GAN Based Indian Sign Language Synthesis

Shyam Krishna*
International Institute of Information Technology, Bangalore
Bengaluru, India
krishnshyam@gmail.com

Janmesh Ukey*
Reliance Jio, AICoE
Hyderabad, India
janmeshukey@gmail.com

Dinesh Babu J
International Institute of Information Technology, Bangalore
Bengaluru, India
jdinesh@iiitb.ac.in

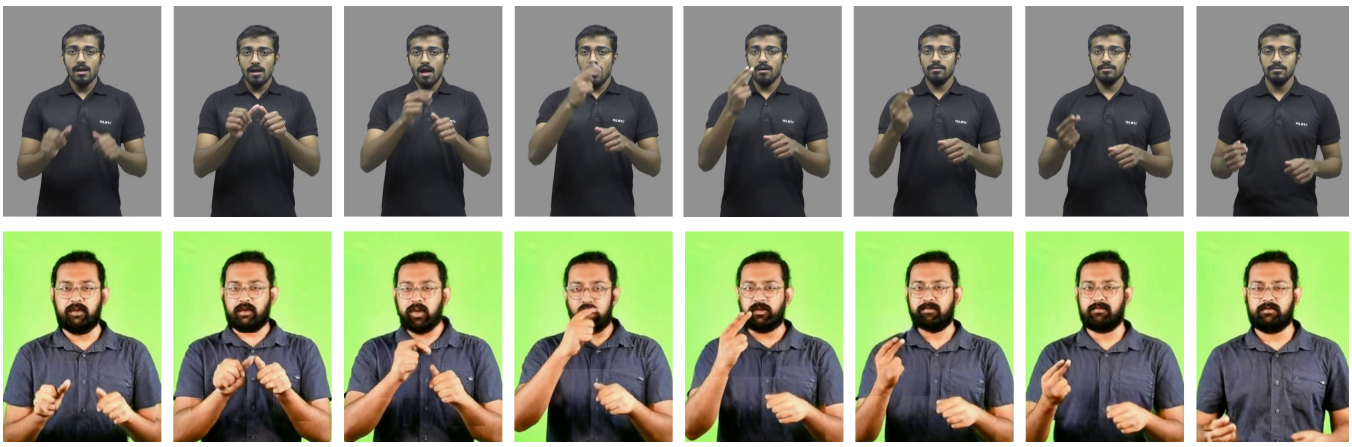

Figure 1: Original Video of Performer Doing the ISL Sign for "Art" (Above) Compared to the Output Generated by Our Model (Below).

ABSTRACT

Controllable visual reproduction of sign language, termed Sign Language Synthesis (SLS), is a major and challenging task in sign language processing. Traditional methods have used computer animation to perform this task, but these have faced several limitations. Animation usually requires expensive equipment to perform motion capture and intensive manual oversight to ensure accuracy. Recently, Generative Adversarial Networks (GANs) have had very promising results in pose and motion transfer. This has been explored as a SLS method in Stoll, et al. [22] and related work. However, this work has required datasets with manual annotation, both in the form of requiring a corpus of manually selected "good" hand poses, as well a large corpus of videos of continuous signing which are annotated for the sequence of signs appearing in them. Most sign languages, however, face a dearth in data, especially annotated data, and this is the case for Indian Sign Language (ISL). In this paper, we present a method for overcoming this issue in the first GAN based SLS model created specifically for ISL. We use a combination of separate generators for the hand and body to overcome the problem of requiring hand-picked "good" hand images from training videos. We further refine the output with another network to remove the artefacts appearing due to combining separate GAN outputs. We also experiment with creating continuous sign language output without requiring an annotated corpus, by stitching together individual signs obtained from a publicly available video lexicon of ISL. We show our model performs competitively in these tasks in both quantitative measures as well as in human perception tests.

CCS CONCEPTS

• **Computing methodologies** → **Natural language generation**; **Image representations**; *Neural networks.*

KEYWORDS

sign language synthesis, video production, generative adversarial networks

ACM Reference Format:
Shyam Krishna, Janmesh Ukey, and Dinesh Babu J. 2021. GAN Based Indian Sign Language Synthesis. In *Proceedings of 12th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP'21),* Chetan Arora, Parag Chaudhuri, and Subhransu Maji (Eds.). ACM, New York, NY, USA, Article 44, 8 pages. https://doi.org/10.1145/3490035.3490301

1 INTRODUCTION

India has a population of 1 to 2.7 million Hearing Impaired (HI) individuals, and Indian Sign Language (ISL) is the main mode of

*Both authors contributed equally to the paper

ICVGIP'21, December 19–22, 2021, Jodhpur, India

ACM ISBN 978-1-4503-7596-2/21/12…$15.00
https://doi.org/10.1145/3490035.3490301

communication among them [7]. This segment of the population is unable to access general video and audio media normally. Reading comprehension in this population is also low, and studies have found that reading comprehension of HI in their late teens is comparable to that of hearing children less than 10 years old [16]. The most appropriate mode of consumption is therefore through interpreters converting audio and video to sign language. However given the dearth of these interpreters in comparison to the vast population, as well as quantity of media, a way of automating this process is necessary. Sign Language Synthesis (SLS) seeks to bridge this gap by providing a way to parametrize and generate sign language without human supervision.

SLS has been a challenging problem in the field of sign language processing. Presence of several thousand unique signs, each having multiple complex channels of output including handshapes, precise points of articulation and varied and semantically meaningful facial expressions, contribute to the difficulty of the task. Traditionally, sign language applications have used computer animations to control avatars and generate sign language output, but this has been by no means a perfect solution. Designing and animating these avatars has been done using expensive motion capture technology. This requires extensive human oversight as the data obtained from the process is noisy and needs to be manually cleaned. Add to this the fact that it is to be done for a vocabulary of thousands of signs, and it becomes clear how resource hungry and impractical the approach is.

In the last decade, Generative Adversarial Networks (GANs) have been greatly successful in several image generation tasks, specifically conditional image generation. One such task which is relevant to our work has been pose transfer, which involves generating images and video of a person in different poses, given the pose as input. The work of Chan, et al. [4] has been very successful in doing full body pose transfer. When using this model directly for the task of SLS, however, it was found to give unsatisfactory results [24]. This is mostly due the fact that the model does not account for sign language specific details, like the importance of generating hands clearly. This method has been further explored taking these specifications into account in [20] and related works. These works however, require manually selected images in order to train the model to generate images of the hand to a sufficient level of detail. Furthermore, they utilize a large corpus of annotated continuous sign language videos to train the model to generate sequences of signs.

Sign languages in general have a great dearth of any form of data, with ISL being no exception to this. In our work, therefore, we have aimed to overcome the downsides of previous GAN based SLS work with regards to this lack of data. Instead of having to manually create a dataset of clear hand images, we have utilized a separate generator for the purpose of generating this hand region alone. As combining the outputs of different GANs creates some artefacts at the boundaries, we further train a smoothing network to enforce consistency. To deal with the issue of creating continuous sign language output for sequences of signs, we experiment with rule based detection of sign onset and end, and interpolation between poses. Using this technique, we can generate individual signs using videos as available in an ISL lexicon, and stitch them together to create continuous signing output. This model is the first of its kind for ISL.

The main contributions of our work are:

- we create a GAN based SLS model, specifically for use with ISL. We implement a separate generator module for the hands, in order to maintain clarity. Handshapes are an essential part of sign language and their accurate reproduction is necessary for intelligibility.
- we implement a smoothing network to create an artefact free composition of the two generated outputs of the body and the hands. This provides a way to successfully combine output generated in separate GANs into one uniform video.
- we create a rule based framework to generate continuous sign language output from individual signs as available in a sign language video dictionary.

2 RELATED WORK

This work is a method of SLS using GANs for image production and builds on top of research in these two fields.

SLS involves controllable generation of visual sign language output. Usually in the past, SLS projects have used computer animations, using animated 3D avatars to generate this output [1, 2, 5, 11, 17]. This method requires a database of animations of the signs in the sign language to create the output. These individual sign animations are generally created using motion capture which requires very expensive equipment [6]. This motion capture data also needs to be cleaned for it to be usable, a task that requires technical knowledge and a large amount of time. Therefore, it has been very impractical to generate databases for the entire vocabulary of a sign language by this method, and projects have focused only on specific use cases such as railway information portals and hospital interactions, in order to reduce the total number of signs required in the database [2].

Alternatively, there have been attempts to symbolically parametrize signs, much like dividing words into phonemes [17]. However, for this method to accurately capture all nuances of sign language performance, a large amount of technical knowledge is required to build the animation system, and it still ends up missing important features such as expressiveness [6]. Furthermore, it is necessary to generate these symbolic parametrizations for each sign in the language, a task that requires expert linguistic knowledge. Due to these drawbacks, there has been a recent turn towards alternative approaches to SLS, mainly GAN based video generation.

In recent years, GANs have seen great success in conditional generation of images, with generation usually conditioned on a simplified representation of the image. They have been used to perform various tasks such as creating landscape photos, creating stylized photos and transferring expressions or poses between people [9, 18, 25]. The last example, also termed "do-as-I-do" motion transfer as established in [4], is most relevant to our work. Here, motion is transferred between people, conditioned on the pose information, which is usually obtained using another model like OpenPose [3]. Since this involves motion as opposed to still images, temporal consistency becomes important and is also incorporated in training.

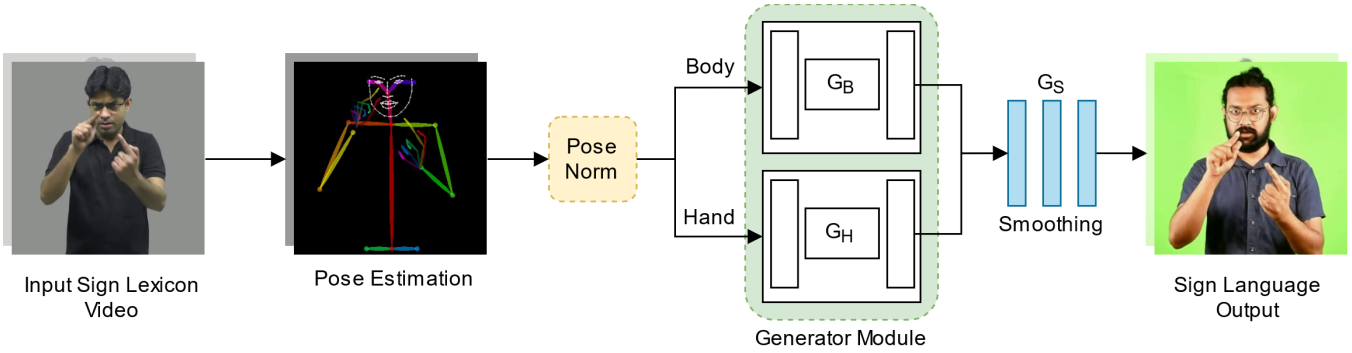

Figure 2: Overview of the Sign Language Transfer, From Input Dictionary Video [19], to Output by Target Signer

Directly using this method to generate sign language provided inadequate results [24], due to not taking into consideration sign language specific requirements, like accuracy in depicting hands. The approach was modified for use in SLS in [20, 22] and related works. Here, videos from multiple signers were used to train models that can perform style transfer, giving control over avatar appearance as well. Hence, a larger amount of data is required to train the model. To improve hand accuracy, they also use a manually selected subset of good hand images, something we wish to avoid. The project generates continuous sign language output by training on available corpora of annotated continuous sign language performance [13]. This form of data is very rare for sign languages due to the amount of labour and expertise required for annotation. We explore an alternative using rule-based combination of individual signs available in the ISLRTC dictionary [19].

Work on sign language processing in the context of Indian Sign Language (ISL) has been fairly limited, and mainly focused on Sign Language Recognition (SLR), either from RGB videos alone [12, 21], or using different depth related sensor information [10, 15]. In the field of SLS, there have been a few projects using either parametric sign animation [23] or a manually animated sign database [14]. Our work using GAN to perform SLS is, to our knowledge, the first of its kind for ISL.

3 METHOD

Our goal is synthesising a video of sign language performance, based on a "do-as-I-do" approach. This involves reproducing videos of signing, performed by different individuals, as output performed a single target individual. Continuous signing output involves sequences of multiple signs. Simply stitching together videos of different signs performed by different individuals with different camera parameters produces very unsatisfactory and jarring output. Our method enables playback of multiple signs by a single person consistently, without change in the performer or camera parameters. The model also gives control on video output solely based on the 2D skeleton, which can be modified to suit motion blending and other modifications. This pipeline involves pose estimation and normalization, followed by generation of entire image which involves separate generation of hands, and finally, combining the two images and refining to remove artefacts, to obtain the output (Fig. 2). Additionally, we experiment with producing a sequence of signs by combining different individual sign videos through a process we term "sign-stitching".

3.1 Pose Extraction and Normalization

We obtain the skeleton pose from video using available pretrained models provided by OpenPose [3] and MediaPipe [27]. OpenPose is used for the full body pose, and MediaPipe specifically for hands alone, based on the accuracy we observed. We use the skeleton pose image generated through these as the input. The ISLRTC video dictionary has several signers of different body shapes. To generate output using these videos, we therefore need to normalize the pose obtained from these images. We perform pose normalization through an affine transformation of the pose image frame. The three points chosen are the shoulders and neck joints, and these are transformed from the lexicon video values to the values in a single frame of the training video. For the hand generator, for our performer, we crop a 200x200 pixel section centered around each visible hand as input for the hand generator.

3.2 Pose to Sign Video Generation

Our method of video generation is more based on Everybody Dance Now (EBDN) [4], which is for general pose transfer, than on [20], which focuses on sign language generation. The former is trained on the performance of a single person, unlike the latter which requires multiple high quality recordings of several people to train. Furthermore, the latter requires a manually selected dataset of clean hand images for training. These two requirements are at odds with our goal of requiring minimal data, especially annotated data, to train, which explains our choice of EBDN as our base model. The generator is based on the pix2pixHD model [25], and uses a pair of generators at full scale and down-scaled to half the size, termed local and global respectively: $G = \{G_1, G_2\}$. For adversarial training, a set of three discriminators at different scales, $D = \{D_1, D_2, D_3\}$, is used to compete with the generators. Temporal consistency of the video sequence is also enforced in a similar fashion as in EBDN, generating pairs of consecutive frames and discriminating for the sequence of both to be realistic. The revised GAN objective to account for this smoothing is then given to be:

Figure 3: Sample Output on a Frame From the Test Dataset, L-R: Original Frame, pix2pixHD, EBDN, Hand Keypoint, Hand Generator, Full Model. Best Viewed in Colour

$$\mathcal{L}_{temporal}(G,D) = \mathbb{E}_{(x,y)}[logD(x_{t-1}, x_t, y_{t-1}, y_t)] + \mathbb{E}_x[log(1 - D(x_{t-1}, x_t, G(x_{t-1}), G(x_t)))] \quad (1)$$

where $t-1$ and t refer to the two consecutive time steps, x refers to the skeleton pose input and y refers to the corresponding frames of the original video.

3.2.1 Hand GAN. Proper generation of hands is vital for intelligibility of the produced signs. However, the hand is visually very complex, and also occupies a comparatively small area of the frame. This means that the model needs to be forced to pay attention to the hand section. Previous work on sign language GANs used a combination of adding a loss based on hand keypoints as obtained from the pose estimation model, as well as training on a dataset of "good hands" [20], presumably frames where the hand is not blurry and clearly visible. Our experiments using just the hand keypoint loss without this special manually cleaned subset of hand images gave unsatisfactory results.

Instead, taking inspiration from the Face GAN module of EBDN [4], we use a separate generator G_H working purely on generating a 200x200 pixel patch surrounding each visible hand. The Face GAN actually generates the residual refinement pixel differences, instead of the actual pixels themselves. This approach, however, gave inadequate results for hands, and we therefore generate the actual hand image completely. Also unlike EBDN, we train G_H to be temporally consistent as described previously, generating pairs of successive frames each time. This is because temporal consistency is very important, as the hands are very actively in motion during signing, as opposed to the relatively stiller facial features modelled in the Face GAN.

The hand generator G_H is trained after the full body generator G_B. However, instead of having a multiscale generator of global and local networks, a single global network is used. This is because of the much smaller size of the hand patch being generated. We had first experimented with a U-Net based model for the hand GAN, but the results were unsatisfactory, and we use the same global network architecture as used in G_B. The same discriminator as used for the entire body is used on the composite image from the outputs of both G_B and G_H to train G_H. This is done because G_H needs to generate not just the hands, but the rest of the patch around the hand as well, which includes other portions of the body. Therefore the full image discriminator is used to enforce this consistency. To increase homogeneity of the composite image, both G_B and G_H are trained together for a few epochs after this as a final step.

3.2.2 Smoothing GAN. Even after this final step of training both generators together, the edge of the hand patch is sometimes visible as an artefact of using separate generators. To smooth this out, a generator, G_S, based on the local generator of [25] is trained to smooth the final composite image. This generator is trained on the individual frame level, without the additional modifications for temporal smoothing. The same multiscale discriminator as used for the entire image is utilized to train this generator.

3.2.3 Objectives. The general objective function for both G_B and G_H are the same as as the general objective in EBDN [4]:

$$\min_G((\max_D \sum_{k=1,2,3} \mathcal{L}_{temporal}(G, D_k)) + \lambda_{FM} \sum_{k=1,2,3} \mathcal{L}_{FM}(G, D_k) + \lambda_P \mathcal{L}_P(G(x), y)) \quad (2)$$

Here $\mathcal{L}_{temporal}$ is the GAN objective modified for temporal consistency as described above (Eq. 1), $\mathcal{L}_{FM}$ is the feature matching loss and $\mathcal{L}_P(G(x), y))$ is the sum of the VGG perceptual losses at two successive time steps $t-1$ and t, comparing the generated outputs $(G(x_{t-1}), y_{t-1})$ and $(G(x_t), y_t)$. During the final stage of training both G_B and G_H together, the objectives of each module are summed to give the total objective function of the combined images.

The objective for the smoothing network G_S is simply the GAN objective for the local generator along with feature matching and VGG losses per frame, as in [25]:

$$\min_{G_S}((\max_D \sum_{k=1,2,3} \mathcal{L}_{GAN}(G_S, D_k)) + \lambda_{FM} \sum_{k=1,2,3} \mathcal{L}_{FM}(G_S, D_k) + \lambda_P \mathcal{L}_P(G_S(x), y)) \quad (3)$$

where $\mathcal{L}_{GAN}$ is the typical single frame GAN objective as given in [9].

3.3 Sign Stitching

Individual sign videos from the ISLRTC dictionary have a rest phase before and after sign production. This will result in long pauses if we generate sign sequences by directly concatenating signs generated from these videos. These pauses seem very unnatural and significantly affect intelligibility. Therefore, we explore removing

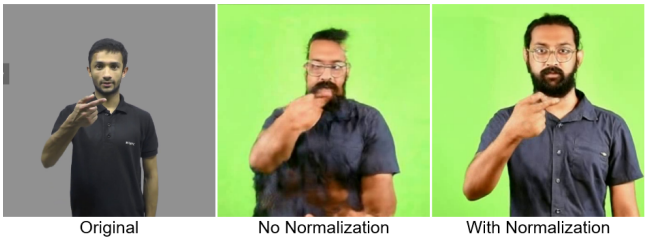

Figure 4: Effect of Normalization on Output on a Performer With Different Body Type

these pauses by detecting start and end points of the individual sign videos, followed by interpolating the trajectories of the pose between the signs using cubic splines. Since there are no annotations of start and end sign production timestamps in the videos, we created a set of handcrafted conditions to obtain these points. The conditions are based on location of the hands and hand motion. In the rest pose before and after the sign, the hands lie relaxed by the sides, and we detect the start and end of the signs by using limits on the height of the hands in the video. Sometimes, after the sign is completed, the pose is held still to provide emphasis, something which will be unnatural in continuous signing. To detect this, we check the velocity of the hand in the frame, as it goes to zero during this "hold" phase. Once we obtain start and end positions of signs, two consecutive signs have the skeleton motion interpolated using cubic splines for each of the joints. This is done to ensure smooth movement between signs.

4 EXPERIMENTS

4.1 Data

We used a video of a performer doing various signs of different handshapes for around 19 minutes. The performance was captured before a green screen, at 1280x720 resolution, and at 60 FPS to reduce motion blur in the hands. This was downsampled to 30 FPS to reduce the number of frames, and training was performed on this set of frames. The position of the performer was stationary at the center of the image frame. The video was also cropped down to 1024x512 to reduce shadows and other disturbances from the studio background.

4.2 Baseline Models

The main GAN generation model was evaluated against three baseline models:

- *pix2pixHD*: Here we use the image generation model as described in [25] to generate image sequences from video. This has no adjustments for temporal consistency.
- *EBDN*: This is the vanilla model from the "Everybody Dance Now" paper [4], without any additional losses or training to account for preserving hand details.
- *Hand Keypoint Model*: This is an extension of the previous model trained with additional keypoint loss for the hands as in [20]. Unlike the original work, we don't use a cleaned dataset of "good hands", however. We also use a single signer to train the model, and we have added our pose normalization method to make it competitive for use with different signers. The keypoint loss is obtained by running the keypoint estimation model on the original hand image and the generated hands and comparing the two with a discriminator:

$$\mathcal{L}_{HK}(G, D_H) = \mathbb{E}_{x_H, y_H}[log D_H(x_H, y_H)] + \mathbb{E}_{x_H}[log(1 - D_H(x_H, y'_H))] \quad (4)$$

where y_H are the hand keypoints obtained from the original image and y'_H are the hand keypoints obtained from the generated image.

For testing our method of sign stitching, we evaluated it against the sequences as generated by simply concatenating the individual sign videos from the dictionary.

4.3 Ablation Conditions

We tested the following ablation conditions for our model:

- *Hand GAN*: This is our first condition, which has an additional generator for generating the 200x200 hand patches.
- *Smoothing GAN (Full)*: In this condition we add the smoothing network to remove the artefacts formed at the edges of the hand patches generated in the previous condition.

4.4 Evaluation Parameters

We have used both the structural similarity index SSIM [26] and the Frèchet Inception Distance (FID) [8] scores to evaluate all models. Further, these two metrics have been applied at two scales: one at the entire image level, and one on patches of 250x250 centered around the hand. This is patch is larger than the patch generated by the Hand GAN specifically to account for artefacts at the edges of the generated patch.

4.5 Implementation Details

Training the full model requires training four networks, mostly independently, in sequence. The networks, in the order of training, are: the full body global network, the full body local network, the hand global network and the smoothing local network. These networks are trained for 12 epochs total each. In the case of the hand global and body local networks, they are first trained for 10 epochs separately, and finally they are trained together for 2 epochs, as described above. The networks are all trained on a single NVIDIA RTX 2080 Ti GPU with 12 GB RAM. Both global networks take roughly two days each to train, the body local network takes around five days and the smoothing local network takes three days in training. The total model, therefore, takes around twelve days of training.

5 RESULTS

5.1 Quantitative Results

Our model is evaluated against baselines and ablation condition as described in the above section.

For the perceptual preference study, 16 pairs of videos of individual signs were created, each pair containing the output of the full model and the output of the baseline or ablated method. Along with

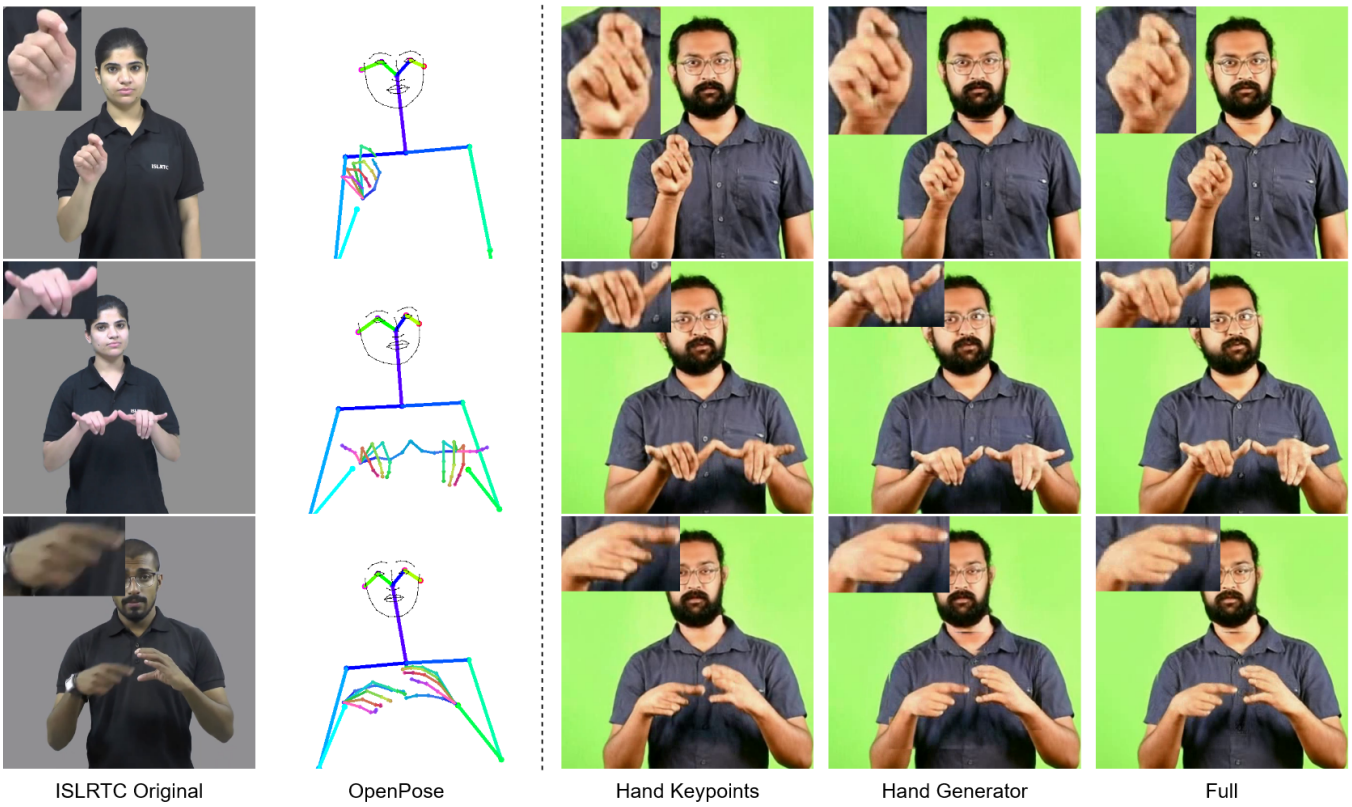

Figure 5: Sample Output on Videos From the ISLRTC Dictionary. Best Viewed in Colour [19]

Table 1: Perceptual Study Results for the GAN Models. Value Shown Is Percentage of Participants Preferring Our Method to the Baseline and Ablated Methods

Method	Percentage Favourable
pix2pixHD	94.4%
EBDN	97.2%
Hand Keypoints	72.2%
Hand Generator	80.6%

Table 2: Perceptual Study Results for Sign Stitching. Value Shown Is Percentage of Participants Preferring Our Method to the Baseline

Method	Percentage Favourable
Concatenate	96.7%

Table 3: Baseline Comparison Results

Method	FID ↓		SSIM ↑	
	Body	Hand	Body	Hand
pix2pixHD	14.37	18.79	0.796	0.690
EBDN	12.15	15.06	**0.817**	0.716
Hand Keypoints	**7.61**	**9.43**	0.782	**0.813**
Ours (Full)	9.15	11.05	0.794	0.779

Table 4: Ablation Comparison Results

Method	FID ↓		SSIM ↑	
	Body	Hand	Body	Hand
Hand Generator	9.35	15.16	0.782	0.758
Full	**9.15**	**11.05**	**0.794**	**0.779**

this, videos for 6 sentences were created to test the sign stitching technique against the baseline of concatenating entire individual sign videos as described above. A total of ten participants took part in the survey, divided equally between five people who knew ISL and five laymen without ISL knowledge. Among the ISL users, two were Hearing Impaired and three were a mix of ISL interpreters, and hearing students of ISL. All participants evaluated the individual sign videos, but the sentence videos were only evaluated by ISL users. This is because linguistic information like pause length can only be reliably evaluated by people who know the language.

Table 5: Execution Times Expressed in Frames Generated Per Second

Method	Execution Speed (Hz)
Hand Keypoint	1.71
Hand Generator	1.44
Ours (Full)	1.53

The results of the survey, presented in Table 1, clearly showed that our method was preferred as compared to the rest of the baseline methods, as well as the ablation condition. Among the competing methods, the hand keypoints baseline modelled on Stoll, et al. [22] performed best. However, even here we noted that the only times it was sometimes preferred over our method was when the hand was prominently in front of the face. In this condition, since the patch generated by the hand overlaps the face, there may be a degradation of facial features. In the case of testing the sign stitching technique, there was an almost unanimous preference of our method over the baseline (Table 2). This is expected due to the long pauses being very unnatural and interfering greatly with the perception of the language.

The results of SSIM and FID metrics for both body and hand are presented in Tables 3 and 4, for the baseline models and ablation condition respectively. In the baseline models, our model is only outperformed by the hand keypoints model, in both the SSIM and FID scores. This result is at odds with the results of the perceptual study, where our model clearly outperforms all other models. Qualitative analysis also shows our model giving better results as discussed in the next section. We believe this discrepancy in the results using these metrics to reflect the inadequacy of both SSIM and FID to accurately encapsulate human perception. In case of the ablation condition however, we see a clear improvement on adding the smoothing network. This is also evident in the improvement of consistency in coloration and capturing finer details when the smoothing network is added, and the improved score corroborates this.

Finally, we tested the execution time of our method against the current prevalent SLS method of [22]. This was mainly to see if the additional generators that are used in our method significantly affects the execution time. We see there is an average increase of around 10%, which is acceptable considering the decidedly improved output of our model. In fact we found the execution time varies significantly from one run to the next, with the range of percentage increase in time being between 3% to 15%.

5.2 Qualitative Results

The outputs of the baseline and ablated methods on a frame from the validation dataset can be seen in Fig. 3. As visible in the figure, our final model has much clearer hand generation than both the baseline methods as well as the ablation condition. Additionally, the ablation result has discoloration and edge artefacts due to the separate hand patch generation. This can be seen to visibly improve on addition of the smoothing network.

Our model is meant to be used with a video dictionary which contains several signers performing the different signs in it. These signers have different body types, and furthermore, there are differences in the camera angles employed across the different signs in the dictionary. Hence, we implemented an affine transform based normalization method to account for this. The effect of adding this normalization is clearly visible in Fig. 4, where the performer has a significantly slimmer build than the performer in the training video. As can been seen here, the generation fails when there is a marked difference in body types, and this is effectively overcome by our normalization technique.

Finally, we show a comparison of outputs of the hand keypoint model and our ablation condition, versus our final model, in Fig. 5. Here too, we see our final model outperforming the others. There is a marked improvement over the hand keypoint model when it comes to capturing the finer details of the hands such as clear demarcation of individual fingers. In the case of the ablation condition, the improvement is not as evident, but an improvement in the general image quality is visible, with features such as shirt buttons being clearer, and consistent coloration being present in the full model output. Another interesting observation is the removal of motion blur that is present in the original image. This is due to the fact that the training was done on video captured at 60 FPS which has significantly less motion blur, as well as the robustness of the pose estimation model used.

6 CONCLUSION

Very recent methods in Sign Language Synthesis (SLS) have started using GANs to generate photo-realistic output of controllable sign language performance. However, these methods rely on data that is not readily available for most sign languages currently, including for ISL. In this paper, we presented a method to achieve these results while requiring no specially constructed datasets.

Instead of requiring high quality videos of multiple performers of sign language and a cleaned dataset of hand images, we use a novel combination of using an additional generator module to recreate the hands, along with an affine transform based normalization technique to ensure performance across different individuals. In addition to this, we implement a smoothing network to perform the task of removing artefacts created by using a combination of separate generators to create the full image. Using these methods, we are able to successfully utilize an existing comprehensive ISL dictionary to potentially create most of the signs in ISL.

Previous SLS methods have also utilized extensively annotated continuous sign language performance to train models to create continuous signing output. As this is available for only a couple of sign languages currently, we have devised a method to naturally stitch together signs from different individual sign videos present in the ISL dictionary. This is done to get rid of the large pauses that are present at the start and end of each of these individual videos. We create a rule based technique dependant on the position and velocity of the hands to do this, and demonstrate it performs much better than the baseline technique.

Therefore, we have presented a successful adaptation of a GAN based SLS method specifically for ISL, overcoming the requirements

of annotated data that current techniques relied on. This SLS model is the first of its kind for ISL.

ACKNOWLEDGMENTS

This work was funded by the Mphasis Foundation CSR grant to IIITB.

REFERENCES

[1] Inês Almeida. 2014. Exploring Challenges in Avatar-based Translation from European Portuguese to Portuguese Sign Language.

[2] Andrew Bangham, Stephen Cox, John Glauert, I. Marshall, S. Rankov, and Mariah Wells. 2000. Virtual signing: capture, animation, storage and transmission-an overview of the ViSiCAST project. 6/1 – 6/7. https://doi.org/10.1049/ic:20000136

[3] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. 2019. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019).

[4] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A Efros. 2019. Everybody dance now. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5933–5942.

[5] A. Conway and T. Veale. 1994. A Linguistic Approach to Sign Language Synthesis. In *BCS HCI*.

[6] Sylvie Gibet, Nicolas Courty, Kyle Duarte, and Thibaut Le Naour. 2011. The signcom system for data-driven animation of interactive virtual signers: Methodology and evaluation. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 1, 1 (2011), 1–23.

[7] Raymond G Gordon Jr. 2005. Ethnologue, languages of the world. *http://www.ethnologue. com/* (2005).

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 30 (2017).

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. *CVPR* (2017).

[10] Nayan M. Kakoty and Manalee Dev Sharma. 2018. Recognition of Sign Language Alphabets and Numbers based on Hand Kinematics using A Data Glove. *Procedia Computer Science* 133 (2018), 55–62. https://doi.org/10.1016/j.procs.2018.07.008 International Conference on Robotics and Smart Manufacturing (RoSMa2018).

[11] Richard Kennaway, John Glauert, and Inge Zwitserlood. 2007. Providing signed content on the Internet by synthesized animation. *ACM Transactions on Computer-Human Interaction (TOCHI)* 14 (09 2007), 15. https://doi.org/10.1145/1279700.1279705

[12] PVV Kishore and P Rajesh Kumar. 2012. A video based Indian sign language recognition system (INSLR) using wavelet transform and fuzzy logic. *International Journal of Engineering and Technology* 4, 5 (2012), 537.

[13] Oscar Koller, Jens Forster, and Hermann Ney. 2015. Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. *Computer Vision and Image Understanding* 141 (Dec. 2015), 108–125.

[14] Shyam Krishna, Ankit Rajiv Jindal, Mahesh R, Rahul K, and Dinesh Jayagopi. 2020. Virtual Indian Sign Language Interpreter. In *Proceedings of the 2020 4th International Conference on Vision, Image and Signal Processing* (Bangkok, Thailand) *(ICVISP 2020)*. Association for Computing Machinery, New York, NY, USA, Article 16, 5 pages. https://doi.org/10.1145/3448823.3448830

[15] Pradeep Kumar, Himaanshu Gauba, Partha Pratim Roy, and Debi Prosad Dogra. 2017. A multimodal framework for sensor based sign language recognition. *Neurocomputing* 259 (2017), 21–38. https://doi.org/10.1016/j.neucom.2016.08.132 Multimodal Media Data Understanding and Analytics.

[16] Fiona Kyle and Kate Cain. 2015. A Comparison of Deaf and Hearing Children's Reading Comprehension Profiles. *Topics in Language Disorders* 35 (04 2015), 144–156. https://doi.org/10.1097/TLD.0000000000000053

[17] Ian Marshall and Éva Sáfár. 2003. A Prototype Text to British Sign Language (BSL) Translation System. 113–116. https://doi.org/10.3115/1075178.1075194

[18] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. GauGAN: semantic image synthesis with spatially adaptive normalization. In *ACM SIGGRAPH 2019 Real-Time Live!* 1–1.

[19] Indian Sign Language Research and Training Center. [n.d.]. ISLRTC New Delhi. http://www.islrtc.nic.in/.

[20] Ben Saunders, Necati Cihan Camgoz, and Richard Bowden. 2020. Everybody sign now: Translating spoken language to photo realistic sign language video. *arXiv preprint arXiv:2011.09846* (2020).

[21] Advaith Sridhar, Rohith Gandhi Ganesan, Pratyush Kumar, and Mitesh Khapra. 2020. INCLUDE: A Large Scale Dataset for Indian Sign Language Recognition. In *Proceedings of the 28th ACM International Conference on Multimedia* (Seattle, WA, USA) *(MM '20)*. Association for Computing Machinery, New York, NY, USA, 1366–1375. https://doi.org/10.1145/3394171.3413528

[22] Stephanie Stoll, Necati Camgoz, Simon Hadfield, and Richard Bowden. 2020. Text2Sign: Towards Sign Language Production Using Neural Machine Translation and Generative Adversarial Networks. *International Journal of Computer Vision* (04 2020). https://doi.org/10.1007/s11263-019-01281-2

[23] Sugandhi, Parteek Kumar, and Sanmeet Kaur. 2020. Sign Language Generation System Based on Indian Sign Language Grammar. 19, 4, Article 54 (April 2020), 26 pages. https://doi.org/10.1145/3384202

[24] Lucas Ventura, Amanda Duarte, and Xavier Giró-i Nieto. 2020. Can everybody sign now? Exploring sign language video generation from 2D poses. *arXiv preprint arXiv:2012.10941* (2020).

[25] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. 2018. High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[26] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[27] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, and Matthias Grundmann. 2020. MediaPipe Hands: On-device Real-time Hand Tracking. arXiv:2006.10214 [cs.CV]