AMERICAN SIGN LANGUAGE FINGERSPELLING RECOGNITION IN THE WILD

Bowen Shi[1]*, Aurora Martinez Del Rio*[2]*, Jonathan Keane*[2]*, Jonathan Michaux*[1]
Diane Brentari[2]*, Greg Shakhnarovich*[1]*, and Karen Livescu*[1]

[1]Toyota Technological Institute at Chicago, USA [2]University of Chicago, USA
{bshi, jmichaux, greg, klivescu}@ttic.edu, {amartinezdelrio, dbrentari,jonkeane}@uchicago.edu

ABSTRACT

We address the problem of American Sign Language fingerspelling recognition "in the wild", using videos collected from websites. We introduce the largest data set available so far for the problem of fingerspelling recognition, and the first using naturally occurring video data. Using this data set, we present the first attempt to recognize fingerspelling sequences in this challenging setting. Unlike prior work, our video data is extremely challenging due to low frame rates and visual variability. To tackle the visual challenges, we train a special-purpose signing hand detector using a small subset of our data. Given the hand detector output, a sequence model decodes the hypothesized fingerspelled letter sequence. For the sequence model, we explore attention-based recurrent encoder-decoders and CTC-based approaches. As the first attempt at fingerspelling recognition in the wild, this work is intended to serve as a baseline for future work on sign language recognition in realistic conditions. We find that, as expected, letter error rates are much higher than in previous work on more controlled data, and we analyze the sources of error and effects of model variants.

Index Terms— American Sign Language, fingerspelling, connectionist temporal classification, attention models

1. INTRODUCTION

Sign languages, consisting of sequences of grammatically structured handshapes and gestures, is a chief means of communication among deaf people around the world.[1] In the US, American Sign Language (ASL) is the primary language for about 350,000 to 500,000 deaf people [1] and is used by many others as a second language. Automatic recognition of sign language would help facilitate communication between deaf and hearing individuals. It could also enable services such as search and retrieval in deaf social and news video media, which often has little or no text associated with it.

A number of challenges are involved in sign language recognition. Sign language employs multiple elements such as handshapes, arm movement and facial expressions. All of these gestures are subject to coarticulation and phonological effects, so they often do not appear in their canonical forms. In addition, there is a great deal of variability in the appearance of different signers' hands and bodies. Finally, the linguistics of sign language is less well studied than that of spoken language, and there is much less annotated data than there is for spoken languages.

In this paper we focus on the recognition of ASL fingerspelling, a component of ASL in which words are signed by a series of handshapes or trajectories corresponding to single letters (using the English alphabet). The ASL fingerspelling alphabet is shown in Figure 1. Fingerspelling is mainly used for lexical items that do not have their own ASL signs, such as proper nouns or technical terms, which are often important content words. Overall, fingerspelling accounts for 12–35% [2] of ASL, and appears frequently in technical language, colloquial conversations involving names, conversations involving current events, emphatic forms, and the context of codeswitching between ASL and English [3, 4, 5]. Transcribing even only the fingerspelled portions of videos in online media could add a great deal of value, since these portions are often dense in content words.

Compared to sign language recognition in general, fingerspelling recognition is in some ways more constrained because it involves a limited set of handshapes, and in ASL it is produced with a single hand (unlike in some other sign languages), which makes hand occlusion less problematic [6].[2] On the other hand, fingerspelling recognition presents its own challenges. It involves very quick, small motions that can be highly coarticulated. In lower-quality video, motion blur can be very significant during fingerspelled portions.

Most publicly available sign language data sets have been collected in a studio or other carefully controlled environment. Collecting such data is expensive and time-consuming, and as a result most existing sign language data sets are fairly small. On the other hand, there are large amounts of fingerspelling (and, more generally, sign language) video available online on deaf social media and news sites (e.g.,

[1]In some settings there is a cultural distinction between the terms "deaf" and "Deaf". In this paper, we use the term "deaf" to refer to both.

[2]Two-handed fingerspelling occasionally occurs, including in our data.

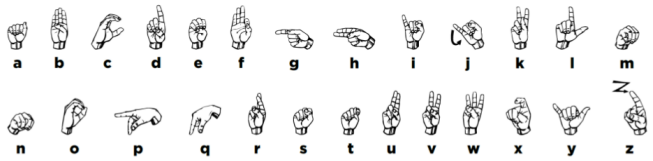

Fig. 1. The ASL fingerspelling alphabet, reproduced from [7].

deafvideo.tv, aslized.org).

In this paper we focus on recognition of fingerspelling occurring in online videos, which allow us to study the recognition problem in a more natural and practical setting than in previous work. This work is to our knowledge the first attempt at fingerspelling recognition (or any sign language recognition) "in the wild", that is in naturally occurring video. For this purpose we use a newly collected data set (see Section 3), which we make available publicly.[3] Collecting and annotating such naturally occurring video clips involves some effort, but it is much quicker to obtain a large quantity of video from a large variety of signers than in studio data collection.

Our recognizer consists of a hand detector trained to detect the signing hand, whose output (the cropped signing hand image) is fed to an end-to-end neural sequence model. As expected, we find that our new data set is challenging, leading to accuracies that are significantly lower than previously reported results on studio-based data sets [8] when using similar models. We explore a number of neural sequence models including encoder-decoder and connectionist temporal classification (CTC)-based models. Our experiments show the importance of the signing hand detector for obtaining high-resolution regions of interest, and that CTC-based models outperform encoder-decoder models on our task. We analyze the sources of error and the effect of a number of design choices.

2. RELATED WORK

There has been a significant amount of work on automatic sign language recognition. Video corpora have been collected for a variety of sign languages [9, 10, 11]. These data sets are all recorded in a studio environment, which makes the variability lower than in natural day-to-day signing. One example of a more naturalistic data set is the RWTH-PHOENIX-Weather corpus [12], which contains German Sign Language in the context of daily television weather forecasts; however, the number of signers is still limited (7 signers) and the visual variability is fairly controlled. Fingerspelling-specific data sets are much rarer. The ChicagoFSVid data set is the largest of which we are aware; it includes 4 native ASL signers fingerspelling 600 sequences each, and has been used in recent work on lexicon-free recognition and signer adaptation [13, 8]. The National Center for Sign Language and Gesture Resources (NCSLGR) Corpus includes about 1,500 fingerspelling sequences (as well as a variety of other ASL signs) [14, 15]. In addition to video, many efforts have been devoted to using depth sensors instead of or in addition to video, which can be very helpful for developing new interfaces [6, 16]. In this work, however, we focus on naturally occurring online data, which is typically in the form of video.

Automatic sign language recognition can be approached similarly to speech recognition, with image frames and signs being treated analogously to audio signals and words or phones respectively. As in a number of other domains, convolutional neural networks (CNNs) have recently been replacing engineered features in sign language recognition research [17, 18, 19, 11, 8]. For sequence modeling, most previous work has used hidden Markov models (HMMs) [20, 17, 18, 13], and some has used segmental conditional random fields [21, 22, 13]. Much of this work relies on frame-level labels for the training data. Due to the difficulty of obtaining frame-level annotation, recent work has increasingly focused on learning from sequence-level labels alone [17, 11, 19].

Specifically for fingerspelling recognition, most prior work has focused on restricted settings. When the lexicon is restricted to a small size (20 - 100 words), letter error rates lower than 10% have been achieved [23, 24, 25]. Another important restriction is the signer identity. In [13, 8], letter error rates of less than 10% were achieved in a lexicon-free setting (unrestricted vocabulary) when training and testing on the same signer, but the error rate increases to above 40% in the signer-independent setting. The large performance gap between these two settings has also been observed in general sign language recognition [12].

Most fingerspelling recognition approaches begin by extracting the signing hand from the image frames [21, 13, 11]. Due to the high quality of video used in prior work, hand detection (or segmentation) is usually treated as a preprocessing step with high accuracy, with little analysis of its impact on performance. In our new data set, the variation in hand appearance, motion blur, and backgrounds makes the hand extraction problem much more challenging.

3. DATA

We have collected a new data set consisting of fingerspelling clips from ASL videos on YouTube, aslized.org and deafvideo.tv. ASLized is an organization that creates educational videos that pertain to the use, study, and structure of ASL. DeafVideo.tv is a social media website for deaf vloggers, where users post videos on a wide range of topics. The videos include a variety of viewpoints and styles, such as webcam videos and lectures. 214 raw ASL videos were collected, and all fingerspelling clips within these videos were

[3]The data set is available for download from http://ttic.edu/livescu/chicago-fingerspelling-in-the-wild.

Fig. 2. Illustrations of ambiguity in fingerspelled handshapes. Upper row: different letters with similar handshapes, all produced by the same signer. Lower row: the same letter (u) signed by different signers.

manually located and annotated.

The videos were annotated by in-house annotators at TTIC and the U. Chicago Sign Language Linguistics lab, using the ELAN video annotation tool [26].Annotators viewed the videos, identified instances of fingerspelling within these videos, marked the beginning and end of each fingerspelling sequence, and labeled each sequence with the letter sequence being fingerspelled. No frame-level labeling has been done; we use only sequence-level labels. Annotators also marked apparent misspellings and instances of fingerspelling articulated with two hands. The fingerspelled segments include proper nouns, other words, and abbreviations (e.g., N-A-D for National Association of the Deaf). The handshape vocabulary contains the 26 English letters and the 5 special characters {<sp>, &, '„.,@} that occur very rarely.

We estimate the inter-annotator agreement on the label sequences to be about 94%, as measured for two annotators who both labeled a small subset of the videos; this is the letter accuracy of one annotator, viewing the other as reference.

As a pre-processing step, we removed all fingerspelling video sequences containing fewer frames than the number of labels. We split the remaining data (7304 sequences) into 5455 training sequences, 981 development (dev) sequences, and 868 test sequences. Using frames per second (FPS) as a proxy for video quality, we ensured that the distribution of FPS was roughly the same across the three data partitions. The dataset includes about 168 unique signers (91 male, 77 female).[4] 192 of the raw videos contain a single signer, while 22 videos contain multiple people. Each unique signer is assigned to only one of the data partitions. The majority of the fingerspelling sequences are right-handed (6782 sequences), with many fewer being left-handed (522 sequences) and even fewer two-handed (121 sequences). Roughly half of the sequences come from spontaneous sources such as blogs and interviews; the remainder comes from scripted sources such as news, commercials, and academic presentations. The frame resolution has a mean and standard deviation of $640 \times 360 \pm 290 \times 156$. Additional statistics are given in Figure 3.

This data set collected "in the wild" poses serious challenges, such as great visual variability (due to lighting, background, camera angle, recording quality) and signing variability (due to speed and hand appearance). To illustrate some of these challenges, Figure 2 shows a number of representative frames from our data set. There can be a great deal of variability in fingerspelling the same letter, as illustrated in the bottom row of Figure 2. In addition, many fingerpselled letters have similar handshapes. For example, the letters *a*, *s*, *t* and *n* are only distinguished by the position of the thumb, and the letters *r*, *u* and *v* are all signed with the index and middle fingers extended upward. The small differences among these letters can be even harder to detect in typical lower-quality online video with highly coarticulated fingerspelling, as seen in the top row of Figure 2.

[4]These numbers are estimated by visual inspection of the videos, as most do not include meta-data about the signer.

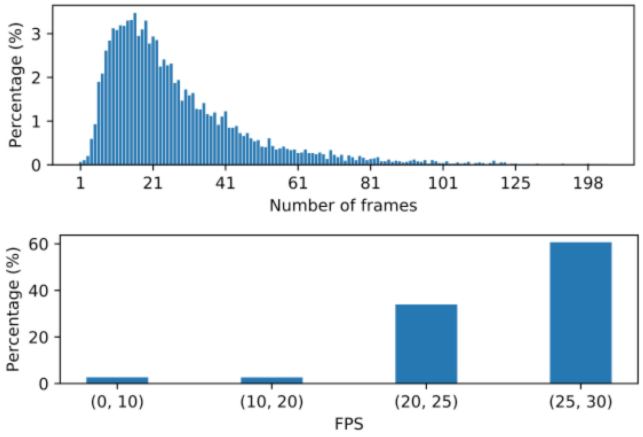

Fig. 3. Histograms of the number of frames per fingerspelled sequence and frames per second (FPS) for fingerspelled sequences in the data set.

4. MODEL

Our approach to fingerspelling recognition consists of a signing hand detector followed by a sequence recognizer, illustrated in Figure 4.

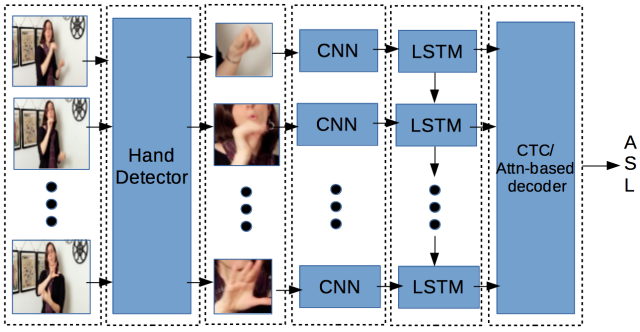

Fig. 4. Sketch of our approach. After the hand detector component, the rest of the model is trained end-to-end.

4.1. Signing hand detection

The hand detection problem here is somewhat different from typical hand detection. A large proportion of the video frames contain more than one hand, but since ASL finger-

spelling generally involves a single hand, the objective here is to detect the *signing hand*. This can be viewed as a problem of action localization [27]. As in prior work on action localization [27], we train a detection network that takes as input both the image appearance and optical flow, represented as a motion vector for every pixel computed from two neighboring frames [28].[5] For the detection network, we adapt the design of the Faster R-CNN object detector [30]. As in [30], the detector is based on an ImageNet-pretrained VGG-16 network [31, 32]. Unlike the general object detector, we only preserve the first 9 layers of VGG-16 and the stride of the network is reduced to 4. Lower layers able to capture more fine details [33] combined with finer stride/localization are beneficial for detecting signing hands, which tend to be small relative to the frame size.

Unlike much work in action localization [27], which processes optical flow and appearance images in two distinct streams, we concatenate the optical flow and RGB image as the input to a single CNN. In our video data, motion involves many background objects like faces and non-signing hands, so a separate optical flow stream may be misleading.

Given bounding boxes predicted framewise by the Faster R-CNN, we first filter them by spatial non-maxima suppresion (NMS) [34], greedily removing any box with high overlap with a higher-scoring box in the same frame. Next, we link the surviving boxes across time to form a video region likely to be associated with a fingerspelling sequence, which we call a "signing tube" (analogously to action tubes in action recognition [35]). Even after NMS, there may be multiple boxes in a single frame (e.g., when the signer is signing with both hands). Our temporal linking process helps prevent switching between hands in such cases. It also has a smoothing effect, which can reduce errors in prediction compared to that based on a single frame.

More formally, the input to the signing tube prediction is a sequence of sets of bounding box coordinate and score pairs: $\{(b_t^1, s_t^1), (b_t^2, s_t^2), ..., (b_t^n, s_t^n)\}$, $1 \leq t \leq T$, produced by the frame-level signing hand detector. The score s_t^i is the probability of a signing hand output by the Faster R-CNN. We define the *linking score* of two boxes b_t^i and b_{t+1}^j in two consecutive frames as:

$$e(b_t^i, b_{t+1}^j) = s_t^i + s_{t+1}^j + \lambda * IoU(b_t^i, b_{t+1}^j) \tag{1}$$

where $IoU(b_t^i, b_{t+1}^j)$ is the intersection over union of b_t^i and b_{t+1}^j and λ is a hyperparameter that is tuned on held-out data. Generation of the optimal signing tube is the problem of finding a sequence of boxes $\{b_1^{l_1}, ..., b_T^{l_T}\}$ that maximizes the sequence score, defined as

$$E(l) = \frac{1}{T} \sum_{t=1}^{T-1} e(b_t^{l_t}, b_{t+1}^{l_{t+1}}) \tag{2}$$

[5]For optical flow we use the OpenCV implementation of [29].

This optimization problem is solved via a Viterbi-like dynamic programming algorithm [36].

4.2. Fingerspelling sequence model

We next take the signing tube, represented as a sequence of image patches $\{\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_T\}$, as input to a sequence model that outputs the fingerspelled word(s) w. We work in a lexicon-free setting, in which the word vocabulary size is unlimited, and represent the output w as a sequence of letters $w_1, w_2, ..., w_s$. The model begins by applying several convolutional layers to individual image frames to extract feature maps. The convolutional layers transform the frame sequence $\{\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_T\}$ into a sequence of features $\{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_T\}$.

The sequence of image features $\{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_T\}$ is then fed as input to a long short-term memory recurrent neural network (LSTM) [37] that models the temporal structure, producing a sequence of hidden state vectors (higher-level features) $\{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_T\}$. Given the features produced by the LSTM, the next step is to compute the probabilities of the letter sequences $w_1, w_2, ..., w_s$. We consider two approaches for decoding, neither of which requires frame-level labels at training time: an attention-based LSTM decoder, and connectionist temporal classification (CTC) [38]. In the former case, the whole sequence model becomes a recurrent encoder-decoder with attention [39].

In the **attention-based model**, temporal attention weights are applied to $(\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_T)$ during decoding, which allows the decoder to focus on certain chunks of visual features when producing each output letter. If the hidden state of the decoder LSTM at timestep t is $\mathbf{d}_t$, the probability of the output letter sequence is given by

$$\begin{aligned}
&\alpha_{it} = \text{softmax}(\mathbf{v}_d^T \tanh(\mathbf{W}_e \mathbf{e}_i + \mathbf{W}_d \mathbf{d}_t)) \\
&\mathbf{d}'_t = \sum_{i=1}^{T} \alpha_{it} \mathbf{e}_i \\
&p(w_t | w_{1:t-1}, \mathbf{e}_{1:T}) = \text{softmax}(\mathbf{W}_o[\mathbf{d}_t; \mathbf{d}'_t] + \mathbf{b}_o) \\
&p(w_1, w_2, \ldots, w_s | \mathbf{e}_{1:T}) = \prod_{t=1}^{s} p(w_t | w_{1:t-1}, \mathbf{e}_{1:T})
\end{aligned} \tag{3}$$

where $\mathbf{d}_t$ is given by the standard LSTM update equations [37]. The model is trained to minimize log loss.

In the **CTC-based model**, for an input sequence of m-dimensional visual feature vectors $\mathbf{e}_{1:T}$, we define a continuous map $\mathcal{N}_w : (\mathcal{R}^m)^T \mapsto (L')^T$ representing the transformation from m-dimensional features $\mathbf{e}_{1:T}$ to frame-level label probabilities and a many-to-one map $\mathcal{B} : {L'}^T \mapsto L^{\leq T}$ where $L^{\leq T}$ is the set of all possible labelings. Letting L be the output label vocabulary, $L' = L \cup \{blank\}$, and y_k^t the probability of label k at time t, the posterior probability of any labeling $\pi \in {L'}^T$ is

$$p(\pi | \mathbf{e}_{1:T}) = \prod_{t=1}^{T} y_{\pi_t}^t = \prod_{t=1}^{T} \text{softmax}_{\pi_t}(\mathbf{A}^e \mathbf{e}^t + \mathbf{b}^e) \tag{4}$$

At training time, the probability of a given labeling $w = w_1, w_2, ..., w_s$ is obtained by summing over all the possible frame-level labelings π, which can be computed by a forward/backward algorithm:

$$p(w|\mathbf{e}_{1:T}) = \sum_{\pi \in \mathcal{B}^{-1}(w)} p(\pi|\mathbf{e}_{1:T}) \quad (5)$$

The CTC model is trained to optimize this probability for the ground-truth label sequences.

Finally, in decoding we combine these basic sequence models with an RNN language model. To decode with a language model, we use beam search to produce several candidate words at each time step and then rescore the hypotheses in the beam using the summed score of the sequence model, weighted language model, and an insertion penalty to balance the insertion and deletion errors. The language model weight and insertion penalty are tuned.

5. EXPERIMENTS

All of the experiments are done in a signer-independent, lexicon-free (open-vocabulary) setting using the data set and partitions described in Section 3.

Evaluation We measure the letter accuracy of predicted sequences, as is commonly used in sign language recognition and speech recognition: $Acc = 1 - \frac{S+I+D}{N}$, where S, I and D are the numbers of substitutions, insertions, and deletions (with respect to the ground truth) respectively, and N is the number of letters in the ground-truth transcription.

Hand detection details We manually annotated every frame in 180 video clips from our training set with signing and non-signing hand bounding boxes.[6] Of these, 123 clips (1667 frames) are used for training and 19 clips (233 frames) for validation. All images are resized to $640 \times 368 \times 3$. We use stochastic gradient descent (SGD) for optimization, with initial learning rate 0.001 and decreased by a factor of 2 every 5 epochs. We apply greedy per-frame NMS with intersection-over-union (IoU)[7] threshold of 0.9, until 50 boxes/frame remain. The bounding boxes are then smoothed as described in Section 4.1. λ is tuned to 0.3, which maximizes the proportion of validation set bounding boxes with IoU > 0.5. Using our bounding box smoothing approach, the proportion of bounding boxes with IoU > 0.5 is increased from 70.0% to 77.5%.

Letter sequence recognition details The input to the recognizer is a bounding box of the predicted signing hand region. All bounding boxes are resized to 224×224 before being fed to the sequence model. For the convolutional layers of the sequence model, we use AlexNet [40] pre-trained on ImageNet as the base architecture.[8] For the recurrent network, we use a single-layer long short-term memory (LSTM) network with 600 hidden units. (A model with more recurrent layers does not consistently improve performance.) The network weights are learned using mini-batch stochastic gradient descent (SGD) with weight decay. The initial learning rate is 0.01 and is decayed by a factor of 10 every 15 epochs. Dropout with a rate of 0.5 is used between fully connected layers of AlexNet. The batch size is 1 sequence in all experiments. The hyperparameters were tuned to maximize the dev set letter accuracy. The language model is a single-layer LSTM with 600 hidden units, trained on the letter sequences in our training set.

5.1. Main results

Table 1 shows the performance of our models ("Hand") using the cropped hand region as input to the sequence model, as well as of a baseline model ("Global") with the same sequence model architecture but without hand detection (i.e., taking the whole image as input). This baseline model is based on commonly used approaches for video description [41]. For the Global baseline, image frames are resized to 224×224 due to memory constraints. We also report the result of a "guessing" baseline ("LM") that predicts words directly from our language model. This baseline only uses statistics of fingerspelled letter sequences, uses no visual input for prediction, and always predicts the output of a greedy decoding of the language model.

The Global baseline outperforms the language model baseline by a small margin, suggesting that the full-image model is unable to use much of the visual information. Compared to the baseline, our approach with hand detection performs much better. The hand detection step both filters out irrelevant information (e.g. background, non-signing hand) and allows us to use higher resolution image regions. CTC-based models consistently outperform the encoder-decoder models on this task. Since fingerspelling sequences are expected to have largely monotonic alignment with the video, this may benefit the simpler CTC model.

Human performance Although we do not have a precise measure of human performance,[9] we estimated it informally in the following way. We took a small set of (153) fingerspelling sequences that were located and labeled by one annotator who had access to the full videos as usual. Another annotator then labeled the fingerspelling sequences only, without access to the surrounding video. Relative to the first annotator, the second annotator had a letter accuracy of 82.7%. We do not have this measure on our full test set, and did not carefully control for the order of presentation of the data to

[6] The non-signing hand category is annotated in training to help the detector learn the distinction between signing hands, other hands, and background; once the detector is trained we ignore the non-signing hand category, and only use the signing hand detections.

[7] IoU is the ratio of the area of overlap over area of union of two regions.

[8] A deeper network like VGG cannot be used due to the memory requirements introduced by its small stride.

[9] Inter-annotator agreement is not a good measure, since the annotators see the entire video surrounding each fingerspelled sequence, while the automatic recognizers see only the fingerspelled sequence.

Table 1. Letter accuracies (%) for the language model and global baseline models and our hand detector-based models.

	LM	Global enc-dec	Global CTC	Hand enc-dec	Hand CTC
Test Acc (%)	9.4	12.7	10.0	35.0	41.9

Table 2. Percentage of several important substitution error pairs on the development set. For a given label pair ($x_1 \rightarrow x_2$), this is the percentage of occurrences of the ground-truth label x_1 that are recognized as x_2.

	(u → r)	(o → e)	(y → i)	(w → u)	(j → i)
%	17.0	11.7	7.9	7.6	6.7

the annotator; but this provides a rough idea of the difficulty of the task for humans.

Additional model variants Besides the proposed model, we also considered a number of variants that we ultimately rejected. The bounding boxes output by the hand detector fail to contain the whole hands in many cases. We considered enlarging the predicted bounding box by a factor of s in width and height before feeding it to the recognizer. In addition, we also considered using optical flow as an additional input channel to the sequence model (in addition to the hand detector), since motion information is important in our task.

We find that neither of these variants consistently and/or significantly improves performance (on the dev set) compared to the baseline with $s = 1$ and no optical flow input. Thus we do not pursue these model variants further.

Error analysis The most common types of errors are deletions, followed by insertions. The encoder-decoder model makes more insertion errors and fewer deletion errors than the CTC model, that is its error types are more balanced, but its overall performance is worse. The most common substitution pairs for the CTC model are (u → r), (o → e), (y → i), (w → u), (y → i) and (j → i) (see Table 2). (u → r), (y → i) and (w → u) involve errors with infrequent letters, which may be due to the relative dearth of training data for these letters. The pair (j → i) is interesting in that the most important difference between them is whether the gesture is dynamic or static. Compared to studio data, the frame rates in our data are much lower, which may make it more difficult to distinguish between static and dynamic letters with otherwise similar handshapes.

Since deletions are the most frequent error type, applying an insertion/deletion penalty is one possible way to improve performance. Using such a penalty produces a negligibly small improvement, as seen in Table 3.

To measure the impact of video quality on performance, we divided the dev set into subsets according to the frame rate (FPS) and report the average error in each subset (see Figure 5). In general, higher frame rate corresponds to higher accuracy.

Effect of the language model Next we consider to what extent the language model improves performance. It is not clear how much the language model can help, or what training

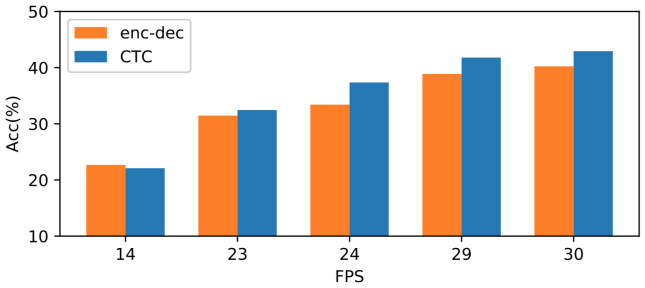

Fig. 5. Development set accuracy for sequences with different frame rates (FPS) for our CTC and encoder-decoder models.

Table 3. Development set letter accuracies (%) when decoding with a language model (lm: LM trained with words from our training set, beam: beam search, ins: insertion penalty, no LM: greedy decoding).

	no LM	+ beam	+ beam + ins	+ beam + ins +lm
CTC	41.1	41.1	41.4	42.8
Enc-dec	35.7	35.8	35.9	36.7

material is best, since fingerspelling does not follow the same distribution as English words and there is not a great deal of transcribed fingerspelling data available. In addition to training on the letter sequences in our own training set, we also consider training on all words in the CMUdict (version 0.7a) dictionary [42], which contains English words and common names, and no improvement was found. The development set perplexity of our LM trained with in-house data is 17.3. Since the maximum perplexity is 32 (31 characters plus end-of-sequence), this perplexity reflects the difficulty of learning the statistics of fingerpselled letter sequences. We also experimentally check the effect of the insertion penalty and beam search. The beam size, language model weight, and insertion penalty are tuned and the best development set results are given in Table 3. Using a language model, the accuracy is improved by a small margin (∼1%).

6. CONCLUSION

This work has studied for the first time the recognition of ASL fingerspelling in naturally occurring online videos. Our newly collected data set includes a variety of challenging visual conditions. We have seen that a purpose-built hand detector, with smoothing over time, is very helpful. The best test set letter accuracies we obtain, using a CTC-based recognizer, are around 42%, indicating that there is room for much future work. Although our data set is the largest fingerspelling data set to our knowledge, it is still much smaller than typical speech recognition corpora, and we are continuing to collect additional online video data.

Acknowledgements

We are grateful to our data annotators Raci Lynch and Amy Huang, and to Raci Lynch also for data collection and help with the annotation setup. This research was funded by NSF grants 1433485 and 1409886.

7. REFERENCES

[1] R.E. Mitchell, T.A. Young, B. Bachleda, and M.A. Karchmer, "How many people use ASL in the United States? Why estimates need updating," *Sign Language Studies*, vol. 6, no. 3, pp. 306–335, 2006.

[2] C. Padden and D.C. Gunsauls, "How the alphabet came to be used in a sign language," *Sign Language Studies*, pp. 10–33, 4 (1) 2003.

[3] D. Brentari and C. Padden, "A language with multiple origins: Native and foreign vocabulary in American Sign Language," in *Foreign Vocabulary in Sign Language: A Crosslinguistic Investigation of Word Formation*, pp. 87–119. Lawrence Erlbaum, 2001.

[4] C. Padden and D.C. Gunsauls, "The ASL lexicon," *Sign Language and Linguistics*, vol. 1, no. 1, pp. 39–60, 1998.

[5] K. Montemurro and D. Brentari, "Emphatic fingerspelling as code-mixing in American Sign Language," in *Proceedings of the Linguistic Society of America*, 2018.

[6] N. Pugeault and R. Bowden, "Spelling it out: Real-time ASL fingerspelling recognition," in *Proceedings of the 1st IEEE Workshop on Consumer Depth Cameras for Computer Vision, jointly with ICCV*, 2011.

[7] J. Keane, *Towards an articulatory model of handshape: What fingerspelling tells us about the phonetics and phonology of handshape in American Sign Language*, Ph.D. thesis, University of Chicago, 2014.

[8] B. Shi and K. Livescu, "Multitask training with unlabeled data for end-to-end sign language fingerspelling recognition," in *ASRU*, 2017.

[9] U. von Agris and K.-F. Kraiss, "Towards a video corpus for signer-independent continuous sign language recognition," in *Gesture in Human-Computer Interaction and Simulation. International Gesture Workshop*, 2007.

[10] V. Athitsos, C. Neidle, S. Sclaroff, J. Nash, A. Stefan, A. Thangali, H. Wang, and Q. Yuan, "Large lexicon project: American Sign Language video corpus and sign language indexing/retrieval algorithms," *Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, 2010.

[11] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation," in *AAAI*, 2018.

[12] F. Jens, S. Christoph, H. Thomas, K. Oscar, Z. Uwe, P. Justus, and N. Hermann, "RWTH-PHOENIX-Weather: A large vocabulary sign language recognition and translation corpus," *Language Resources and Evaluation*, pp. 3785–3789, 2012.

[13] T. Kim, J. Keane, W. Wang, H. Tang, J. Riggle, G. Shakhnarovich, D. Brentari, and K. Livescu, "Lexicon-free fingerspelling recognition from video: Data, models, and signer adaptation," *Computer Speech and Language*, pp. 209–232, November 2017.

[14] "National Center for Sign Language and Gesture Resources (NCSLGR) Corpus," `http://www.bu.edu/asllrp/ncslgr-for-download/download-info.html`.

[15] C. Neidle, S. Sclaroff, and V. Athitsos, "SignStream: A tool for linguistic and computer vision research on visual-gestural language data," *Behavior Research Methods, Instruments, & Computers*, vol. 33, no. 3, pp. 311–320, 2001.

[16] N. Gkigkelos and C. Goumopoulos, "Greek sign language vocabulary recognition using Kinect," in *PCI*, 2017.

[17] O. Koller, H. Ney, and R. Bowden, "Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled," in *CVPR*, 2016.

[18] O. Koller, S. Zargaran, and H. Ney, "Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent cnn-hmms," in *CVPR*, 2017.

[19] O. Koller, S. Zargaran, and H. Ney, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization," in *CVPR*, 2017.

[20] A. Katsamanis S. Theodorakis and P. Maragos, "Product-HMMs for automatic sign language recognition," in *ICASSP*, 2009.

[21] T. Kim, G. Shakhnarovich, and K. Livescu, "Fingerspelling recognition with semi-Markov conditional random fields," in *ICCV*, 2013.

[22] T. Kim, W. Wang, H. Tang, and K. Livescu, "Signer-independent fingerspelling recognition with deep neural network adaptation," in *ICASSP*, 2016.

[23] P. Goh and E.-J. Holden, "Dynamic fingerspelling recognition using geometric and motion features," in *ICIP*, 2006.

[24] S. Liwicki and M. Everingham, "Automatic recognition of fingerspelled words in British Sign Language," in *2nd IEEE workshop on CVPR for Human Communicative Behavior Analysis*, 2009.

[25] S. Ricco and C. Tomasi, "Fingerspelling recognition through classification of letter-to-letter transitions," in *ACCV*, 2009.

[26] "Elan (version 5.2)," https://tla.mpi.nl/tools/tla-tools/elan/.

[27] P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Learning to track for spatio-temporal action localization," in *ICCV*, 2015.

[28] B. K. P. Horn and B. G. Schunck, "Determining optical flow," Tech. Rep., Massachusetts Institute of Technology, Cambridge, MA, USA, 1980.

[29] G. Farneback, "Two-frame motion estimation based on polynomial expansion," in *Proceedings of the 13th Scandinavian Conference on Image Analysis*, 2003.

[30] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[31] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "ImageNet: A large-scale hierarchical image database," in *CVPR*, 2009.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[33] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014.

[34] R. Girshick, "Fast r-cnn," in *ICCV*, 2015.

[35] G. Gkioxari and J. Malik, "Finding action tubes," in *CVPR*, 2015.

[36] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. IT-13, pp. 260 – 269, 04 1967.

[37] S. Hochreiter and J. Schmidhüber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[38] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *ICML*, 2006.

[39] O. Vinyals, L. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language," in *NIPS*, 2015, pp. 2773–2781.

[40] K. Alex, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *NIPS*, 2012.

[41] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in *CVPR*, 2015.

[42] "The CMU pronouncing dictionary," http://www.speech.cs.cmu.edu/cgi-bin/cmudict.