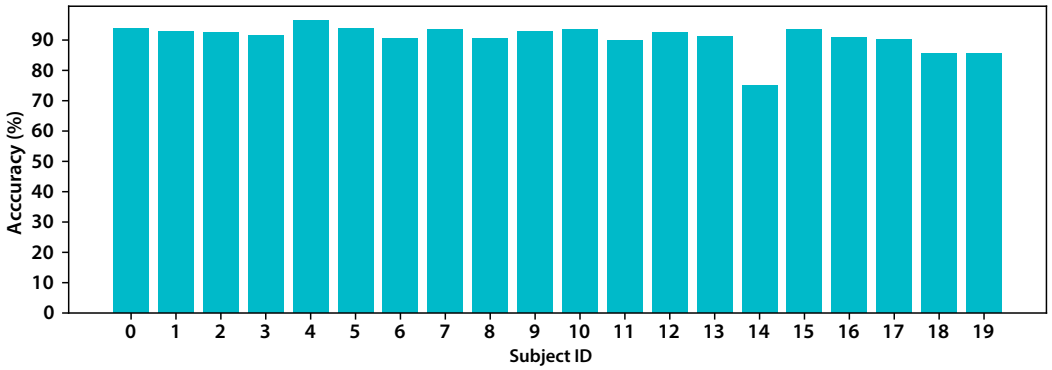


Fig. 12. Accuracy among different subjects. The per-subject results suggest that *SUGO* can be applied to new users with no prior learning of the user's sign gestures.

share the gesture of showing the back of the right hand to the camera with the palm opened towards the chest. For word #13, the thumb, index, and middle finger are open with the other two fingers closed, while word #14 additionally opens the ring finger, and word #15 opens all fingers. Therefore, the gesture difference among these motions is minimal.

Deep investigations into our dataset revealed that a few non-professional participants failed to fold and open their fingers properly when making these gestures, which led to misclassification in some cases. Nevertheless, word similarity was not always problematic. For example, take words #43 and #44, which represent "wonder" and "only". These gestures both take place near the face, but one gesture puts the hand near the chin, and the other makes the same hand gesture but near the nose. As Figure 11 (b) shows, these two cases were successfully classified. We note that compared to the first example, in this case, we could see that most participants properly performed the gestures. Next, we note that words #3, #4, #33, and #34 (not explicitly highlighted) each represent "none" , "go out", "action", and "income". These words are an example of complex gestures, but the confusion matrix results suggest that the complexity of sign language gestures only minimally impacts classification accuracy.

Lastly, in Figure 11 (c), we present an example of a compound word. The word #23 ("ability") is a combination of words #22 ("can") and #21 ("power") taking place sequentially. We noticed that the classification accuracy of #23 is high but observe 12% of word #22 being incorrectly classified as #23. Word #22 is a gesture where the right hand is fully open towards the mouth and moves farther away from the mouth with no changes to the hand postures, and #21 is a motion where the users make a fist with their right hand pull the fist towards the chest. When these two gestures are combined sequentially, it becomes the gesture for #23. By analyzing the data, we noticed that when users recorded videos for word #22, they put down their hand after the gesture, and some of these motions showed similarity with #23. This caused the inference model to make improper decisions leading to wrong classification results.

Overall, *SUGO* is capable of classifying the words correctly, but the lack of well-captured sign gestures lead to limitations on the classification accuracy in some cases. Nevertheless, with a larger training dataset, we believe that such limitations can be overcome. Furthermore, while out of the scope of this work, a natural language processing module that constructs full sentences with "candidate words" offered by *SUGO* can help enhance the overall sign language translation performance.

Table 8. Classification accuracy with respect to different lighting conditions. The human gesture emphasis module effective adds robustness towards variation in lighting conditions and helps maintain a high classification accuracy.

Environment	Location	Brightness (Lux)	IR-level	Accuracy w/o HGE Module (%)	Overall Accuracy (%)
Indoor	Office	1077	188	73.36 (14.26)	91.31 (3.19)
	Bright Setting	2163	295	50.45 (4.46)	95.51 (4.07)
	Dark Setting	1	0.0	48.30 (15.17)	90.08 (7.17)
Outdoor	Daytime	65535	30905	63.10 (14.67)	92.19 (4.25)
	Dark Night	82	12	56.16 (13.49)	93.36 (4.99)
	Night w/ Street Lamp	1	0	58.78 (1.91)	90.69 (8.96)

6.1.6 Per-participant Classification Accuracy. Finally, we present the per-participant classification accuracy in Figure 12. For this experiment, we train the model (the final version including pruning) with samples from all study participants in the core dataset *except* for the target test subject; thus, we train 20 different models, one for each test subject (i.e., data from 19 subjects as training data and the remaining one's data as test data). Figure 12 indicates that the classification accuracy over different participants is relatively even, except for cases such as participant #14. We noticed that the depth video samples produced by this participant were made unclear even with the human eye for some signs, especially for the signs that introduce motion similarity. Therefore, a large number of misclassifications occurred in such cases. Nevertheless, this result suggests that the inference model designed for *SUGO* is suitable for delivering satisfying performance for newly participating subjects without any personalized prior learning of the participant's sign gestures.

6.2 Applicability in Real-World Use Cases

In this section, we quantify the impact of such real-world environmental variations using the multi-environment dataset collected under different conditions (e.g., changes to external lighting and different motion factors that can affect the input video quality). We point the readers to Section 4.2.2 for details on the multi-environment dataset.

6.2.1 Impact of Lighting Conditions. Note that *SUGO* uses depth video samples collected from an iPhone X, which offers IR-based depth measurements. While IR-based depth videos capture accurate depth information, they can be vulnerable to external IR input sources and their variations. Thus, there can be questions on how the IR depth camera-based sign language translation can operate under bright sunlight conditions. Given that *SUGO* targets to be applied to everyday conversations, it is important to validate its performance in such cases.

We perform experiments in two different physical environments: (1) indoors and (2) outdoors. For each environment, we test for three different lighting conditions, as summarized in Table 8. The classification results, also presented in Table 8, suggest that despite the differences in overall brightness and IR levels, the classification accuracy is kept high under any lighting conditions. By comparing the accuracy results in the final two columns of the table, human gesture emphasis (HGE) plays an important role as the usage environment differs from a typical indoor office environment. Specifically, we can notice that applying human gesture emphasis adds robustness to *SUGO* towards light condition changes.

6.2.2 Impact of Human/Environmental Motion Artifacts. We now introduce different types of motion artifacts that the smartphone can encounter in daily usage scenarios. Specifically, in this set of experiments, we no longer mount the camera to a stable location, but rather test for different cases while holding the camera in one hand. For this reason, the samples collected for these experiments were sign language words that could be done with

Table 9. Classification accuracy with respect to different real-world motions. Adding augmented data in the model training phase contributes to keeping a high classification accuracy even when motion artifacts persist in the input data.

Motion Type	Accuracy w/o Augmentation (%)	Overall Accuracy w/ Augmentation (%)
Standing Still (Device hand held)	87.38 (6.13)	90.83 (6.20)
Vehicle	85.86 (4.91)	87.02 (4.76)
Walking	81.71 (2.20)	87.23 (2.48)

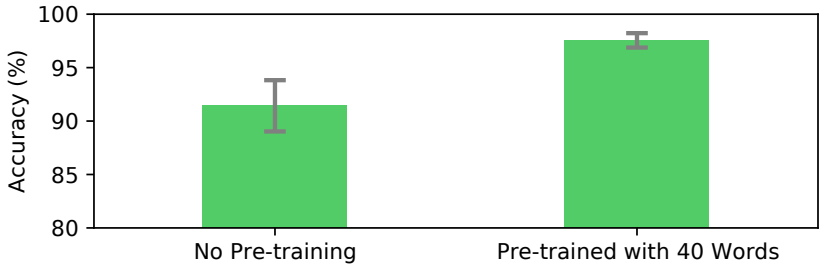


Fig. 13. Accuracy among different dataset cases. The result of training an additional 10 words for the model initialized in advance for each case.

a single hand. Specifically, we ask five participants to make sign gestures for numbers from 1-10 to see the classification accuracy of these gestures in mobile situations (also part of the multi-environment dataset). As Table 9 shows, we test for three different cases: (1) standing still with the phone in hand, (2) riding a vehicle in the passenger seat, and (3) walking while holding the phone with one hand and making sign gestures with the other. Table 9 also presents the classification accuracy for two different models. Firstly, we present results without the augmented data included in the training dataset, and secondly, with augmented data included for model training. The results show that the impact of data augmentation significantly improves classification performance. Specifically, we see improvements ranging from as low as 2% to as much as 5.5% for different cases. This performance gain is mainly because *SUGO*'s data augmentation discussed in Section 5.4 primarily focus on the trembling artifacts that can occur in daily motions, where the center of the subject moves slightly in different directions, Thus, adding such augmented data to the training data allows *SUGO*'s inference model to be robust under such input noise.

6.2.3 Potential System Scalability. To examine the potential of scaling *SUGO* with untrained sign language words, we perform an additional experiment in which we pre-train the 3DCNN model with 40 words from the 50-word dataset and use the remaining (non-overlapping) 10 words to fine tune the model and test the classification performance. The 40 words for pre-training were selected to cover the different gesture characteristics we discussed in Section 4.1. We hypothesise that since the dataset in *SUGO* includes words with different gesture characteristics, a sign language translation system that exploits such features can perform well even for new words, with a small amount of model fine-tuning. For fine-tuning, among the data collected from 20 participants, we re-train the pre-trained model with data from 15 participants, and test with the remaining five and perform four-folds cross validation. Results in Figure 13 compares this with the case where no pre-training is applied to the model. The plots indicate that pre-training the model with words of different motion characteristics has a noticeable positive impact on the classification performance.

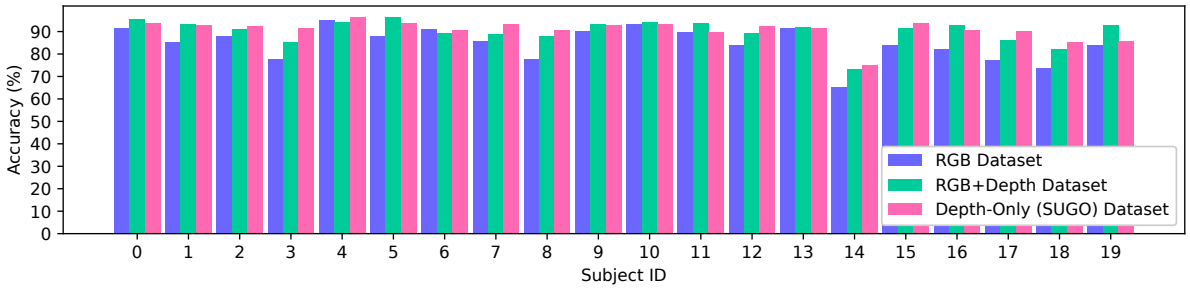


Fig. 14. Accuracy among different subjects. We present results for the cases when using the RGB dataset, the RGB+depth dataset and the Depth-only (*SUGO*) dataset (taken from Fig. 12).

Table 10. Classification accuracy with respect to different lighting conditions for the RGB, RGB+depth and Depth-only(*SUGO*) datasets (taken from Tab. 8).

Environment	Location	RGB Dataset	RGB+Depth Dataset	Depth-only (*SUGO*) Dataset
Indoor	Office	82.89	87.78	91.31
	Bright Setting	72.61	87.56	95.51
	Dark Setting	64.08	81.55	90.08
Outdoor	Daytime	67.83	80.67	92.19
	Dark Night	64.95	83.51	93.36
	Night w/ Street Lamp	74.63	82.35	90.69

6.3 Comparisons with RGB and RGB+Depth Approaches

To examine the impact of exploiting depth-only videos for sign language translation compared to previously proposed methods of using RGB or RGB+depth videos, in the following experiments, we train the *SUGO* deep learning model using the RGB videos that were simultaneously collected during our data collection phase. For RGB+depth, we add pre-processed depth images to the dataset and configure four input channels to the 3DCNN model. Overall, the classification accuracy for the RGB-trained model (via the same four-fold cross validation) was 86.04% (stdev: 3.17), suggesting that using the depth only videos not only allow for application-level gains in terms of privacy, but also shows superior performance. Surprisingly, the RGB+depth video-based model showed an average classification accuracy of 89.99% (stdev: 1.70), which is better than the RGB only case, but still does not outperform *SUGO*. Figure 14 presents the per-user accuracy achieved for the RGB-only and RGB+depth test cases. By comparing the two case, we can see that the results here also agree with our overall accuracy results. From the results, we can see that *SUGO* shows a more balanced performance over different users (stdev: 7.4 - RGB, 5.4 - RGB+Depth, 4.5 - *SUGO*).

We also present the performance of RGB only and RGB+depth inputs with different lighting conditions in Table 10. Note that for the depth data used in this experiment, we apply the human gesture emphasis module on the raw depth video inputs. Results suggest that the depth information indeed plays an important role in maintaining high accuracy under different lighting conditions. Compared with the results for *SUGO*, we can see that using the depth-only dataset shows the best performance among all cases. We note that the performance of different datasets with motion artifacts also show similar trends, in which exploiting depth information plays an important role in increasing the classification accuracy.

Table 11. Per-sign classification latency and the latency experienced by each frame in the human gesture emphasis module for different resource-limited platforms. *SUGO* is light-weight and fast enough to support real-world conversations.

Device	Per-sign Classification Latency (s)		Per-frame HGE Module Latency (s)
	`float32`	`float16`	
NVIDIA Jetson TX2	0.1632	0.1487	0.0006
NVIDIA Jetson Nano	0.3615	0.3419	0.0010
iPhone X	1.8393	-	0.0088
iPhone 11 Pro	1.2471	-	0.0065

We conjecture that such limited performance of RGB and RGB+depth is due to two (related) main reasons. First, when using RGB videos, the color and patterns present on the user's clothing can be an obstacle to the model. Similar colors and complex patterns can complicate the hand/body feature extraction process on RGB-based sign language translation systems. Second, and similarly, such a complex feature set would require a more sophisticated model architecture or a significant amount of additional training data to achieve high classification performance. Finally, we emphasize the importance of depth information in achieving high accuracy. We believe that a more tailored model for exploiting both RGB and depth information can potentially achieve better performance. Nevertheless, these comparison results suggest the importance of exploiting depth information for human gesture detection applications such as sign language translation.

6.4 Latency on Mobile Platforms

The target of *SUGO* is to operate on mobile platforms without any support from external computing platforms in the inference phase. The increased computational power of recently commercialized smartphones, especially their on-board GPUs, allow for deep learning model inference operations to take place on the mobile device. Nevertheless, compared to server-scale GPUs, their computational power is still limited; thus, making it important to consider the computational complexity of the model. We note that the model training is a one time process and takes place at the server; thus, we do not consider the training latency in this work.

We use four different platforms to observe *SUGO*'s latency performance in mobile/resource-limited computing environments: (1) NVIDIA Jetson TX2, (2) NVIDIA Jetson Nano, (3) Apple iPhone X, and (4) Apple iPhone 11 Pro. We use the PyTorch framework for the Jetson implementations and implement *SUGO* using Apple's Metal API for the iPhone platforms. Note that `float16` based implementations are not yet supported on the Apple Metal API; thus, the iPhone implementations are done based on `float32` type variables.

A summary of the per-sign classification latency and per-frame human gesture emphasis module's processing latency can be found in Table 11. Here, we can make some interesting observations. First, we can see that the human gesture emphasis module operations do not consume a noticeable amount of computation overhead compared to the actual classification process. This is important given that all input frames to *SUGO* must experience this overhead. Second, notice that the Jetson platforms' sign classification latency is much faster compared to the iPhone implementations. This is mainly because the implementations for Jetson exploit `float16` types, which reduces the computation complexity, and the PyTorch implementations are heavily optimized for the device's GPU operations. Despite being relatively higher, given that conversational pauses can be as long as 1-2 seconds [47], the latency measurements for the iPhone implementation operating on the two phones are still acceptable (especially for the iPhone 11 Pro), given typical conversation scenarios. Nevertheless, we point out that there is still room for additional optimization for the iPhone implementations. The newly announced update of the Swift framework supports `float16` implementations (which is currently only part of a beta release

Table 12. Multiple word input sequences used for our experiments.

Case #	Word Sequence
1	I enjoy drinking coffee
2	I am tired of having no money
3	Motorcycles are the best
4	I don't want to drink coffee

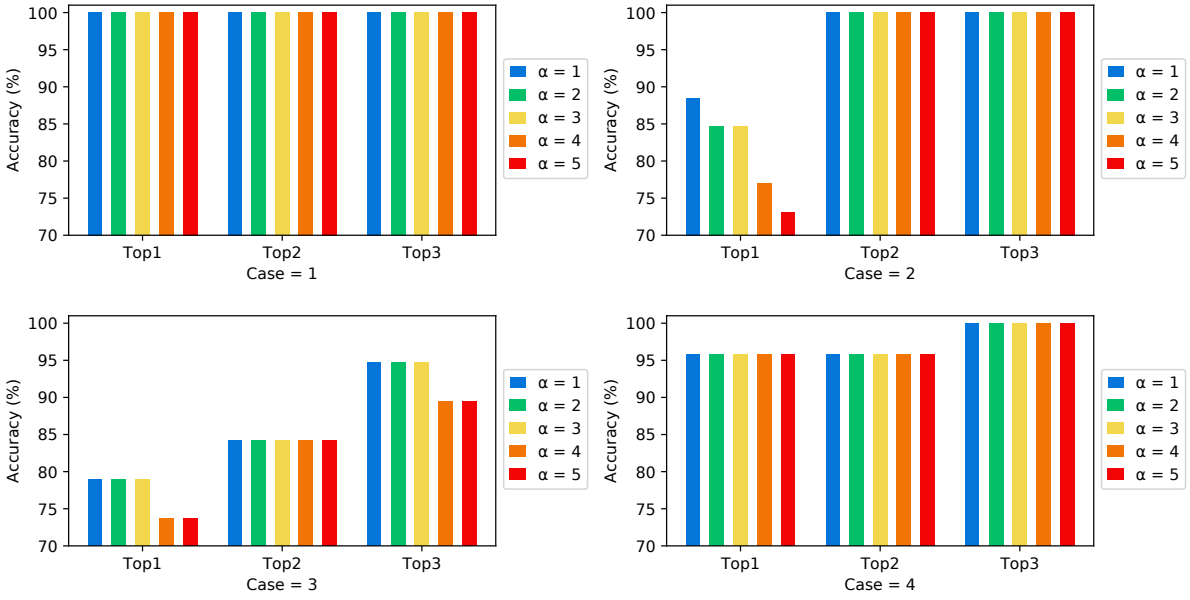


Fig. 15. Classification accuracy (top-1, 2 and 3) for different multi-word input sequences with varying α. Increasing α has minimal impact of classification accuracy; thus, can be adjusted to reduce computational overhead and latency when applying *SUGO* to input sequences of multiple words.

at the point of this writing), and re-implementing all operations to operate fully on the GPU in an optimized way can potentially further accelerate the iOS app operations. We see these issues as essential steps towards commercial-scale deployments, but do not focus on implementation optimization in this work.

6.5 Continuous Word Segmentation

Finally, we examine how continuous word-based input sequences are effectively detected in *SUGO* using the word segmentation scheme presented in Section 5.5. As mentioned, we vary α to show how frame skipping impacts word classification accuracy for continuous word-based input as we perform sliding window-based word segmentation. The data for this experiment was collected from five participants for four different word sequences. Specifically, the sequences were (1) "I enjoy drinking coffee", (2) "I am tired of having no money", (3) "Motorcycles are the best", and (4) "I don't want to drink coffee" as presented in Table 12. Note that these word sequences are a combination of the words in our dataset, and are designed so that the sequences contain compound words, complex signs, and signs that have motion similarity with other signs. In Figure 15 we present the top-1, 2 and 3

accuracy for the four word sequences (cases 1-4) with varying α. Note that α determines the number of frames to skip when operating with video inputs embedding multiple words. Results suggest that the overall accuracy of matching the proper word improves with a lower α, which is expected given that a low α will check more possible cases for word detection. What is surprising is that for some cases, increasing α has nearly no impact on classification performance. Even for cases 2 and 3, where the top-1 accuracy degrades with increasing α, the top-3 performance is kept high. This suggests that with an effective natural language processing (NLP) module to choose the most grammatically and contextually probable word among the three, we can dramatically reduce the computation cost (near linearly to α) and still achieve high accuracy in word/sentence estimation. While detailed studies on such NLP schemes are out of the scope of this work, *SUGO* shows promising results in being applied to sentence-level sign language translation.

7 LIMITATIONS AND FUTURE DIRECTIONS

While we demonstrate the strong potential of *SUGO*, we need further investigation regarding some issues due to the complexity of the application domain. In this section, we list a few limitations of the current system and describe future research directions related to such issues.

- **Performance with Heterogeneous Depth Cameras**: *SUGO* focuses on analyzing depth videos for sign language translations, but one limitation of this work is that we have focused on a single type of depth camera, IR-based depth sensors. We do so given that the iPhone's TrueDepth sensor is easy to access and exploiting this data can be considered as a suitable candidate to show the feasibility of using depth-only video data for sign language translation. However, other mobile platforms adopt different sensors such as stereo cameras to collect depth data. While both represent depth information, we need further studies on how a model trained with one sensor type can perform with input data collected at heterogeneous sensing components. This will be an interesting direction of research to observe how inference models for depth videos can be generalized to support heterogeneous depth camera inputs.
- **Sentence Level Analysis**: As discussed in Section 5.5, it is crucial to design a system with the capability to understand and process sentence-level input sequences. In this work, we perform first-stage experiments to validate that word sequences can be detected with depth videos being used as input, but is limited in the fact that we do not focus on how an entire sentence can be structured with grammatical accuracy. It is meaningful to note that the grammar of various sign languages and the grammar of the corresponding verbal language do not necessarily match. Thus, to perfectly support sign language translation, there is a need to understand the input sentence sequence at the word-level, then present a translation with grammar-aware sentences. Furthermore, as our results in Figure 15 suggest, with a word-level translation system offering the top-K candidates, a natural language processing module can be designed to select to most probable match. As such an example, we believe that many interesting natural language processing-related research issues exist, while they are out of the scope of this work. Such future efforts can improve the performance and usability of sign language translation systems.
- **Validating Word Diversity**: Given that sign language words are continuously created and there are near 5,000 commonly used words in most sign languages, further research is required to validate the scalability of *SUGO*. While we carefully chose 50 words with diverse motion characteristics to assure that the system can be scalable to more words, validating the performance with additional words is essential. Thus, we see the diversification of vocabulary as the most important step prior to real-world deployments. For this, we are collaborating with local associations for the deaf in creating a large scale dataset with commonly used KSL vocabulary.

8 CONCLUSION

We live in a society where different types of communication methods coexist: verbal and visual. As a ubiquitous computing research community, we have continuously proposing enhanced systems for bridging the gap between the two communication groups as we believe that such efforts are essential in realizing a true language barrier-free society. In this work, building up on many previous efforts, we proposed *SUGO*. Unlike previous work, *SUGO* targets to use only depth videos as its input from the user, and the smartphone as the only computing platform for inference (after on-server training) to achieve effective sign language translation. Such attributes of *SUGO* offer a less privacy-invasive system and frees‘ sign language users from carrying external sensing components just for the sake of pursuing everyday conversations. The dataset used in this study, collected from 20 subjects for 50 Korean Sign Language words, is designed to validate the effectiveness of *SUGO* with respect to different motion characteristics of various sign language gestures. We verify the performance of *SUGO* in various real-world environments and use cases to show that *SUGO* is robust enough to overcome the diverse input noise that practical scenarios introduce and produce accurate word classification results. Furthermore, we show that *SUGO* is light-weight and the inference latency is suitable for supporting real-world conversations. With a larger dataset of sign language gestures, we believe that the findings in this work can act a catalyst for supporting privacy-aware smartphone-based sign language translation that can socially connect the people using different communicating methods.

ACKNOWLEDGMENTS

The authors would like to thank all the study participants and the members of the Gyeonggi-Do Sign Language Education Institute. The authors would especially like to express gratitude to Mr. Gwangchul Park for his help in offering detailed application needs on a sign language user’s perspective, and for being a wonderful sign language teacher to the research team. This work was financially supported by the National Research Foundation of Korea (NRF) Grant funded by the Ministry of Science and ICT (No. 2021R1A2C4002380), and by the Yonsei University Research Fund (Grant No. 2020-22-0513).

REFERENCES

[1] 2020. *Deafness and hearing loss.* https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss
[2] 2020. *Harness AI at the Edge with the Jetson TX2.* https://developer.nvidia.com/embedded/jetson-tx2-developer-kit
[3] 2020. *Jetson Nano.* https://developer.nvidia.com/embedded/jetson-nano
[4] J. Ahn, H. Nguyen Loc, R. Krishna Balan, Y. Lee, and J. Ko. 2018. Finding Small-Bowel Lesions: Challenges in Endoscopy-Image-Based Learning Systems. *Computer* 51, 5 (2018), 68–76.
[5] Saleh Aly and Walaa Aly. 2020. DeepArSLR: A novel signer-independent deep learning framework for isolated arabic sign language gestures recognition. *IEEE Access* 8 (2020), 83199–83212.
[6] Walaa Aly, Saleh Aly, and Sultan Almotairi. 2019. User-independent American sign language alphabet recognition based on depth image and PCANet features. *IEEE Access* 7 (2019), 123138–123150.
[7] Ursula Bellugi and Susan Fischer. 1972. A comparison of sign language and spoken language. *Cognition* 1, 2 (1972), 173 – 200. https://doi.org/10.1016/0010-0277(72)90018-2
[8] Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, et al. 2019. Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective. In *The 21st International ACM SIGACCESS Conference on Computers and Accessibility*. 16–31.
[9] Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6299–6308.
[10] Xiujuan Chai, Guang Li, Yushun Lin, Zhihao Xu, Yili Tang, Xilin Chen, and Ming Zhou. 2013. Sign language recognition and translation with kinect. In *IEEE Conf. on AFGR*, Vol. 655. 4.
[11] Cao Dong, Ming C Leu, and Zhaozheng Yin. 2015. American sign language alphabet recognition using microsoft kinect. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 44–52.
[12] Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, et al. 2020. DepthLab: Real-Time 3D Interaction With Depth Maps for Mobile Augmented Reality. In *Proceedings of the 33rd*

Annual ACM Symposium on User Interface Software and Technology. 829–843.
[13] Deniz Ekiz, Gamze Ege Kaya, Serkan Buğur, Sıla Güler, Buse Buz, Bilgin Kosucu, and Bert Arnrich. 2017. Sign sentence recognition with smart watches. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 1–4.
[14] Biyi Fang, Jillian Co, and Mi Zhang. 2017. DeepASL: Enabling ubiquitous and non-intrusive word and sentence-level sign language translation. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*. 1–13.
[15] Susan Goldin-Meadow and Diane Brentari. 2017. Gesture, sign, and language: The coming of age of sign language and gesture studies. *Behavioral and Brain Sciences* 40 (2017), e46. https://doi.org/10.1017/S0140525X15001247
[16] Dan Guo, Wengang Zhou, Houqiang Li, and Meng Wang. 2017. Online early-late fusion based on adaptive HMM for sign language recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14, 1 (2017), 1–18.
[17] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. 2017. Learning spatio-temporal features with 3D residual networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 3154–3160.
[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.
[19] Jiahui Hou, Xiang-Yang Li, Peide Zhu, Zefan Wang, Yu Wang, Jianwei Qian, and Panlong Yang. 2019. SignSpeaker: A Real-Time, High-Precision SmartWatch-Based Sign Language Translator. In *The 25th Annual International Conference on Mobile Computing and Networking (MobiCom '19)*. Association for Computing Machinery, New York, NY, USA, Article 24, 15 pages. https://doi.org/10.1145/3300061.3300117
[20] Jie Huang, Wengang Zhou, Houqiang Li, and Weiping Li. 2015. Sign language recognition using 3d convolutional neural networks. In *2015 IEEE international conference on multimedia and expo (ICME)*. IEEE, 1–6.
[21] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. 2018. Video-based sign language recognition without temporal segmentation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.
[22] Loc N Huynh, Youngki Lee, and Rajesh Krishna Balan. 2017. Deepmon: Mobile gpu-based deep learning framework for continuous vision applications. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. 82–95.
[23] Hamid Reza Vaezi Joze and Oscar Koller. 2018. Ms-asl: A large-scale data set and benchmark for understanding american sign language. *arXiv preprint arXiv:1812.01053* (2018).
[24] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. 2017. The Kinetics Human Action Video Dataset. *CoRR* abs/1705.06950 (2017). arXiv:1705.06950 http://arxiv.org/abs/1705.06950
[25] Wajahat Kazmi, Sergi Foix, Guillem Alenyà, and Hans Jørgen Andersen. 2014. Indoor and outdoor depth imaging of leaves with time-of-flight and stereo vision sensors: Analysis and comparison. *ISPRS journal of photogrammetry and remote sensing* 88 (2014), 128–146.
[26] Hyun Myung Kim, Min Seok Kim, Gil Ju Lee, Hyuk Jae Jang, and Young Min Song. 2020. Miniaturized 3D depth sensing-based smartphone light field camera. *Sensors* 20, 7 (2020), 2129.
[27] Oscar Koller, Cihan Camgoz, Hermann Ney, and Richard Bowden. 2019. Weakly supervised learning with multi-stream CNN-LSTM-HMMs to discover sequential parallelism in sign language videos. *IEEE transactions on pattern analysis and machine intelligence* (2019).
[28] Oscar Koller, Jens Forster, and Hermann Ney. 2015. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding* 141 (Dec. 2015), 108–125.
[29] Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. 2018. Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs. *International Journal of Computer Vision* 126, 12 (2018), 1311–1325.
[30] Okan Köpüklü, Neslihan Kose, Ahmet Gunduz, and Gerhard Rigoll. 2019. Resource efficient 3d convolutional neural networks. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 1910–1919.
[31] Dongxu Li, Cristian Rodriguez, Xin Yu, and Hongdong Li. 2020. Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1459–1469.
[32] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. 2016. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710* (2016).
[33] Yanqiu Liao, Pengwen Xiong, Weidong Min, Weiqiong Min, and Jiahao Lu. 2019. Dynamic sign language recognition based on video sequence with BLSTM-3D residual networks. *IEEE Access* 7 (2019), 38044–38054.
[34] Yongsen Ma, Gang Zhou, Shuangquan Wang, Hongyang Zhao, and Woosub Jung. 2018. Signfi: Sign language recognition using wifi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 1–21.
[35] Ilya Makarov, Nikolay Veldyaykin, Maxim Chertkov, and Aleksei Pokoev. 2019. American and russian sign language dactyl recognition. In *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. 204–210.
[36] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: enabling innovation in campus networks. *ACM SIGCOMM computer communication review* 38, 2 (2008), 69–74.
[37] Mohamed A Mohandes. 2013. Recognition of two-handed Arabic signs using the CyberGlove. *Arabian Journal for Science and Engineering* 38, 3 (2013), 669–677.

[38] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. 2016. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4207–4215.
[39] Chaithanya Kumar Mummadi, Frederic Philips Peter Leo, Keshav Deep Verma, Shivaji Kasireddy, Philipp Marcel Scholl, and Kristof Van Laerhoven. 2017. Real-time embedded recognition of sign language alphabet fingerspelling in an imu-based glove. In *Proceedings of the 4th international Workshop on Sensor-based Activity Recognition and Interaction*. 1–6.
[40] myoarmband [n. d.]. . https://support.getmyo.com/hc/en-us
[41] Deepali Naglot and Milind Kulkarni. 2016. Real time sign language recognition using the leap motion controller. In *2016 International Conference on Inventive Computation Technologies (ICICT)*, Vol. 3. IEEE, 1–5.
[42] JaeYeon Park, Hyeon Cho, Rajesh Balan, and JeongGil Ko. 2020. HeartQuake: Accurate Low-Cost Non-Invasive ECG Monitoring Using Bed-Mounted Geophones. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 3 (2020).
[43] Sunitha Ravi, Maloji Suman, PVV Kishore, Kiran Kumar, Anil Kumar, et al. 2019. Multi modal spatio temporal co-trained CNNs with single modal testing on RGB–D based sign language gesture recognition. *Journal of Computer Languages* 52 (2019), 88–102.
[44] Panneer Selvam Santhalingam, Al Amin Hosain, Ding Zhang, Parth Pathak, Huzefa Rangwala, and Raja Kushalnagar. 2020. mmASL: Environment-Independent ASL Gesture Recognition Using 60 GHz Millimeter-wave Signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–30.
[45] Jiacheng Shang and Jie Wu. 2017. A robust sign language recognition system with sparsely labeled instances using Wi-Fi signals. In *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. IEEE, 99–107.
[46] Bowen Shi, Aurora Martinez Del Rio, Jonathan Keane, Jonathan Michaux, Diane Brentari, Greg Shakhnarovich, and Karen Livescu. 2018. American sign language fingerspelling recognition in the wild. In *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 145–152.
[47] Harold P. Stern, Samy A. Mahmoud, and Kin-Kwok Wong. 1996. A comprehensive model for voice activity in conversational speech-development and application to performance analysis of new-generation wireless communication systems. *Wireless Networks* 2 (1996), 359—-367. Issue 4.
[48] Chao Sun, Tianzhu Zhang, and Changsheng Xu. 2015. Latent support vector machine modeling for sign language recognition with Kinect. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 2 (2015), 1–20.
[49] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. 2018. A survey on deep transfer learning. In *International conference on artificial neural networks*. Springer, 270–279.
[50] Ultraleap. [n. d.]. Leap Motion Contorller. Available at https://www.ultraleap.com/product/leap-motion-controller/.
[51] Hanjie Wang, Xiujuan Chai, Xiaopeng Hong, Guoying Zhao, and Xilin Chen. 2016. Isolated sign language recognition with grassmann covariance matrices. *ACM Transactions on Accessible Computing (TACCESS)* 8, 4 (2016), 1–21.
[52] Seongok Won. 2019. A Study on the Korean Sign Language(KSL) Grammer.
[53] Jian Wu, Lu Sun, and Roozbeh Jafari. 2016. A wearable system for recognizing American sign language in real-time using IMU and surface EMG sensors. *IEEE journal of biomedical and health informatics* 20, 5 (2016), 1281–1290.
[54] Quan Yang. 2010. Chinese sign language recognition based on video sequence appearance modeling. In *2010 5th IEEE Conference on Industrial Electronics and Applications*. IEEE, 1537–1542.
[55] Pujianto Yugopuspito, I Made Murwantara, and Jessica Sean. 2018. Mobile sign language recognition for bahasa indonesia using convolutional neural network. In *Proceedings of the 16th International Conference on Advances in Mobile Computing and Multimedia*. 84–91.
[56] Lei Zhang, Yixiang Zhang, and Xiaolong Zheng. 2020. WiSign: Ubiquitous American Sign Language Recognition Using Commercial Wi-Fi Devices. *ACM Transactions on Intelligent Systems and Technology (TIST)* 11, 3 (2020), 1–24.
[57] Qian Zhang, Dong Wang, Run Zhao, and Yinggang Yu. 2019. MyoSign: enabling end-to-end sign language recognition with wearables. In *Proceedings of the 24th International Conference on Intelligent User Interfaces*. 650–660.
[58] Zhengyou Zhang. 2012. Microsoft Kinect Sensor and Its Effect. *IEEE MultiMedia* 19 (April 2012), 4–12. https://www.microsoft.com/en-us/research/publication/microsoft-kinect-sensor-and-its-effect/
[59] Henry Zhong, Salil S Kanhere, and Chun Tung Chou. 2017. VeinDeep: smartphone unlock using vein patterns. In *2017 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 2–10.
[60] Zijie Zhu, Xuewei Wang, Aakaash Kapoor, Zhichao Zhang, Tingrui Pan, and Zhou Yu. 2018. EIS: A Wearable Device for Epidermal American Sign Language Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 4 (2018), 1–22.