Fingerspelling recognition in the wild with iterative visual attention

Bowen Shi[1], Aurora Martinez Del Rio[2], Jonathan Keane[2], Diane Brentari[2]
Greg Shakhnarovich[1], Karen Livescu[1]
[1]Toyota Technological Institute at Chicago, USA [2]University of Chicago, USA
{bshi,greg,klivescu}@ttic.edu {amartinezdelrio,jonkeane,dbrentari}@uchicago.edu

Abstract

Sign language recognition is a challenging gesture sequence recognition problem, characterized by quick and highly coarticulated motion. In this paper we focus on recognition of fingerspelling sequences in American Sign Language (ASL) videos collected in the wild, mainly from YouTube and Deaf social media. Most previous work on sign language recognition has focused on controlled settings where the data is recorded in a studio environment and the number of signers is limited. Our work aims to address the challenges of real-life data, reducing the need for detection or segmentation modules commonly used in this domain. We propose an end-to-end model based on an iterative attention mechanism, without explicit hand detection or segmentation. Our approach dynamically focuses on increasingly high-resolution regions of interest. It outperforms prior work by a large margin. We also introduce a newly collected data set of crowdsourced annotations of fingerspelling in the wild, and show that performance can be further improved with this additional data set.

1. Introduction

Automatic recognition of sign language has the potential to overcome communication barriers for deaf individuals. With the increased use of online media, sign language video-based web sites (e.g., `deafvideo.tv`) are increasingly used as a platform for communication and media creation. Sign language recognition could also enable web services like content search and retrieval in such media.

From a computer vision perspective, sign language recognition is a complex gesture recognition problem, involving quick and fine-grained motion, especially in realistic visual conditions. It is also relatively understudied, with little existing data in natural day-to-day conditions.

In this paper, we study the problem of American Sign Language (ASL) fingerspelling recognition from naturally occurring sign language videos collected from web sites. Fingerspelling is a component of ASL in which words are signed letter by letter, using an alphabet of canonical letter handshapes (Figure 2). Words are fingerspelled mainly (but not only) when they do not have their own ASL signs, for example technical items or proper nouns. Fingerspelling accounts for up to 35% [29] of ASL and is used frequently for content words in social interaction or conversations involving current events or technical topics. In Deaf online media, fingerspelling recognition is crucial as there is often a high proportion of such content words. Fingerspelling recognition is in some ways simpler than general sign language recognition. In ASL, fingerspelled signs are usually one-handed, and the hand remains in a similar position throughout a fingerspelled sequence. However, the task is challenging in other ways, due to the quick, highly coarticulated, and often hard-to-distinguish finger motions, as well as motion blur in lower-quality video "in the wild" (Figures 1, 4).

Automatic sign language recognition is commonly addressed with approaches borrowed from computer vision

Figure 1: Fingerspelling images in studio data vs. in the wild. Leftmost: example frame from the ChicagoFSVid studio data set [17]. Rest: Example frames from the ChicagoFSWild data set [33] (see Section 4).

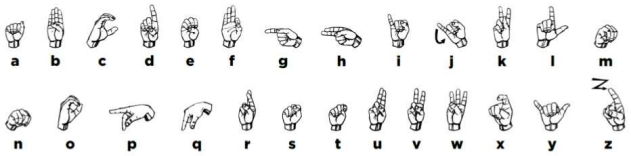

Figure 2: The ASL fingerspelling alphabet, from [16].

and speech recognition. The "front end" usually consists of hand detection [33, 15] or segmentation [18], followed by visual feature extraction. Features are then passed through a sequence model, similar to ones used in speech recognition [6, 17, 20]. Hand detection is a typical first step, as sign language often involves long sequences of large image frames. For example, in a recently introduced fingerspelling data set [33], a substantial proportion of sequences are more than 100 frames long, with an average frame size of 720×480, while the informative region is on average only about 10%. An end-to-end recognition model on raw image frames may have prohibitive memory requirements.

Most prior work on sign language recognition has focused on data collected in a controlled environment. Figure 1 shows example images of fingerspelling data collected "in the wild" in comparison to a studio environment. Compared to studio data, naturally occurring fingerspelling images often involve more complex visual context and more motion blur, especially in the signing hand regions. Thus hand detection, an essential pre-processing step in the typical recognition pipeline, becomes more challenging.

We propose an approach for fingerspelling recognition that does not rely on hand detection. Ours is an attention-based fingerspelling recognition model, trained end-to-end from raw image frames. We make two main contributions: (1) We propose *iterative attention*, an approach for obtaining regions of interest of high resolution with limited computation (see Figure 3). Our model trained with iterative attention achieves higher accuracy than the previous best approach [33], which requires a custom hand detector. We further show that even when a hand or face detector is available, our approach provides significant added value. (2) We introduce a new, publicly available[1] data set of crowd-sourced fingerspelling video annotations, and show that it leads to significantly improved fingerspelling recognition.

[1]https://ttic.edu/livescu/ChicagoFSWild.htm

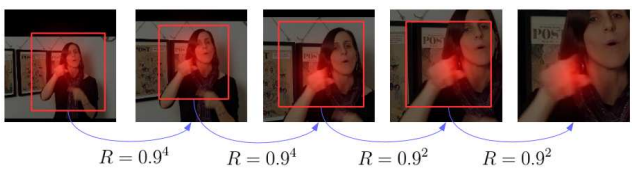

Figure 3: Iterative attention; R=zoom factor.

Figure 4: Ambiguity in fingerspelled handshapes. Top: different letters with similar handshapes, produced by the same signer. Bottom: same letter, different signers.

2. Related Work

Early work on sign language recognition from video[2] mainly focused on isolated signs [7, 2]. More recent work has focused on continuous sign language recognition and data sets [9, 10, 18, 17]. Specifically for fingerspelling, the ChicagoFSVid data set includes 2400 fingerspelling sequences from 4 native ASL signers. The RWTH-PHOENIX-Weather Corpus [10] is a realistic data set of German Sign Language, consisting of sign language videos from 190 television weather forecasts. However, its visual variability is still fairly controlled (e.g. uniform background) and it contains a small number of signers (9) signing in a fairly formal style appropriate for weather broadcasts. In this paper we consider a constrained task (fingerspelling recognition), but with looser visual and stylistic constraints than in most previous work. The recently introduced Chicago Fingerspelling in the Wild (ChicagoFSWild) data set [33] consists of 7304 fingerspelling sequences from online videos. This data set includes a large number of signers (168) and a wide variety of challenging visual conditions, and we use it as one of our test beds.

Automatic sign language recognition approaches often combine ideas from computer vision and speech recognition. A variety of sign language-specific visual preprocessing and features have been proposed in prior work, including ones based on estimated position and movement of body parts (e.g. hand, head) combined with appearance descriptors [3, 44]. Recent work has had success with convolutional neural network (CNN)-based features [20, 21, 23, 22, 4, 32, 33]. Much previous work on sign language recognition, and the vast majority of previous work on fingerspelling recognition, uses some form of hand detection or segmentation to localize the region(s) of interest as an initial step. Kim *et al.* [18, 19, 17] estimate a signer-dependent skin color model using manually annotated hand regions for fingerspelling recognition. Huang *et al.* [15] learn a hand detector based on Faster R-CNN [31] using

[2]There has also been work on sign language recognition using other modalities such as depth sensors (e.g., [30, 15]). Here we consider video-only input, as it is more abundant in naturally occurring online data.

manually annotated signing hand bounding boxes, and apply it to general sign language recognition. Some sign language recognition approaches use no hand or pose preprocessing as a separate step (e.g., [22]), and indeed many signs involve large motions that do not require fine-grained gesture understanding. However, for fingerspelling recognition it is particularly important to understand fine-grained distinctions in handshape. Shi *et al.* [32] find that a custom-trained *signing* hand detector for fingerspelling recognition, which avoids detecting the non-signing hand during fingerspelling, vastly improves performance over a model based on the whole image. This distinction motivates our work on iterative visual attention for zooming in on the relevant regions, without requiring a dedicated hand detector.

Once visual features are extracted, they are typically fed into sequence models such as hidden Markov models [18, 20, 21], segmental conditional random fields [19, 17], or recurrent neural networks (RNNs) [32, 4, 15]. In this paper, we focus on sequential models combining convolutional and recurrent neural layers due to their simplicity and recent success for fingerspelling recognition [32, 33].

There has been extensive work on articulated hand pose estimation, and some models (e.g., [34]) have shown real-life applicability. However, directly applying hand pose estimation to real-life fingerspelling data is challenging. Fingerspelling consists of quick, fine-grained movements, often complicated by occlusion, often at low frame rates and resolutions. We find available off-the shelf pose estimation methods too brittle to work on our data (examples of typical failures included in the supplementary material). An additional challenge to using estimated handshapes as the basis of fingerspelling recognition is the typically large discrepancies between canonical handshapes and actual articulation of hands in continuous signing in real-world settings.

Other related tasks include gesture recognition and action recognition. Gesture recognition is related to isolated sign recognition, which can be understood as classification of handshapes/trajectories from a sequence of frames. Most recent work [28, 27, 26] on gesture recognition relies on depth images and typically also involves hand segmentation as a pre-processing step. On the other hand, action recognition [37, 35, 38] is focused on classification of general video scenes based on visual appearance and dynamics. While our task can be viewed as an example of recognizing a sequence of gestures or actions, sign language (especially fingerspelling) recognition involves discriminating fine-grained handshapes and trajectories from relatively small image regions, further motivating our approach for zooming in on relevant regions.

Spatial attention has been applied in vision tasks including image captioning [42] and fine-grained image recognition [40, 41, 45, 25]. Our use of attention to iteratively zoom in on regions of interest is most similar to the work of Fu *et al.* [11] using a similar "zoom-in" attention for image classification. Their model is trained directly from the full image, and iterative localization provides small gains; their approach is also limited to a single image. In contrast, our model is applied to a frame sequence, producing an "attention tube", and is iteratively trained with frame sequences of increasing resolution, yielding sizable benefits.

The most closely related work to ours is that of Shi *et al.* [32], which first addressed fingerseplling recognition in the wild. In contrast to this prior work, we propose an end-to-end approach that directly transcribes a sequence of image frames into letter sequences, without a dedicated hand detection step. To our knowledge this is the first attempt to address the continuous fingerspelling recognition problem in challenging visual conditions, without relying on hand detection. Our other main contribution is the first successful, large-scale effort to crowdsource sign language annotation, which significantly increases the amount of training data and leads to a large improvement in accuracy.

3. Task and model

The fingerspelling recognition task takes as input a sequence of image frames (or patches) $\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_T$ and produces as output a sequence of letters $w = w_1, w_2, ..., w_K$, $K \leq T$. Note that there is no alignment between the input and output, and typically K is several times smaller than T as there are several frames per letter. We consider the lexicon-free setting, that is we do not assume a closed dictionary of allowed fingerspelled words, since fingerspelled sequences are often ones that do not occur in typical dictionaries. Our approach for fingerspelling recognition includes the attention-based sequence model and the iterative attention approach, each described below.

3.1. Attention-based recurrent neural network

The attention-based recurrent neural network transcribes the input image sequence $\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_T$ into a letter sequence $w = w_1, w_2, ..., w_K$. One option is to extract visual features with a 2D-CNN on individual frames and feed those features to a recurrent neural network to incorporate temporal structure. Alternatively, one can obtain a spatio-temporal representation of the frame sequence by applying a 3D-CNN to the stacked frames. Both approaches lack *attention* – a mechanism to focus on the informative part of an image. In our case, most information is conveyed by the hand, which often occupies only a small portion of each frame. This suggests using a spatial attention mechanism.

Our attention model is based on a convolutional recurrent architecture (see Figure 5). At frame t, a fully convolutional neural network is applied on the image frame $\mathbf{I}_t$ to extract a feature map $\mathbf{f}_t$. Suppose the hidden state of the recurrent unit at timestep $t-1$ is $\mathbf{e}_{t-1}$. We compute the attention map $\boldsymbol{\beta}_t$ based on $\mathbf{f}_t$ and $\mathbf{e}_{t-1}$ (where i, j index spatial locations):

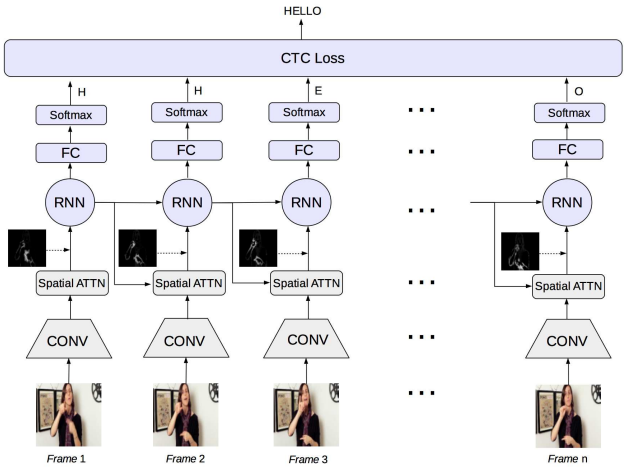

Figure 5: Recurrent CNN with attention.

$$v_{tij} = \mathbf{u}_f^T \tanh\left(\mathbf{W}_d \mathbf{e}_{t-1} + \mathbf{W}_f \mathbf{f}_{tij}\right) \quad \beta_{tij} = \frac{\exp\left(v_{tij}\right)}{\sum_{i,j} \exp\left(v_{tij}\right)}$$

The attention map $\boldsymbol{\beta}_t$ reflects the importance of features at different spatial locations to the letter sequence. Optionally, we include a *prior-based attention* term $\mathbf{M}$, which represents prior knowledge we have about the importance of different spatial locations for our task. For instance, $\mathbf{M}$ may be based on optical flow, as regions in motion are more likely than static regions to correspond to a signing hand. The visual feature vector at time step t is a weighted average of $\mathbf{f}_{tij}$, $1 \leq i \leq h, 1 \leq j \leq w$, where w and h are the width and height of the feature map respectively:

$$\mathbf{A}_t = \frac{\boldsymbol{\beta}_t \odot \mathbf{M}_t^\alpha}{\sum_{p,q} \beta_{tpq} M_{tpq}^\alpha}, \qquad \mathbf{h}_t = \sum_{i,j} \mathbf{f}_{tij} A_{tij} \tag{1}$$

where $\mathbf{A}$ represents the (posterior) attention map and α controls the relative weight of the prior and attention weights learned by the model. The state of the recurrent unit at time step t is updated via $\mathbf{e}_t = LSTM(\mathbf{e}_{t-1}, \mathbf{h}^t)$ where $LSTM$ refers to a long short-term memory unit [14] (though other types of RNNs could be used here as well).

The sequence $\mathbf{E} = (\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_T)$ can be viewed as high-level features for the image frames. Once we have this sequence of features, the next step is to decode it into a letter sequence: $(\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_T) \rightarrow w = w_1, w_2, ..., w_K$. Our model is based on connectionist temporal classification (CTC) [13], which requires no frame-to-letter alignment for training. For a sequence of visual features $\mathbf{e}$ of length T, we generate frame-level label posteriors via a fully-connected layer followed by a softmax, as shown in Figure 5. In CTC, the frame-level labels are drawn from $L \cup \{blank\}$, where L is the true label set and $blank$ is a special label that can be interpreted as "none of the above". The probability of a complete frame-level labeling $\pi = (\pi_1, \pi_2, \ldots, \pi_T)$ is then

$$p(\pi|\mathbf{e}_{1:T}) = \prod_{t=1}^{T} \underset{\pi_t}{\text{softmax}}(\mathbf{W}^e \mathbf{e}_t + \mathbf{b}^e) \tag{2}$$

At test time, we can produce a final frame-level label sequence by taking the highest-probability label at each frame (greedy search). Finally, the label sequence $w = w_1, w_2, ..., w_K$ is produced from the frame-level sequence π via the CTC "label collapsing function" $\mathcal{B}$, which removes duplicate frame labels and then $blank$s.

At training time CTC maximizes log probability of the final label sequence, by summing over all compatible frame-level labelings using a forward-backward algorithm.

Language model In addition to this basic model, at test time we can also incorporate a language model providing a probability for each possible next letter given the previous ones. We use a beam search to find the best letter sequence, in a similar way to decoding approaches used for speech recognition: The score for hypotheses in the beam is composed of the CTC model's score (softmax output) combined with a weighted language model probability, and an additional bias term for balancing insertions and deletions.

3.2. Iterative visual attention via zooming in

The signing hand(s) typically constitute only a small portion of each frame. In order to recognize fingerspelling sequences, the model needs to be able to reason about fine-grained motions and minor differences in handshape. The attention mechanism enables the model to focus on informative regions, but high resolution is needed in order to retain sufficient information in the attended region. One straightforward approach is to use very high-resolution input images. However, since the convolutional recurrent encoder covers the full image sequence, using large images can lead to prohibitively large memory footprints. Using the entire frame in real-world videos also increases vulnerability to distractors/noise.

To get the benefit of high resolution without the extra computational burden, we propose to iteratively focus on regions within the input image frames, by refining the attention map. Given a trained attention model $\mathcal{H}$, we run inference with $\mathcal{H}$ on the target image sequence $\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_T$ to generate the associated sequence of posterior attention maps: $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_T$. We use the sequence of attention maps to obtain a new sequence of images $\mathbf{I}'_1, \mathbf{I}'_2, \ldots, \mathbf{I}'_T$ consisting of smaller bounding boxes within the original images. The fact that we extract the new frames by *zooming in on the original images* is key, since this allows us to retain the highest resolution available in the original video, while restricting our attention and without paying the price for using large high resolution frames.

We then train a new model $\mathcal{H}'$ that takes $\mathbf{I}'_1, \mathbf{I}'_2, \ldots, \mathbf{I}'_T$ as input. We can iterate this process, finding increasingly

smaller regions of interest (ROIs).This iterative process runs for S steps (producing S trained models) until ROI images of sufficiently high resolution are obtained. In practice, the stopping criterion for iterative attention is based on fingerspelling accuracy on held-out data.

Given a series of zooming ratios (ratios between the size of the bounding box and the full frame) $R_1, R_2..., R_S$, the zooming process sequentially finds a series of bounding box sequences $\{b_t^1\}_{1\le t\le T}, ..., \{b_t^S\}_{1\le t\le T}$. We describe in the experiments section how we choose Rs.

This iterative process generates S models. At test time, for each input image sequence $\mathbf{I}_1, \mathbf{I}_2, ..., \mathbf{I}_T$, the models $\mathcal{H}_{1:S-1}$ are run in sequence to get a sub-region sequence $\mathbf{I}_1^{S-1}, \mathbf{I}_2^{S-1}, ..., \mathbf{I}_T^{S-1}$. For simplicity we just use the last model $\mathcal{H}_S$ for word decoding based on input $\mathbf{I}_1^{S-1}, \mathbf{I}_2^{S-1}, ..., \mathbf{I}_T^{S-1}$. The iterative attention process is illustrated in Algorithm 1 and Figures 3, 6.

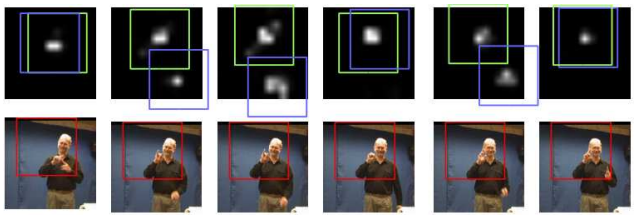

Figure 6: Illustration of one iteration of iterative attention, consisting of finding a zoomed-in ROI sequence based on the sequence of visual attention maps. 1st row: sequence of attention maps overlaid by candidate boxes of every frame. Green boxes are selected by dynamic programming. 2nd row: final sequence of bounding boxes after averaging.

We next detail how bounding boxes are obtained in each iteration of iterative attention (illustrated in Figure 6). In each iteration s, the goal is to find a sequence of bounding boxes $\{b_1, b_2, ..., b_T\}$ based on the posterior attention map sequence $\{\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_T\}$ and the zoom factor R, which determines the size of b_i relative to the size of $\mathbf{I}^s$. In each frame $\mathbf{I}_t^s$, we put a box of size $R_s|\mathbf{I}^s|$ centered at each of the top k peaks in the attention map $\mathbf{A}_t$. Each box b_t^i, $t \in [T]$, $i \in [k]$, is assigned a score a_t^i equal to the attention value at its center. We define a linking score between two bounding boxes b_t^i in consecutive frames as follows:

$$sc(b_t^i, b_{t+1}^j) = a_t^i + a_{t+1}^j + \lambda * IoU(b_t^i, b_{t+1}^j), \quad (3)$$

where $IoU(b_t^i, b_{t+1}^j)$ is the Jaccard index (intersection over union) of b_t^i and b_{t+1}^j and λ is a hyperparameter that trades off between the box score and smoothness. Using IoU has a smoothing effect and ensures that the framewise bounding box does not switch between hands. This formulation is analogous to finding an "action tube" in action recognition [12]. Finding the sequence of bounding boxes with highest average s can be written as the optimization problem

Algorithm 1 Iterative attention via zooming in.

Training, Input: $\{(\mathbf{I}_{1:T_n}^{n,0}, w^n)\}_{1\le n\le N}$

for $s \in \{1, 2, ..., S\}$ **do**
  Train model $\mathcal{H}_s$ with $(\mathbf{I}_{1:T_n}^{n,s-1}, w^n)_{1\le n\le N}$
  for $n = 1, ...N$ **do**
    Run inference on $\mathbf{I}_{1:T_n}^n$ with $\mathcal{H}_s$ to obtain attention map $\mathbf{A}_{1:T_n}^n$
    Solve Equation 4 to obtain sequence of bounding boxes $b_{1:T_n}^n$
    Crop and resize $\mathbf{I}_{1:T_n}^{n,0}$ with $b_{1:T_n}^n$ to get $\mathbf{I}_{1:T_n}^{n,s}$
Return $\{\mathcal{H}_s\}, 1 \le s \le S$

Test, Input: $\mathbf{I}_{1:T}^0$, $\{\mathcal{H}_s\}, 1 \le s \le S$

for $s \in \{1, 2, ..., S\}$ **do**
  Run inference on $\mathbf{I}_{1:T}^{s-1}$ with $\mathcal{H}_s$ to obtain attention map $\mathbf{A}_{1:T}$ and predicted words w^s
  Solve Equation 4 to obtain sequence of bounding boxes $b_{1:T}$
  Crop and resize $\mathbf{I}_{1:T}^0$ with $b_{1:T}$ to get $\mathbf{I}_{1:T}^s$
Return $\mathbf{w}^S$

$$\arg\max_{i_1,...,i_T} \frac{1}{T}\sum_{t=1}^{T-1} sc(b_t^{i_t}, b_{t+1}^{i_{t+1}}) \quad (4)$$

which can be efficiently solved by dynamic programming. Once the zooming boxes are found, we take the average of all boxes within a sequence for further smoothing, and finally crop the zoom-in region from the original (full resolution) image frames to avoid any repetitive interpolation artifacts from unnecessary resizing. We describe our process for determining the zoom ratios $R_{1:S}$ in Section 5.3.

4. Data and crowdsourced annotations

We use two data sets: Chicago Fingerspelling in the Wild (ChicagoFSWild) [33], which was carefully annotated by experts; and a crowdsourced data set we introduce here, ChicagoFSWild+. Both contain clips of fingerspelling sequences excised from sign language video "in the wild", collected from online sources such as YouTube and `deafvideo.tv`. ChicagoFSWild contains 5455 training sequences from 87 signers, 981 development (validation) sequences from 37 signers, and 868 test sequences from 36 signers, with no overlap in signers in the three sets.

We developed a fingerspelling video annotation interface derived from VATIC [39] and have used it to collect our new data set, ChicagoFSWild+, by crowdsourcing the annotation process via Amazon Mechanical Turk. Annotators are presented with one-minute ASL video clips and are asked to mark the start and end frames of fingerspelling within the clips (if any is present), to provide a transcription (a sequence of English letters) for each fingerspelling sequence,

but not to align the transcribed letters to video frames. Two annotators are used for each clip. The videos in ChicagoFSWild+ include varied viewpoints and styles (Figure 1).

Gender (%)	female	32.7	male	63.2	other	4.1
Handedness (%)	left	10.6	right	86.9	other	2.5

Table 1: Statistics of ChicagoFSWild+ (train+test+dev). "Other" includes multiple signers, unknown, etc.

ChicagoFSWild+ includes 50,402 training sequences from 216 signers (Table 1), 3115 development sequences from 22 signers, and 1715 test sequences from 22 signers. This data split has been done in such a way as to approximately evenly distribute certain attributes (such as signer gender and handedness) between the three sets. In addition, in order to enable clean comparisons between results on ChicagoFSWild and ChicagoFSWild+, we used the signer labels in the two data sets to ensure that there are no overlaps in signers between the ChicagoFSWild training set and the ChicagoFSWild+ test set. Finally, the annotations in the development and test sets were proofread and a single "clean" annotation kept for each sequence in these sets. For the training set, no proofreading has been done and both annotations of each sequence are used. Compared to ChicagoFSWild, the crowdsourcing setup allows us to collect dramatically more training data in ChicagoFSWild+, with significantly less expert/researcher effort.

5. Experiments

We report results on three evaluation sets: *ChicagoFSWild/dev* is used to initially assess various methods and select the most promising ones; *ChicagoFSWild/test* results are directly comparable to prior work [33]; and finally, results on our new *ChicagoFSWild+/test* set provide an additional measure of accuracy in the wild on a set more than twice the size of *ChicagoFSWild/test*. These are the only existing data sets for sign language recognition "in the wild" to our knowledge. Performance is measured in terms of letter accuracy (in percent), computed by finding the minimum edit (Hamming) distance alignment between the hypothesized and ground-truth letter sequences. The letter accuracy is defined as $1 - \frac{S+D+I}{N}$, where S, D, I are the numbers of substitutions, insertions, and deletions in the alignments and N is the number of ground-truth letters.

5.1. Initial frame processing

We consider the following scenarios for initial processing of the input frames:

Whole frame Use the full video frame, with no cropping.

Face ROI Crop a region centered on the face detection box, but 3 times larger.

Face scale Use the face detector, but instead of cropping, resize the entire frame to bring the face box to a canonical size (36 pixels).

Hand ROI Crop a region centered on the box resulting from the signing hand detector, either the same size as the bounding box or twice larger (this choice is a tuning parameter).

Method	Letter accuracy (%)
Whole frame	11.0
Whole frame+attention	23.0
Ours+whole frame	42.8
Ours+whole frame+LM	43.6
Face scale	10.9
Face scale+attention	14.2
Ours+face scale	42.9
Ours+face scale+LM	44.0
Face ROI	27.8
Face ROI+attention	33.4
Face ROI+attention+LM	35.2
Ours+face ROI	45.6
Ours+face ROI+LM	**46.8**
Hand ROI [33]	41.1
Hand ROI+LM [33]	42.8
Hand ROI+attention	41.4
Hand ROI+attention+LM	43.1
Ours+hand ROI	45.0
Ours+hand ROI+LM	45.9

Table 2: Results on *ChicagoFSWild/dev*; training on *ChicagoFSWild/train*. Ours+X: iterative attention (proposed method) applied to input obtained with X. +LM: add language model trained on *ChicagoFSWild/train*.

5.2. Model variants

Given any initial frame processing **X** from the list above, we compare several types of models:

X Use the sequence of frames/regions in a recurrent convolutional CTC model directly (as in Figure 5, but without visual attention). For **X = Hand ROI**, this is the approach used in [33], the only prior work on the task of open-vocabulary fingerspelling recognition in the wild.

X+attention Use the model of Figure 5.

Ours+X Apply our iterative attention approach starting with the input produced by **X**.

The architecture of the recognition model described in Sec. 3.1 is the same in all of the approaches, except for the choice of visual attention model. All input frames (cropped or whole) are resized to a max size of 224 pixels, except for **Face scale** which yields arbitrary sized frames. Images of higher resolution are not used due to memory constraints.

5.3. Implementation Details

We use the signing **hand detector** provided by the authors of [33], and the two-frame motion (**optical flow**) estimation algorithm of Farneback [8].

We use the implementation of [1] for **face detection**, trained on the WIDER data set [43]. To save computation we run the face detector on one in every five frames in each sequence and interpolate to get bounding boxes for the remaining frames. In cases where multiple faces are detected, we form "face tubes" by connecting boxes in subsequent frames with high overlap. Tubes are scored by average optical flow within an (expanded) box along the tube, and the highest scoring tube is selected. Bounding box positions along the tube are averaged, producing the final set of face detection boxes for the sequence. See supplementary material for additional details.

Model training The convolutional layers of our model are based on AlexNet [24][3] pre-trained on ImageNet [5]. The last max-pooling layer of AlexNet is removed so that we have a sufficiently large feature map. When the input images are of size 224×224, the extracted feature map is of size 13×13; larger inputs yield larger feature maps. We include 2D-dropout layers between the last three convolutional layers with drop rate 0.2. For the RNN, we use a one-layer LSTM network with 512 hidden units. The model is trained with SGD, with an initial learning rate of 0.01 for 20 epochs and 0.001 for an additional 10 epochs. We use development set accuracy for early stopping. We average optical flow images at timestep $t-1$, t and $t+1$, and use the magnitude as the prior map $\mathbf{M}_t$ (Equation 1) for time step t. The language model is an LSTM with 256 hidden units, trained on the training set annotations.[4] Experiments are run on an NVIDIA Tesla K40c GPU.

Zoom ratios For each iteration of the iterative visual attention, we consider zoom ratios $R \in \{0.9, 0.9^2, 0.9^3, 0.9^4\}$, and find the optimal sequence of ratios by beam search, with beam size 2, using accuracy on *ChicagoFSWild+/dev* as the evaluation criterion. The parameter λ in Equation (3) is tuned to 0.1.

5.4. Results

Results on dev Table 2 shows results on *ChicagoFSWild/dev* for models trained on *ChicagoFSWild/train*.

First, for all types of initial frame processing, performance is improved by introducing a standard visual attention mechanism. The whole-frame approach (whether scaled by considering face detections, or not) is improved the most, since without attention too much of model capacity is wasted on irrelevant regions; however, attention applied to whole frames remains inferior to ROI-based methods. Using the pre-trained hand or face detector to guide the ROI extraction produces a large boost in accuracy, confirming that focusing the model on a high-resolution, task-relevant ROI is important. These ROI-based methods still benefit from adding standard attention, but the improvements are smaller (face ROI: 5.6%, hand ROI: 0.3%).

In contrast, our iterative attention approach, which does not rely on any pretrained detectors, gets better performance than detector-based methods, including the approach of [33] (**Hand ROI**), even when attention is added to the latter (42.8% for **Ours+whole frame** vs. 41.4% for **Hand ROI+attention**). Our approach of (gradually) zooming in on an ROI therefore outperforms a signing hand detector. Specifically in **Hand ROI**, the improvement suggests signing hands can get more precisely located with our approach after initialization from a hand detector.

Finally, adding a language model yields modest accuracy improvements across the board. The language model has a development set perplexity of 17.3, which is quite high but still much lower than the maximum possible perplexity (the number of output labels). Both the high perplexity and small improvement from the language model are unsurprising, since fingerspelling is often used for rare words.

Method	ChicagoFSWild	ChicagoFSWild+
Hand ROI+LM [33]	41.9	41.2
+new data	57.5	58.3
Ours+whole frame+LM	42.4	43.8
+new data	57.6	61.0
Ours+hand ROI+LM	42.3	45.9
+new data	60.2	61.1
Ours+face ROI+LM	**45.1**	**46.7**
+new data	**61.2**	**62.3**

Table 3: Results on *ChicagoFSWild/test* and *ChicagoFSWild+/test*. Black: trained on *ChicagoFSWild/train*; Green: trained on *ChicagoFSWild/train* + *ChicagoFSWild+/train*.

Results on test We report results on *ChicagoFSWild/test* for the methods that are most competitive on dev (Table 3). All of these use some form of attention (standard or our iterative approach) and a language model. We again note that this table includes comparison to the only prior work applicable to this task known to us [33].

The combination of face-based initial ROI with our iterative attention zooming produces the best results overall. This is likely due to the complexity of our image data. In cases of multiple moving objects in the same image, the zooming-in process may fail especially in initial iterations of whole frame-based processing, when the resolution of the hand is very low because of downsampling given memory constraints. On the other hand, the initial face-based ROI cropping is likely to remove clutter and distractors without loss of task-relevant information. However, even without cropping to the face-based ROI, our ap-

[3] We do not use a deeper network like VGG [36]), as the memory requirements are prohibitive due to its depth/stride combination when working on entire video sequences. Experiments with a relatively shallow ResNet-18 showed no improvement over the AlexNet backbone.

[4] Training on external English text is not appropriate here, since the distribution of fingerspelled words is quite different from that of English.

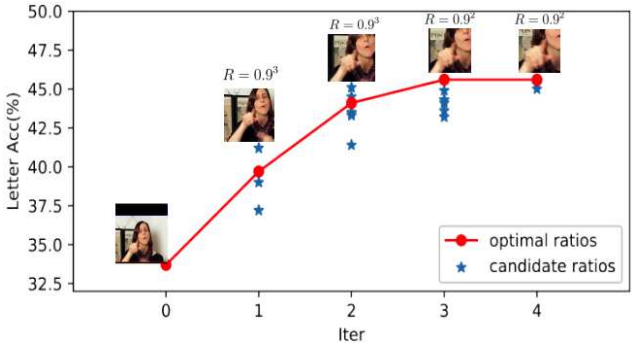

Figure 7: Letter accuracy vs. iteration in the **Ours+face ROI** setting, showing an example ROI zooming ratio sequence found by beam search (shown, red curve). Blue stars: accuracy with other zooming ratios considered.

proach (**Ours+whole frame+LM**) still improves over the hand detection-based one of [33].

Training on additional data Finally, we report the effect of extending the training data with the *ChicagoFSWild+/train* set, increasing the number of training sequences from 5,455 to 55,856. The crowdsourced annotations in ChicagoFSWild+ may be noisier, but they are much more plentiful. In addition, the crowdsourced training data includes two annotations of each sequence, which can be seen as a form of natural data augmentation. As Table 3 shows (in green), all tested models benefit significantly from the new data. But the large gap between our iterative attention approach and the hand detector-based approach of [33] remains. The improvement of our approach over [33] applied to whole frames is *larger* on the ChicagoFSWild+ test set. The hand detector could become less accurate due to possible domain discrepancy between ChicagoFSWild (on which it was trained) and ChicagoFSWild+. In contrast, our model replacing the off-the-shelf hand detector with an iterative attention-based "detector" is not influenced by such a discrepancy.

5.5. Additional analysis

Effect of iterative zooming The results of Table 2 indicate that iterative zooming gives a large performance boost over the basic model. In both face ROI and whole frame setups, the hand corresponds to only a small portion of the input image. Figure 7 shows how the accuracy and the input image evolve in successive zooming iterations. Though no supervision regarding the hand is used for training, the location of the signing hand is implicitly learned through the attention mechanism. Higher recognition accuracy suggests that the learned attention locates the hand more precisely than a separately trained detector. To test this hypothesis, we measure hand detection performance on the dev set from the hand annotation data in ChicagoFSWild. At the same miss rate of 0.158, the average IoU's of the attention-based detector and of the separately trained hand detector are 0.413 and 0.223, respectively. Qualitatively, we also compare the sequences of signing hands output by the two detectors. See supplementary material for more details.

As we zoom in, two things happen: The resolution of the hand increases, and more of the potentially distracting background is removed. One could achieve the former without the latter by *enlarging* the initial input by $1/R$. We compared this approach to iterative attention, and found that (i) it was prohibitively memory-intensive (we could not proceed past one zooming iteration), (ii) it decreased performance, and (iii) the prior on attention became more important (see supplementary material). Therefore, iterative attention allows us to operate at much higher resolution than would have been possible without it, and in addition helps by removing distracting portions of the input frames.

Robustness to face detection accuracy Since our best results are obtained with a face detector-based initial ROI, we investigate the sensitivity of the results to the accuracy of face detection, and we find that recognition performance degrades gracefully with face detector errors. See the supplementary material for details and experiments.

Timing Our best method, **Ours+face ROI+LM**, takes on average 65ms per frame.

Human performance We measured the letter accuracy on *ChicagoFSWild/test* of a native signer and two additional proficient signers. The native signer has an accuracy of 86.1%; the non-native signers have somewhat lower accuracies (74.3%, 83.1%). These results indicate that the task is not trivial even for humans, but there is still much room for improvement from our best machine performance (61.2%).

6. Conclusion

We have developed a new model for ASL fingerspelling recognition in the wild, using an iterative attention mechanism. Our model gradually reduces its area of attention while simultaneously increasing the resolution of its ROI within the input frames, yielding a sequence of models of increasing accuracy. In contrast to prior work, our approach does not rely on any hand detection, segmentation, or pose estimation modules. We also contribute a new data set of fingerspelling in the wild with crowdsourced annotations, which is larger and more diverse than any previously existing data set, and show that training on the new data significantly improves the accuracy of all models tested. The results of our method on both the new data set and an existing benchmark are better than the results of previous methods by a large margin. We expect our iterative attention approach to be applicable to other fine-grained gesture or action sequence recognition tasks.

Acknowledgements This research was supported in part by NSF grant 1433485.

References

[1] The world's simplest facial recognition API for Python and the command line. https://github.com/ageitgey/face_recognition. 7

[2] Vassilis Athitsos, Carol Neidle, Stan Sclaroff, Joan Nash, Alexandra Stefan, Ashwin Thangali, Haijing Wang, and Quan Yuan. Large lexicon project: American Sign Language video corpus and sign language indexing/retrieval algorithms. In *Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies (CSLT)*, 2010. 2

[3] Richard Bowden, David Windridge, Timor Kadir, Andrew Zisserman, and Michael Brady. A linguistic feature vector for visual representation of sign language. In *ECCV*, 2004. 2

[4] Runpeng Cui, Hu Liu, and Changshui Zhang. Recurrent convolutional neural networks for continuous sign language recognition by staged optimization. In *CVPR*, 2017. 2, 3

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 7

[6] Philippe Dreuw, David Rybach, Thomas Deselaers, Morteza Zahedi, and Hermann Ney. Speech recognition techniques for a sign language recognition system. In *Eighth Annual Conference of the International Speech Communication Association*, 2007. 2

[7] Eleni Efthimiou, Stavroula-Evita Fotinea, Thomas Hanke, John R. W. Glauert, Richard Bowden, Annelies Braffort, Christophe Collet, Petros Maragos, and François Lefebvre-Albaret. Sign language technologies and resources of the Dicta-Sign project. In *LREC Workshop on the Representation and Processing of Sign Languages: Interactions between Corpus and Lexicon*, 2012. 2

[8] Gunnar Farneback. Two-frame motion estimation based on polynomial expansion. In *Proceedings of the 13th Scandinavian Conference on Image Analysis*, 2003. 6

[9] Jens Forster, Christoph Schmidt, Thomas Hoyoux, Oscar Koller, Uwe Zelle, Justus Piater, and Hermann Ney. RWTH-PHOENIX-Weather: A large vocabulary sign language recognition and translation corpus. *Language Resources and Evaluation*, pages 3785–3789, 2012. 2

[10] Jens Forster, Christoph Schmidt, Oscar Koller, Martin Bellgardt, and Hermann Ney. Extensions of the sign language recognition and translation corpus RWTH-PHOENIX-Weather. *Computer Vision and Image Understanding*, 141:108–125, 12 2015. 2

[11] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *CVPR*, 2017. 3

[12] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *CVPR*, 2015. 5

[13] Alex Graves, Santiago Fernandez, Faustino. Gomez, and Jürgen. Schmidhuber. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 2006. 4

[14] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. In *Neural Computation*, 1997. 4

[15] Jie Huang, Wengang Zhou, Qilin Zhang, Houqiang Li, and Weiping Li. Video-based sign language recognition without temporal segmentation. In *AAAI*, 2018. 2, 3

[16] Jonathan Keane. *Towards an articulatory model of handshape: What fingerspelling tells us about the phonetics and phonology of handshape in American Sign Language*. PhD thesis, University of Chicago, 2014. 2

[17] Taehwan Kim, Jonathan Keane, Weiran Wang, Hao Tang, Jason Riggle, Gregory Shakhnarovich, Diane Brentari, and Karen Livescu. Lexicon-free fingerspelling recognition from video: Data, models, and signer adaptation. *Computer Speech and Language*, pages 209–232, November 2017. 1, 2, 3

[18] Taehwan Kim, Greg Shakhnarovich, and Karen Livescu. Fingerspelling recognition with semi-Markov conditional random fields. In *ICCV*, 2013. 2, 3

[19] Taehwan Kim, Weiran Wang, Hao Tang, and Karen Livescu. Signer-independent fingerspelling recognition with deep neural network adaptation. In *ICASSP*, 2016. 2, 3

[20] Oscar Koller, Hermann Ney, and Richard. Bowden. Deep hand: How to train a CNN on 1 million hand images when your data is continuous and weakly labelled. In *CVPR*, 2016. 2, 3

[21] Oscar Koller, Sepehr Zargaran, and Hermann Ney. Re-sign: Re-aligned end-to-end sequence modelling with deep recurrent CNN-HMMs. In *CVPR*, 2017. 2, 3

[22] Oscar Koller, Sepehr Zargaran, Hermann Ney, and Richard Bowden. Deep sign: Enabling robust statistical continuous sign language recognition via hybrid CNN-HMMs. *International Journal of Computer Vision*, 126(12):1311–1325, 2018. 2, 3

[23] Oscar Koller, Sepehr Zargaran, Ralf Schlüter, and Richard Bowden. Deep sign: Hybrid CNN-HMM for continuous sign language recognition. In *BMVC*, 2016. 2

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 7

[25] Xiao Liu, Tian Xia, Jiang Wang, and Yuanqing Lin. Fully convolutional attention localization networks: Efficient attention localization for fine-grained recognition. *ArXiv*, abs/1603.06765, 2016. 3

[26] Qiguang Miao, Yunan Li, Wanli Ouyang, Zhenxin Ma1, Xin Xu, Weikang Shi, and Xiaochun Cao. Multimodal gesture recognition based on the resc3d network. In *CVPR*, 2017. 3

[27] Pavlo Molchanov, Xiaodong Yang, Shalini Gupta, Kihwan Kim, Stephen Tyree, and Jan Kautz. Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network. In *CVPR*, 2016. 3

[28] Pradyumna Narayana, J. Ross Beveridge, and Bruce A. Draper. Gesture recognition: Focus on the hands. In *CVPR*, 2018. 3

[29] Carol A. Padden and Darline C. Gunsals. How the alphabet came to be used in a sign language. *Sign Language Studies*, pages 10–33, 4 (1) 2003. 1

[30] Nicolas Pugeault and Richard 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. 2

[31] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[32] Bowen Shi and Karen Livescu. Multitask training with unlabeled data for end-to-end sign language fingerspelling recognition. In *ASRU*, 2017. 2, 3

[33] Bowen Shi, Aurora Martinez Del Rio, Jonathan Keane, Jonathan Michaux, Diane Brentari, Greg Shakhnarovich, and Karen Livescu. American Sign Language fingerspelling recognition in the wild. In *SLT*, 2018. 1, 2, 3, 5, 6, 7, 8

[34] Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017. 3

[35] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 3

[36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 7

[37] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 3

[38] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 3

[39] Carl Vondrick, Donald Patterson, and Deva Ramanan. Efficiently scaling up crowdsourced video annotation. In *IJCV*, 2012. 5

[40] Dequan Wang, Zhiqiang Shen, Jie Shao, Wei Zhang, Xiangyang Xue, and Zheng Zhang. Multiple granularity descriptors for fine-grained categorization. In *ICCV*, 2015. 3

[41] Tianjun Xiao, Yichong Xu, Kuiyuan Yang, Jiaxing Zhang, Yuxin Peng, and Zheng Zhang. The application of two-level attention models in deep convolutional neural network for fine-grained image classification. In *CVPR*, 2015. 3

[42] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 3

[43] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *CVPR*, 2016. 7

[44] Mahmoud M. Zaki and Samir I. Shaheen. Sign language recogntion using a combination of new vision based features. In *Pattern Recognition Letters*, 2011. 2

[45] Bo Zhao, Xiao Wu, Jiashi Feng, Qiang Peng, and Shuicheng Yan. Diversified visual attention networks for fine-grained object classification. *IEEE Transactions on Multimedia*, 19:1245–1256, 2016. 3