Concept Embedding through Canonical Forms: A Case Study on Zero-Shot ASL Recognition

A. Kamzin*, V. N. S. A. Amperayani †,P. Sukhapalli‡, A. Banerjee§ and S. K.S. Gupta¶
IMPACT Lab CIDSE, Arizona State University, Tempe, AZ, USA

Email: *akamzin, †vamperay, ‡psukhapa, §abanerj3, ¶sandee.gupta@asu.edu

***Abstract*—In the recognition problem, a canonical form that expresses the spatio-temporal relation of concepts for a given class can potentially increase accuracy. Concepts are defined as attributes that can be recognized using a soft matching paradigm. We consider the specific case study of American Sign Language (ASL) to show that canonical forms of classes can be used to recognize unseen gestures. There are several advantages of a canonical form of gestures including translation between gestures, gesture-based searching, and automated transcription of gestures into any spoken language. We applied our technique to two independently collected datasets: a) IMPACT Lab dataset: 23 ASL gestures each executed three times from 130 first time ASL learners as training data and b) ASLTEXT dataset: 190 gestures each executed six times on an average. Our technique was able to recognize 19 arbitrarily chosen previously unseen gestures in the IMPACT dataset from seven individuals who are not a part of 130 and 34 unseen gestures from the ASLTEXT dataset without any retraining. Our normalized accuracy on the ASLTEXT dataset is 66% which is 13.6 % higher than the state-of-art technique.**

I. Introduction

Learning concepts is a high level cognitive task that is at the frontier of Artificial Intelligence (AI) research. In a recognition problem, concepts are attributes of examples, which exhibit the following properties: a) *soft matching*, where two concepts c_1 and c_2, are considered to be equal if $dist(c_1, c_2) \leq \epsilon$, where $\epsilon > 0$ governs the degree of match, b) *structure*, an example can be expressed as a combination of the concepts following a temporal or spatial order, c) *uniqueness*, each example has a unique unambiguous structural representation in terms of concepts, and d) *coverage*, every example in the given recognition problem has a structural representation in terms of the concepts.

Enabling a machine to recognize concepts can potentially increase the number of examples that can be correctly identified by it. As shown in Figure 1, domain experts provide classes which are divided into two groups: a) seen classes, where examples are available, and b) unseen classes, where examples are unavailable. Every class can be defined using a spatio-temporal ordering of a set of concepts, which is provided by the expert. This is the *canonical form* for a class.

Canonical form has two properties: a) it is machine readable encoding, and b) each class has a unique canonical form.

Examples from seen classes can be used to learn models that can recognize each concept. In the testing phase, given the first example of a previously unseen class, the canonical form can be utilized to segment. Each segment can then be compared with concept models. The comparison output and canonical form of unseen class can be utilized for recognition.

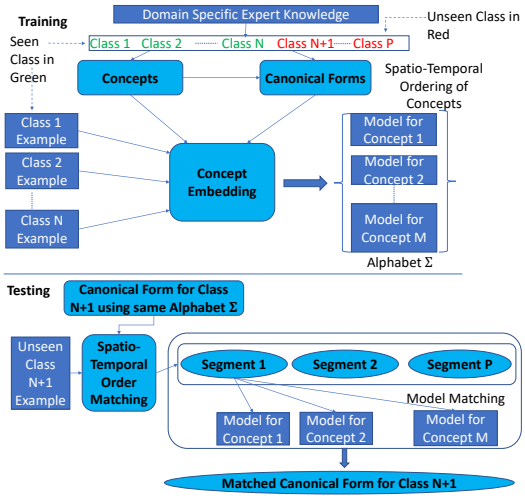

Fig. 1. Concept Embedding for zero-shot recognition

In the example of the American Sign Language (ASL) recognition problem, all of the nearly 10,000 gestures for English words are composed using a set of over 80 handshapes, six locations and around 20 unique movements [1]. Each handshape, movement and location has a semantic relation with the English word and can be considered as concepts. Each gesture can be expressed using a unique ordering of start handshape start location, a movement type, end handshape and end location, which is the canonical form for that gesture. If a machine learns these unique concepts, then by combining them following a language, the machine can potentially recognize gestures that it has never seen before. This concept of recognizing previously unseen classes without access to training data is known as zero-shot learning [2]–[6]. It can be used for many purposes such as ASL learning [7], training personnel in various domains such as construction or military [8], or validating the quality of unsupervised physiotherapeutic exercises [9].

The backbone of state-of-art zero-shot recognition is attribute-based learning [2]. Here the raw data of a labeled training example is projected in a given attribute space and the resulting projection is qualified with a semantic meaning. An unseen test case with a semantic definition is then expressed as a combination of seen projections. The semantic meanings associated with each projection in the combination are then utilized to match with the given definition to recognize the unseen test case [2]. The attributes that are learned from a seen training example are parameters such as weights of CNN

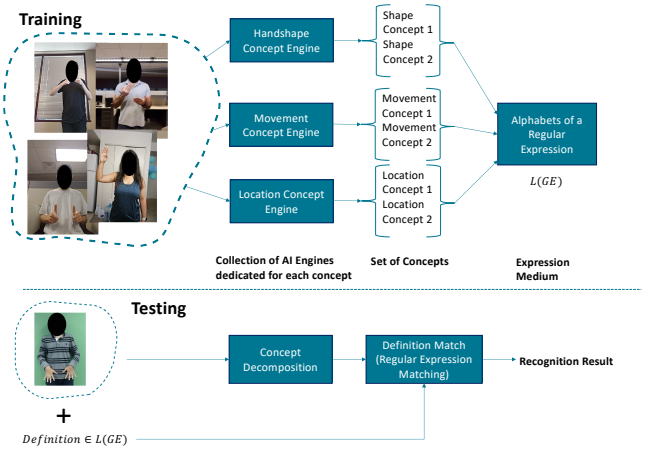

Fig. 2. Application of concept embedding on ASL

layers or activation levels of neurons [2]. These attributes may not be directly semantically relevant. They are rather outputs of a randomized learning algorithm, which are manually assigned semantic meaning by an expert observer. This results in drawbacks:

A) The *projection domain shift* problem, where due to differences in data distributions between two domains, examples with same semantic meanings may not have the same projections. Existing solutions to this problem, typically increase the complexity of zero-shot learning with only incremental improvement in accuracy [10]. For example, in the case of ASL recognition using video, data may be obtained from different environments resulting in significant variation in camera angles, background, and resolution. This can result in differences in projections of visually similar gesture examples.

B) Often exact matching of semantic mappings between examples of different classes is not required for correct recognition. For example, in ASL the exact pixel level location of a gesture is not important, rather the general proximity of the hand to a given part of the body is relevant. Current zero-shot techniques cannot benefit from such "soft" matching notions.

C) Semantic matching can be multi-dimensional with both spatial and temporal structures. For example, each gesture in ASL has a handshape at a certain location, (spatial information) transitioning to another handshape in the same or different location, resulting in a temporal evolution of spatial information. The state-of-art way to handle such spatio-temporal semantics will be to map to properties of 3D CNNs or RNNs, which can result in complex features increasingly making matching difficult.

D) Semantic matching between two classes may not spatio-temporally align. In ASL, gestures can have same handshapes but in different spatial locations or at varying times. As such semantic matching for ASL will require tackling both spatial and temporal alignments simultaneously which is far more difficult to achieve in zero-shot sense.

In this paper, we show that our canonical form can be used as the intermediate modular representation that is required for zero-shot learning. The fundamental difference is in the definition of a concept that enables soft matching and the usage of canonical forms that convert an example into concepts arranged in spatio-temporal order. We apply our concept embedding strategy for zero-shot learning of ASL gestures. In our experiments, we utilize two datasets: a) IMPACT Lab dataset, which consists of 23 ASL gestures from 130 users and identify 19 unseen ASL gestures from seven users not part of the initial 130, and b) ASLTEXT dataset, from which we select 190 unseen gestures collected in an independent experimental environment [4]. In the IMPACT dataset gestures were performed with three repetitions each, resulting in a database size of 8,970 gesture executions. Our zero-shot mechanism can recognize 43 unseen gestures from both IMPACT (9) and ASLTEXT (34) datasets using training data from 23 gestures only obtained from the IMPACT dataset. The normalized accuracy as reported by [4] is around 66% for our mechanism which is 13.6% higher than the state-of-art [4].

II. System Model and Problem Statement

In this section, we discuss the ASL recognition system model and the proposed canonical embedding of concepts.

A. Video based ASL Recognition

Video based gesture recognition has been a topic of recent interest among researchers in the mobile computing domain. The basic idea (Figure 2) is, given a set of videos of users performing a set of gestures in a given language such as ASL, along with correct labels, the application recognizes executions of the same gestures by other users. In examples where the total number of possible gestures is limited, this approach is well established. However, in the case of language learning such as ASL, the number of possible gestures can be more than 10,000. Moreover, ASL is an organically generated language which is also constantly evolving with new gestures. In such cases, the requirement of available training examples for a given gesture is problematic.

To overcome such requirements, we consider the system model described in Figure 2. We assume that any user only performs few examples of a limited subset of gestures and labels them correctly. A gesture for which training examples is available is denoted by S_G^i. In addition to S_G^i, a user can also perform gestures which are previously unseen. We denote these gestures by G_i. Given such definitions, we assume that $\{S_G^1 \dots S_G^n\} \bigcap \{G_1 \dots G_m\} = \phi$.

B. Canonical Form of ASL Gestures

The first step to defining the problem of zero-shot gesture understanding is to characterize a gesture. Following our experiences from ASL recognition, we can express a gesture in ASL using a set of regular expressions.

We consider the set $\Sigma = \Sigma_H \bigcup \Sigma_L \bigcup \Sigma_M$ to be the alphabet of ASL. The alphabet of ASL consists of three subsets, a) Σ_H is a set of handshapes, (ASL has a finite set of handshapes), b) Σ_L is a set of locations, (We divide the head and torso region of the human body into six buckets), and c) Σ_M is a set of movements of the arm. The alphabets can be individually performed by the right or the left hand. We define a set of regular expressions, called Gesture Expression (GE) as:

$$
\begin{aligned}
Hand &\rightarrow \Sigma_H \\
Mov &\rightarrow \Sigma_M \\
Loc &\rightarrow \Sigma_L \\
GE &\rightarrow GE_{Left} GE_{Right} \\
GE_X &\rightarrow Hand | \epsilon, \text{ where } X \in \{Right, Left\} \\
GE_X &\rightarrow Hand\ Loc \\
GE &\rightarrow Hand\ Loc\ Mov\ Hand\ Loc
\end{aligned} \tag{1}
$$

Here ϵ denotes null gesture, i.e. a particular hand is not used. We define a valid gesture using Definition 1

Definition 1: A gesture g is a valid gesture if and only if $g \in L(GE)$, where $L(GE)$ denotes the language of the regular expression GE in Equation 1.

Justification of Equation 1: A deeper analysis of ASL gestures also reveal that ASL only has a limited set of hand shapes nearly 90 with which all 10,000 ASL gestures can be executed [11]. For analysis based on location, only the general position of the palm with respect to other parts of the body is semantically relevant. The palm's exact location in terms of absolute pixel numbers in the video frame is unnecessary [12]. Only the start and end handshapes are required for expressing correct semantics in ASL, handshapes when moving from start handshape to end are irrelevant. Finally, ASL gestures only a specific set of movements of both arms, which is often limited by the human motor capabilities [13]. Given such knowledge about ASL, the Definition 1 can encompass a significant percentage if not all of the ASL gesture dictionary.

C. Problem Definition

Definition 2 gives our problem statement.

Definition 2: Given:

- Training videos of each gesture in the set $\{S_G^1 \dots S_G^n\}$
- Definitions of each gesture in the set $\{G_1 \dots G_m\}$ in terms of regular expressions in Equation 1.

Recognize examples from $\{G_1 \dots G_m\}$
Such that:

- C1: $S_G^i \in L(GE)\ \forall i$ and $G_i \in L(GE)\ \forall i$
- C2: $\{S_G^1 \dots S_G^n\} \bigcap \{G_1 \dots G_m\} = \phi$
- C3: $x \in \Sigma\ iff\ \exists g \in \{S_G^1 \dots S_G^n\}$ such that $g = C_1 \dots C_k\ x\ C_{k+1} \dots$, where $C_k \in \Sigma\ \forall k$.

For an ASL gesture, two hands are used simultaneously to perform two gestures following Definition 1. We require identification of gestures performed by both the hands.

III. Related Work

Zero-shot learning for gestures has been studied and applied to some extent mostly in the field of human robot-interaction(HRI) [5], [14], [15] and has been promising. The need for some form of semantic information or labeling of gestures is an issue that hinders zero-shot learning. For HRI, it is envisioned that a human interacting with a robot might want to use a novel and yet unfamiliar gesture to indicate a command. The robot has to first determine that a new gesture is out-of-vocabulary then it has to leverage some form of semantic information in the gesture to understand its meaning. There is a lot of uncertainty in this particular application, because AI agents, unlike humans, don't learn by fundamentally learning underlying concepts, thus transfer learning at a concept level is difficult. This factor is exhibited in recent research by Bilge et al [4], which uses a 3D CNN to learn characteristics of the whole gesture execution and then recognize new gestures in a zero-shot sense. However, they could only achieve an accuracy of 15%. The main novelty in this paper is we decompose gestures into its canonical form which has some correlation with unique concepts in the language. Our technique enables the engine to learn concepts rather than examples.

IV. ASL data collection and Preprocessing

IMPACT Lab dataset:

We collected 23 ASL gesture videos with three repetitions each in real-world settings using a mobile application Learn2Sign (L2S) from 130 learners. No restrictions are laid for light conditions, distance to the camera, recording pose (either sitting or standing). The 23 gestures are used to generate a limited set of ASL alphabet and then 19 additional test gestures are chosen from two new users who are not part of the 130 learners.

Out of these additional 19, three gestures have alphabets that are not part of the alphabet generated by the initial group of 23. The other 16 can be composed of the alphabet generated by the initial group of 23 using Equation 1 gesture expression. Figure 3 shows the handshape alphabet generated by the 23 gestures and also shows the handshapes of the 19 test gestures. Figure 4 shows the movement alphabet for the 23 training gestures and 19 test gestures. We divide location into six buckets (Section V), as any gesture can be classified in these six location buckets numbered with 0 to 5 bucket numbers forming the location alphabets.

ASLTEXT dataset [4]: It is a subset of ASL Lexicon Video Dataset [16] which is collected at Boston University from ASL native signers. The ASLTEXT consisting of 250 unique gestures. There are 1598 videos out of which we utilize 1200 videos of 190 gestures not in the IMPACT dataset. Our aim in this paper is to utilize all 190 unique gestures as a test set to validate our zero-shot capabilities. We do not use any part of the ASLTEXT dataset for training purposes.

V. Approach

Stocke et al. have identified location, handshape and movement as major parts that gives meaning to any sign [12]. In this paper, we refer to them as **tokens**.

A. Token Recognition

The first step in our approach is to recognize tokens from a gesture execution.

1) Location Recognition: We have to consider two locations of the palm: a) start location and b) end location. To achieve that we first consider the PoseNet model for Real-time human post estimation. Given a video frame, this model identifies joint positions such as wrist, nose, eyes, elbow, hips, shoulders in a 2D space. For location estimation of the palm, the wrist joint is the most relevant information. We first use PoseNet to obtain the joint location (key points) frame by frame from a video of ASL gesture execution. Since we only want to understand the concept behind a given location where the gesture is executed, we do not need the exact pixel location.

Signs	About, Day, Deaf, AGREE, HEARING, HURT, TAIL	After, Goodnight (S), CHEER	And, Goout, ADD (E)	Can, ADOPT (E), ANTI, TASK, ALLGONE (E)	Cat, ADVANTAGE , Find, BAR, Decide, Find	Cop, APPETITE	Cost	Father, Hello	Gold (S)	Gold (E), PHONE	Goodnight (E)	Help, Hospital (E), ALIVE	Here	Hospital (S)	If	Large	Sorry, ALIVE(sometimes)	Tiger, ALLGONE (S), ADOPT (S), ADD(S)	ADULT, ADVANCE	ADVENT	AGAPE
Shape No.	H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14	H15	H16	H17	H18	H19	H20	H21
Shape																					

Fig. 3. Handshape alphabet generated from 23 training gestures from 130 users. Additional 19 unseen test gestures (all CAPS) are also depicted in the table.

Signs	ADD, ADVANTAGE , AGAPE, AGREE, APPETITE, Can, Cost, Decide, HURT, Goodnight	ADOPT, ADULT, ADVANCE, ADVENT, ALIVE, Help	ALLGONE, And, BAR, Large	Cat, Cop, Deaf, Father, HEARING, If, PHONE, TAIL, TASK, Tiger	Day	About	After	ANTI	CHEER	Find	Goout	Gold	Hello	Here	Hospital	Sorry
Movement No	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14	M15
Movement Description	Down	Up	Sideways	Stationary Tap	Across	Circling	Down and Up	Up and shake	Up and Down Twice	Wrist rotation up	Lateral Sideways	Shaking and away from body	Move right arm from head away from the body in a diagonal fashion	Shake horizontally	Make a complex right arm movement across the body	More right arm in a circular motion.

Fig. 4. Movement alphabet generated from 23 training gestures from 130 users. Additional 19 unseen test gestures (all CAPS) are also depicted in the table.

We need to capture the location in a more granular manner that corresponds to proximity and relative position of the wrist or palm with respect to other significant body parts. Moreover, we want to compare gesture executions by different individuals who have taken a video of gesture recognition in a constraint-free environment. Hence, we will have to deal with several unwanted artifacts such as unequal frame sizes, different body sizes, different starting points. Furthermore, ASL as a language does not have binding constraints on gesture execution. Hence, two individuals performing the same gesture may vary in their location, however, they have to represent the same general location with respect to the other parts of the body for semantic equivalence. To overcome such issues, we consider location bucketing with individualized bucket parameters. We consider the shoulders of a person to be a fixed reference. We then draw two axes: x-axis is the line that connects the two shoulder joints and the y-axis is perpendicular to the x-axis. The first bucket has a width equal to the shoulder width and height that extends to the top of the frame. We then use five more buckets: a) top left that extends from left shoulder (viewer perspective) to the left edge of the frame along the x-axis and from shoulder to top edge of the frame along the y-axis, b) top right that extends from the right shoulder to the right edge of the frame along x-axis and shoulder to top edge of a frame along the y-axis, c) bottom that extends between the two shoulders along the x-axis and from shoulder to the bottom edge of the frame along the y-axis, d) bottom left and bottom right are same as the top left and top right except they extend from shoulder to the bottom frame edge. To compensate for exaggerated movements or differences in palm sizes, we have considered the forearm length as the distance between the wrist point and the elbow point and extended the wrist point by 1/3 of the arm length to approximately project the fingertip. We then track the fingertip of the hand traversing through the location buckets across all frames. The features of the start and end location of the hand are captured through counting the number of times the projected in the given bucket throughout the first half and second half of the video respectively. The overall result of this step is a 12D vector where the first 6d values correspond to the start location and the next 6d values correspond to the end location and both are normalized separately.

2) Movement Recognition: In the 23 seen gestures considered in this paper, the gestures have 16 unique movement patterns. These gestures are numbered 1 through 16. Our aim is, given execution of a gesture, to identify the top three numbered movement patterns.

For extracting the movement attributes from the collected 2D videos for our experiment, the PoseNet model for Real-Time Human Pose Estimation is used. The TensorFlow is based ML model gives us the coordinates of some parts of the human pose for every frame of the performer's video. Based on the x-axis and y-axis coordinates of some parts, a decent identification of hand gestures is possible.

For our experiment, the right wrist and left wrist hand movements are tracked since they are principal in performing the gestures. For finding similarity between the right-hand movement of two videos of different gestures: `IF` and `DEAF` in our case, the coordinates of nose, left-hip and right-hip are taken as the standard reference points since they persist as stationary points throughout the video. Based on the maximum accuracy score for the individual parts of the model, the corresponding coordinates are considered as reference.

The midpoint of the left and right hip is found and the distance between nose and this midpoint is taken as the torso height whereas the distance between the left and right hip is considered as the torso width for normalization. This kind of geometric scheme is made to balance orientation and scaling across any two videos. For both the videos, the new x and y movement coordinates are calculated as:

$$x_{new}^{wrist} = \frac{x_{old}^{wrist} - x_{nose}}{hip - width}, \; y_{new}^{wrist} = \frac{y_{old}^{wrist} - y_{nose}}{D_H}, \qquad (2)$$

where D_H is the distance between nose and the midpoint between left and right side of the hips. Thus, the new coordinates obtained are collected as movement attributes from both the videos and compared with a suitable time-series metric like Dynamic Time warping in order to synchronize the different onset of movement between the videos. The final 2d-DTW score based on Euclidean distance obtained is used as the metric for similarity. The lower the score, the higher is the similarity between any two videos. For each gesture in our database, we store the top three movement type matches.

3) Handshape Recognition: ASL is a visual language and hand shape is an important part of identifying any sign. In

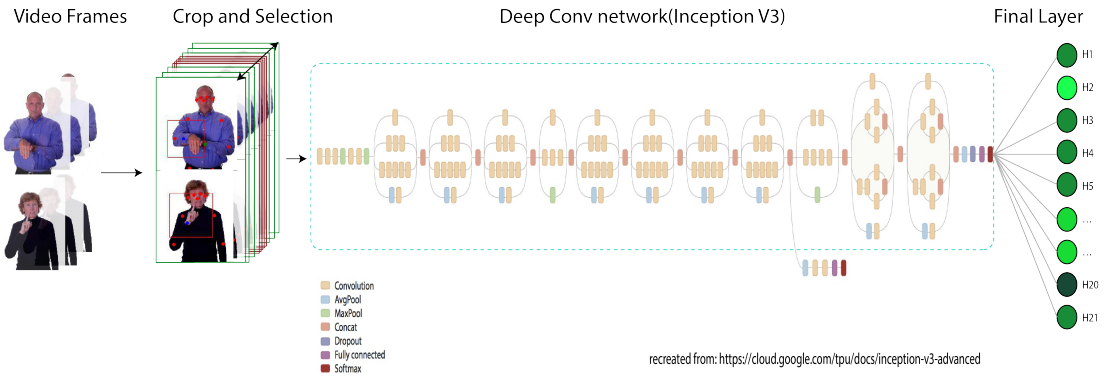

Fig. 5. Handshape Pipeline

the wild, videos produced by ASL users can have different brightness conditions, camera motion blurriness, and low-quality video frames. Deep learning models have shown to exceed human performances in many visual tasks like object recognition [17], medical reading medical imaging [18], and many other visual tasks. For our work, we will be utilizing GoogleNet Inception v3 model. It has been trained on over 1.28 million of images with over 1,000 object categories [19]. Figure 5 shows a layout of the handshape recognition pipeline. From our experiments, we conclude that cropping handshapes from frames before supplying int CNN give a better generalization and accuracies, allows CNN to converge faster to expected results. To reliably detect handshapes out of busy frame images, we have deployed a simple algorithm to extrapolate potential hand palm location using key body positions acquired from pose estimation. It allows us to confidently auto crop the handshapes bounding boxes. In the next phase, we select key handshapes, mostly from the beginning and end of the sign, which also removes blurry transition handshapes.

Once, we have identified key handshapes, we apply traditional image augmentation techniques like random rotations and distortions. With the final set of handshapes, Inception CNN model is retrained that allows us to use the final layer of the model, as an automatic feature extractor. As a result of the handshape pipeline, we can embed video segments of any sign, to fixed vector representation that have shown to generalize well to previously unseen gestures. For the training of the CNN model, we have selected real-world data consisting of 23 gestures with 3 repetitions each from 130 learners. CNN is retrained with handshape images from 23 gestures. For recognition of unseen gestures, we have selected 19 gestures with videos. The unseen gestures are run through the same handshape pipeline, with CNN model acting as a feature extractor that produces final feature vector. Once both unseen and seen sings are embedded into a fixed vector, we calculate a cosine similarity and produce top-5 accuracy.

B. Gesture Expression Matching

This module is responsible for the overall recognition. The incoming ASL video is first divided into frames. Typically, in the video there are on an average 45 frames. Out of these 45 only six are considered for recognition of initial and final handshape and locations. The rest are used for movement identification. The entire recognition follows the steps below:

1) The first three frames are passed to the location recognition module to identify the initial location bucket.
2) the first three frames are passed to the handshape recognition module to identify the handshape alphabet.
3) The keypoints for the next 30 to 40 frames are then passed to the movement recognition module
4) the final three frames are first passed to the location recognition module
5) the final three frames are then passed to the handshape recognition module

The output of these steps provides several combinations of initial location and handshape, movement and final location and handshape. Each such combination generates a string that can be derived from the regular expression in Equation 1. However, to recognize a gesture the derived string should match the definition of the gesture. There can be different degrees of match and in our results, we will consider two specific definitions of a match: a) exact definition match, and b) partial match. A gesture video exactly matches a definition in terms of alphabets if initial defined handshape is among the top five initial recognized handshape, initial defined location is within top three of recognized location, defined movement appears in top three recognized movement, final defined handshape appears in top five recognized handshapes, and final defined location appears in top three recognized locations. A partial match occurs when at most one out of five defined components does not appear in the top 3 recognition.

VI. Results

In this section, we will evaluate the usage of canonical form for zero-shot learning on ASL.

A. Evaluation Metrics

We will consider two granularities of evaluation. **Evaluation of each module:** handshape, location, and movement. For modular evaluation we will consider top k match. Since handshape is the most complex and most diverse component of a gesture, we consider top 5 match. For location and movement, we consider top 3 match. We express match in terms of the alphabet and also in terms of the training gestures from which the alphabet was derived.

Evaluation of the overall recognition: We will consider *success rate* $SR = 100\frac{N_S}{N_T}$, where N_S is the number of unseen test gestures which were successfully recognized using the full definition following the canonical form, and N_T is the total number of test gestures which only have alphabets that are seen in the training examples. We will consider *bloating factor* $B_F = 100\frac{N_S+N_{Tr}}{N_{T_r}}$, where N_{Tr} is the number of training examples. This factor is a measure of our capability to extend gesture vocabulary using limited training examples.

Figure 6 shows the statistics for three separate users not available in the training set of users for the 19 previously unseen gestures. We will use this table to evaluate performance of each module and the overall performance of the zero-shot recognition system. The table lists the test gestures in the first column, where grey boxes indicate successful zero-shot recognition, white boxes indicate unsuccessful recognition, and light grey boxes indicate gestures that introduce new alphabets and are unrecognizable. The second column has the definition of each test gesture following the canonical form of Equation 2. The next five columns show top 5 handshape recognition results, the next three show top 3 movement recognition results and the next 3 show top 3 location recognition results. In this paper, for lack of space, we only show the right hand recognition results but the same results can be available for the left hand. For gestures that have a change in handshape during execution, we have to recognize two handshapes. A zero-shot recognition is determined to be successful if both start and end handshapes are in top 5, movement is in top 3 and location is in top 3.

B. Evaluation of each module

1) **Handshape:** As seen in Figure 6, TAIL, AGAPE, ADVENT have gesture components that contribute new alphabets to the language. Hence, we will consider these gestures to be unrecognizable with the given alphabet set. Yet for evaluating individual components they will be used whenever fit. For handshape recognition, Advent and Agape contribute new handshape alphabets but TAIL does not. From the results table in Figure 6 we see that if we consider top 5 handshape recognition accuracy then our system has an identification accuracy of 70%. Although this is low as compared to several image recognition techniques including handshape recognition, this accuracy is for zero-shot recognition. The only competing technology for zero-shot gesture recognition could achieve accuracy of 51.4% [4]. We believe that the main reason for the improvement in accuracy is that we are encoding concepts in a canonical form which can be generalized across domains, whereas Bilge et al are encoding individual examples in terms of attributes such as 3D CNN activation levels, which inherently do not capture any concept information. Further, our training examples used videos collected in noisy environments with varying backgrounds which by no means match the definition of clean. Our test videos are from a totally different set of human users that were not available in the training set. As a result, the recognition accuracy of 70% is for zero-shot user-independent handshape recognition.

Moreover, we consider gestures where handshape changes during execution. For example, if we consider ADOPT, the initial handshape is H18 but final is H4. The handshape recognition system is actually partially correct in identifying handshape. But we consider the recognition of ADOPT as a failure because the start handshape is not recognized. Two other such examples, ADD and ALLGONE, were recognized even though the handshape changed during execution.

2) **Movement:** The gesture TAIL introduces a new movement in the alphabet. Moreover, this new movement has no motion of the wrist and pivoting the wrist the user waves the index finger. This motion cannot be captured by PoseNet architecture [20]. Hence it is unrecognizable in the system. So, we discard this example from the evaluation. From the results table in Figure 6, we see that the overall accuracy of zero-shot user-independent movement recognition considering top 3 results was 83%. One of the limitations of our movement recognition comes from the restriction in PoseNet where we cannot track the palm. We only have a keypoint on the wrist. The movement captured at the wrist is less vigorous making it difficult to capture.

In our test cases, the word ADVANCE was not recognized due to failure to recognize the movement although handshape and location were recognized correctly. The reason for this is that ADVANCE uses the movement of both hands. The PoseNet results actually confused between the left and right arm and designated some of the right wrist points as left wrist points. This factor resulted in failure of movement recognition.

3) **Location:** The location provides almost perfect accuracy and, which is expected, as we are only using 6 bucket representing general proximity areas, where the signer is using their palms, thus a lot of gestures fall with similar areas. This does not affect results significantly, because we don't consider each module as independent recognizer, but rather treat a configuration of handshape, motion, and location as a whole. Thus, if there are two gestures, that have identical handshape and movement, but executed in a different location, our algorithm would be able to recognize them as different sings. For all the 19 unseen gestures, we calculate the top 3 results, location module correctly finds mapped locations and recognizes the approximate locations.

C. Evaluation of Overall Zero-Shot Recognition

Of the 19 gestures considered for zero-shot learning, three introduced a new alphabet and were considered unrecognizable. Hence, we evaluate our overall zero-shot recognition accuracy out of 16 gestures by omitting TAIL, AGAPE, ADVENT. Out of the 16 unknown gesture examples, we could correctly identify 10 gestures consistently across three previously unseen users giving us a success rate of $S_R = 66.6\%$ for the whole gesture. This is four times higher than competing technology that can achieve a success rate of only 15%. This indicates that given a training gesture set comprising of 23 gestures we can identify an additional 10 gestures only from their definition without obtaining training video. Hence the bloating factor $B_F = 143\%$.

This is a significant result since it can potentially be a significant step towards complete automated sign sequence to sentence translation of any sign language communication. This result indicates that through the learning of the unique concepts of a gesture-based language (the alphabets in this case) it is possible to recognize a large set of gestures given a small and limited set of examples.

	Handshape						Movement			Location		
Gesture	Definition	top 0	top1	top2	top3	top4	top 0	top 1	top 2	top 0	top 1	top 2
task	**H4 L1 M3 H4 L1**	**H4(can)**	H13(here)	**H4(adopt)**	H5(advantage)	**H4(anti)**	**M3**	M0	M8	**add**	**alive**	**appetite**
tail	**H1 L1 M16 H1 L1**	H13(here)	H17(sorry)	H4(can)	H14(hospital)	H3(and)	M1	M8	M3	**help**	**sorry**	**add**
phone	**H10 L4 M3 H10 L4**	H5(cat)	**H10(Gold)**	H8(hello)	H1(hearing)	H1(deaf)	**M3**	**M3**	M0	**deaf**	**father**	**cat**
hurt	**H1 L0 M0 H1 L1**	H17(sorry)	H5(decide)	**H1(tail)**	H14(hospital)	H3(and)	**M0**	M13	M1	**after**	bar	**and**
hearing	**H1 L3 M3 H1 L4**	H5(cat)	H9(gold)	H8(hello)	H8(father)	H15(If)	**M3**	**M3**	**M3**	**tiger**	**cat**	**deaf**
cheer	**H2 L1 M8 H2 L4**	H6(cop)	H17(sorry)	H17(alive)	H6(appetite)	H3(and)	M3	M1	M3	**find**	**help**	**add**
bar	**H5 L0 M2 H5 L1**	H1(hurt)	H16(large)	**H5(decide)**	H12(help)	H17(sorry)	**M2**	**M2**	**M6**	**and**	after	hurt
appetite	**H6 L1 M0 H6 L1**	H3(and)	**H6(cop)**	H17(sorry)	H14(hospital)	H1(deaf)	**M0**	**M0**	**M3**	**help**	**add**	alive
anti	**H4 L4 M7 H4 L4**	H5(advantage)	**H4 (adopt)**	**H4(can)**	H13(here)	**H4(task)**	M0	**M7**	M0	**cat**	**father**	**phone**
allgone	**H18 L0 M2 H4 L1**	H3(and)	H5(Advantage)	**H18(add)**	H6(appetite)	**H4(adopt)**	**M2**	**M2**	M0	**large**	**adopt**	cop
alive	**H17 L1 M1 H17 L1**	H2(cheer)	H6(cop)	H6(appetite)	**H17(sorry)**	H3(and)	**M1**	M6	M13	**add**	**appetite**	cop
agree	**H1 L1 M0 H1 L1**	H4 (can)	H3(and)	H5(advantage)	H4(allgone)	H21(agape)	**M0**	**M0**	M3	**good_night**	**advent**	**cost**
agape	**H21 L4 M0 H21 L1**	H7(cost)	H1(agree)	H1(deaf)	H4(adopt)	H1(tail)	M15	M3	**M0**	**cheer**	**cost**	**advent**
advent	**H20 L1 M1 H20 L4**	H4(anti)	H1(about)	H4(adopt)	H13(here)	H17(sorry)	M9	M8	M0	**cost**	**good_night**	**agape**
advantage	**H5 L0 M0 H5 L0**	H4(can)	H4(adopt)	H3(and)	H4(anti)	H13(here)	M5	**M0**	M13	**about**	**can**	here
advance	**H19 L3 M1 H19 L3**	**H19(adult)**	H8(hello)	H3(goout)	H5(cat)	H10(gold)	M4	M0	M5	**adult**	**go_out**	**hearing**
adult	**H19 L3 M1 H19 L3**	**H19(advance)**	H8(hello)	H3(goout)	H10(gold)	H8(father)	**M1**	M12	M5	**go_out**	**advance**	**hearing**
adopt	**H18 L0 M1 H4 L1**	H19(advantage)	**H4(can)**	H13(here)	**H4(anti)**	H1(tail)	**M1**	M6	M0	large	all_gone	**sorry**
add	**H18 L1 M0 H3 L1**	**H18(allgone (S))**	H4(can)	H1(tail)	**H3(and)**	H1(agree)	M3	**M0**	**M0**	**help**	**sorry**	alive

Fig. 6. Zero-shot recognition results of 19 unseen test gestures from three unseen users. Grey boxes denote successful zero-shot recognition. White boxes denote failure cases. Light gray boxes indicate absence of alphabet in the training gestures. Bold font indicates match, regular font indicates no match.

D. Evaluation on the ASLTEXT dataset

To further evaluate the usefulness of canonical form representation of gestures and its ability to facilitate zero-shot application, we have tested our model performance against the ASLTEXT dataset, we have introduced in an earlier section. We have identified 190 unique gestures and 1200 videos, that are completely disjoint from any gestures and videos that we have trained on. *The principal difference from [4], is that instead of splitting the dataset into 170, 30,50 disjoint classes as train, validation, test set, respectively, we used 190 unique unseen gestures as the test set only and* ***none*** *was used to retrain the model.* It represents the eight-fold test set size increase compared to 23 unique gestures that we have trained on. For each of unseen gesture, we assume the definition of the given class in terms of the ASL alphabets discussed in Definition 2.

For recognition of gestures in the ASLTEXT dataset, we follow the same recognition pipeline protocol described in section V-B. As seen in Figure 7, handshape recognition is performed on all 190 unseen gestures. For the sake of concise representation, we have grouped labels on the left side of the figure. The figure on the right shows mean accuracy scores per group. Example G1 labels that have 100 percent accuracy G10 accuracy of zero percent. Please note that each label has six video instances on average. Out of the 190 unseen or novel gestures, we could correctly identify handshapes of 48 gestures with accuracy $\geq 70\%$, 45 gestures with accuracy $\geq 80\%$, and 16 gestures with 100% accuracy. Each sign on an average had six videos executed by five different users. Out of 1200 test videos for the 190 gestures, we have recorded 66% percent handshape recognition normalized accuracy on the ASLTEXT dataset. For location recognition, for the 190 gestures considered from the dataset we have a 74% accuracy for the top three start and end locations. For movement detection, we obtained an accuracy of 73%.

E. Evaluation of ASLTEXT Zero-Shot Recognition

In the ASLTEXT dataset on an average each gesture has six repetitions by different individuals. As such SR can be parameterized on how many of the repetitions can be recognized correctly. If we consider that 100% of repetitions have to be correctly recognized then we achieve a SR of 3%, i.e. six new gestures. The associated bloating factor is $(23 + 9 + 6)/23 = 165\%$ since we only used training for 23 ASL gestures and we could recognize nine gestures from the IMPACT dataset and 6 from ASLTEXT. With 90%, 80%, and 70% and 60% correct recognition of repetitions we can recognize 7 (SR 3.68 %, BF 169 %), 22 (SR 11.58%, BF 245%), 34 (SR 17.89 %, BF 287%), and 55 (28.95%, BF 378 %) new gestures, respectively.

Comparison with state-of-art: Bilge et al. [4] reports a zero-shot recognition accuracy on ASLTEXT dataset of 51.4% on 50 unseen gestures. However, they have used 170 gestures from ASLTEXT as training, whereas we have not used any examples from ASLTEXT for training. Moreover, the accuracy metric used does not specify how many unseen gestures were actually recognized. If we consider the total number of videos correctly recognized out of 1200 from 190 gestures, we report a normalized accuracy of 66%. This 13.6% increase in accuracy is significant because we have not used any part of ASLTEXT as training.

VII. Discussion and Conclusions

In this paper, we demonstrate one usage of the canonical form representation of gestures. Zero-shot recognition of gestures is useful because with training data available from a small subset of gestures many more unseen gestures with definitions can be recognized. However, there are several other advantages of a canonical form representation. A canonical form is in terms of handshape, location and movement and is independent of any sign language semantics. As such the same alphabet can be associated with semantics specific to a different sign language. Hence, the canonical form can be

Groups	Labels
G1	AHEAD,AVERAGE,BOY,CAN,EMBARRASS,EMPHASIZE,FAMILY,FREE,FRIDAY,GHOST,HOW-MANYORMANY,INTRODUCE,MACHINE,MATCH,PASS,SET-UP
G2	AFRAID,AVOIDORFALL-BEHIND,MAD,PROCEED,LIVE,SAUSAGEORHOT-DOG,BANANA,CHAINOROLYMPICS,CHASE,COAT,EARTH,FAR,FENCE,FREEZE,LUNGS,TAKE-UP
G3	ACT,APPLE,BICYCLE,BOSS,BUT,COMB,DESTROY,DRESSORCLOTHES,FOLLOW,MEAT,MEET,METAL,RUN-OUT,DISCONNECT,CAR,DEAF
G4	ANY,CENTER,COUNTRY,CRUEL,EVERYDAY,FINALLY,GREEN,HELLO,BLAME,OVERORAFTER
G5	ASSOCIATION,COME-ON,COOPERATEORUNITE,GOVERNMENT,GRAB-CHANCE,GRASS,HOSPITAL,MAKE,MORNING,MOST,ONE-MONTH,SKIN,STRONG,DEPOSIT,LETTERORMAIL,MESSED-UP,COURT
G6	APPOINTMENT,ARRIVE,COLLECT,DECIDE,DRY,ENGAGEMENT,EXACT,FOOTBALL,GAMBLE,HALLOWEEN,LIPOR MOUTH,PRICE,SHAPEORSTATUE,INCLUDEORINVOLVE,DISAPPOINT,DRUNK,MERGEORMAINSTREAM
G7	BREAD,COUGH,COURSE,CRUSH,DISAPPEAR,EXPENSIVE,GASORGAS-UP,GIRL,IDEA,INSULT,INSURANCEORINFECTION,LIBRARY,MAGAZINE,ONE,WHERE,BRAVEORRECOVER,BAD,BO RE,BREAK-DOWN,CHERISH,DIVORCE,FORGET,FRIEND,GONE,GROW,LEFT,MOSQUITO,PROTEST
G8	BAR,HEAD-COLD,HELMET,ILLEGAL,COLD,GOAL
G9	ALONE,BAWL-OUT,BLACK,EXPLAIN,HARD,NOT-MIND,CANNOT,EAST,GRANDFATHER,GRANDMOTHER,HEAD,HEAVY,PAINT,WORK-OUT,AGAIN,FLY-BY-PLANE,MISSORASSUME,NICEORCLEAN,SHAME,ARTORDESIGN,A-LOT,CONFLICTORINTERSECTION
G10	ANSWER,EXPERT,CANCELORCRITICIZE,ACCEPT,ADVISEORINFLUENCE,AUTUMN,BEAUTIFUL,BLUE,CALL-BY-PHONE,CELEBRATE,DARK,DIRTY,DISMISS,DOWN,EAT,EXPERIENCE,EXPERIMENT,FED-UPORFULL,FULL,GENERAL,GENERATION,GET-UP,GRADUATE,HAPPEN,HAVE,HIT,HOME,INFORM,INJECT,LEARN,LESS-THAN,LIE,LINE,MEMBER,MONDAY,NAB,PULL,REALLY,SAME-OLD,SILLY,TO-FOOL,TRASHORBAG

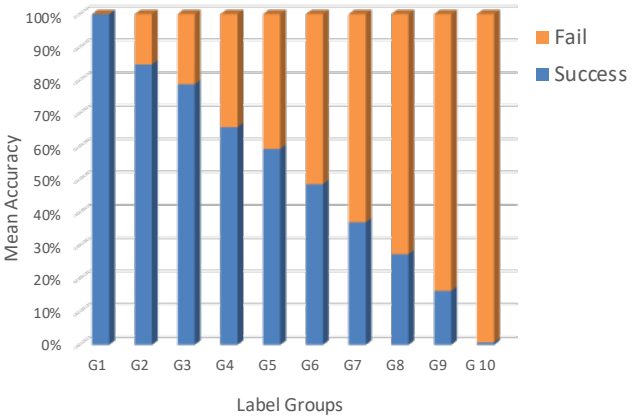

Fig. 7. Zeroshot recognition accuracy for handshape from 190 unique ASL gestures from the ASLTEXT dataset. The gestures are grouped into 10 clusters with respect to their recognition accuracy. 48 gestures have a recognition accuracy $\geq 70\%$ computed for an average of six videos per gesture.

independent of the language. If we can develop a module that can automatically convert a sequence of gestures in any language into a canonical form, then it can enable automated translation between sign languages.

Another advantage of a canonical form is gesture-based search and mining. This can be useful in the disabilities education domain. Gesture transcripts of educational material for the Deaf and Hard of Hearing students can be searched using gesture inputs.

Linguistics research in the domain of ASL has long attempted to develop a common transcription language for gestures. Efforts have resulted in resources such as SignType [21], which is an extensive and granular method of representing ASL gestures. A goal of this research is to automatically convert ASL gestures into a representation like SignType. However, SignType examples are currently generated through and have significant variance and are not currently usable.

We have collected video recordings of gesture performances from 130 users on 23 ASL gestures with 3 repetitions each resulting in a total of 8970 videos. If the paper is accepted for publication, we will make the dataset public.

For zero-shot recognition of gesture videos, we show greater than 15% improvement over currently existing technology. Our system achieves better zero-shot accuracy because it focuses on learning useful concepts from limited examples and uses them through canonical forms to compose other gestures.

REFERENCES

[1] D. J. Napoli and J. Wu, "Morpheme structure constraints on two-handed signs in american sign language: Notions of symmetry," *Sign language & linguistics*, vol. 6, no. 2, pp. 123–205, 2003.

[2] E. Kodirov, T. Xiang, and S. Gong, "Semantic autoencoder for zero-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3174–3183.

[3] B. Romera-Paredes and P. Torr, "An embarrassingly simple approach to zero-shot learning," in *International Conference on Machine Learning*, 2015, pp. 2152–2161.

[4] Y. C. Bilge, N. Ikizler-Cinbis, and R. G. Cinbis, "Zero-shot sign language recognition: Can textual data uncover sign languages?" *arXiv preprint arXiv:1907.10292*, 2019.

[5] N. Madapana and J. P. Wachs, "Hard zero shot learning for gesture recognition," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 3574–3579.

[6] Y. Xian, B. Schiele, and Z. Akata, "Zero-shot learning-the good, the bad and the ugly," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4582–4591.

[7] P. Paudyal, J. Lee, A. Kamzin, M. Soudki, A. Banerjee, and S. K. Gupta, "Learn2sign: Explainable ai for sign language learning." in *IUI Workshops*, 2019.

[8] A. Khosrowpour, J. C. Niebles, and M. Golparvar-Fard, "Vision-based workface assessment using depth images for activity analysis of interior construction operations," *Automation in Construction*, vol. 48, pp. 74–87, 2014.

[9] A. E. F. Da Gama, T. M. Chaves, L. S. Figueiredo, A. Baltar, M. Meng, N. Navab, V. Teichrieb, and P. Fallavollita, "Mirrarbilitation: A clinically-related gesture recognition interactive tool for an ar rehabilitation system," *Computer methods and programs in biomedicine*, vol. 135, pp. 105–114, 2016.

[10] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong, "Transductive multi-view zero-shot learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 11, pp. 2332–2345, 2015.

[11] SignSavvy, 2019, last accessed 1 September. [Online]. Available: https://www.signingsavvy.com

[12] W. C. Stokoe Jr, "Sign language structure: An outline of the visual communication systems of the american deaf," *Journal of deaf studies and deaf education*, vol. 10, no. 1, pp. 3–37, 2005.

[13] H. Lalazar, L. Abbott, and E. Vaadia, "Tuning curves for arm posture control in motor cortex are consistent with random connectivity," *PLoS computational biology*, vol. 12, no. 5, p. e1004910, 2016.

[14] T. Zhou, "Early turn-taking prediction for human robot collaboration," Ph.D. dissertation, Purdue University, 2018.

[15] L. Zhou, W. Li, P. Ogunbona, and Z. Zhang, "Jointly learning visual poses and pose lexicon for semantic action recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[16] C. Neidle, A. Thangali, and S. Sclaroff, "Challenges in development of the american sign language lexicon video dataset (asllvd) corpus," in *5th Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon, LREC*. Citeseer, 2012.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[18] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, p. 115, 2017.

[19] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[20] G. Papandreou, T. Zhu, N. Kanazawa, A. Toshev, J. Tompson, C. Bregler, and K. Murphy, "Towards accurate multi-person pose estimation in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4903–4911.

[21] H. Van Der Hulst and R. Channon, *Notation systems*. na, 2010.