Semi-automatic Deep Learning based animation of 2D videos using 3D body-pose estimation for text to Indian Sign Language.

Abstract— **Sign language translation was formerly approached using methods that were mostly reliant on expensive technologies such as Motion Capture. While extremely precise, they demand, however, heavy resources of skilled labor and cost and hence are less cost effective for usage. Thus, we adopt computer vision approaches such as the popular convolutional neural networks, however, to do so labeled datasets in large numbers and in sound quality are required. Indian sign language lacks a standard corpus for such 3D motion data. This project aims at leveraging Computer Vision, therein, Deep Learning to obtain a set of 3D pose datasets from 2D videos of single-person signing words from the Indian Sign Language using pose estimation. In this work, 3D skeletal data from 2D RGB videos are obtained with the help of a convolutional neural network-based 3D hourglass network. Using supervised learning, the 2D pose file obtained using OpenPose in JSON format which is further post-processed for missing joints and occlusions and transformed into 3D data using a fully connected depth-regressor to obtain the depth information in the frames. This skeletal information is obtained in the form of a .bvh file which makes the rendering of the motion capture data easier on any external animation software. The proposed architecture provides a stellar performance rate even in the CPU processor. This paves way for the creation of an Indian Sign Language motion capture database that can be used for animation in the future which aids applications such as text to sign language or further speech to sign language which would benefit the deaf community largely.**

Keywords— Pose Estimation; Convolutional Networks; Computer Vision; Artificial Intelligence;

I. Introduction

A sign synthesizer is constructed with an intermediate database to provide easy access to the signs corresponding to each text and vice versa. This helps the signer or the non-signer to accumulate the interpretation quickly without the necessity of waiting for processing. The development of this database is significant for this sub-phase. A pose estimator is used to get a set of 137 human pose key points from the input videos to acquire the sign data needed for the database. This 2D data is further inflated to 3D data by an interface that supports the synthesis of the sign data. There exists a lot of prior works in this area but a large part of them is based on American Sign Language. The result of these models is highly accurate but are very expensive to accumulate as they include the use of expensive methods like Motion Capture technology, wearable sensors, or gloves. These pre-existing models do not explain how linguistic communication translation will be employed in real-time.

Thus, a deep learning approach is used to get real-world experience with limited hardware. Intensive multimodal models cannot be used with the existing data. So, automatic data acquisition techniques are created to reduce the extensive data collection and annotation work. This model detects accurate hand and face movements along with the position of each finger. Human Pose Estimation is applied to every frame of the video to detect these features.

The deaf or the hard-of-hearing community is very vast, which is roughly 25% or more of the total population of the country which is 5 million people in India. The means of communication for such a huge community is only one i.e. the Indian Sign language. But the amount of material available to them is not enough to educate all equally. In Spite of that, the awareness regarding Indian Sign Language is very less and the measures taken to provide improved learning material and spread the importance is equal to none. This results in their highest level of education of students who are 15 or more to be secondary level for 20% of them or in worst cases it is just till the primary level for 12% of them. Since the whole world is moving towards digitization and all the book learning is replaced with massive open-sourced courses and material, the deaf community is not able to keep up with the fast-moving world as they are still dependent on native methods and interfaces for self-learning. This is due the language barrier and inaccessibility to the online material as they are not present in the Indian Sign Language. Due to lack of this convenience to access such sources, they suffer from isolation and underrepresentation.

Almost all the information that is communicated is done through audio which includes announcements in educational institutions or the announcements through television. This puts the disabled community at a huge disadvantage of not able to understand and implement these announcements in such situations. Since, conversing in sign language is much faster than its linguistic method, they usually prefer interpreting signs over reading the script. This makes it important to translate text or audio into an articulatory manner.

The use of two hands is very prominent in Indian Sign Language while in the American Sign Language, mostly a single hand is used. The rules followed by the text to sign approach are similar to the ones needed for an audio to sign approach. The sign can be synthesized in two ways: articulatory and concatenative. The articulatory approach replicates how speech is articulated while the concatenative approach joins the words together in a word-by-word representation for each word in the statement.

This documentation focuses on the Convolutional Pose Machine used to extract human pose estimation from RGB videos. The data in human pose contains a total of 115 key points in which 25 points are for the pose, 70 for face, and 21 points each for either hand in each frame of the input video. The entire 2D pose of the signer from the videos is checked for missing joint key points and hand occlusion errors by means of an interpolator in the post-processing step of the module. This pose data is then stored in a .csv format file.

II. Related Work

Pose estimation is said to be the most challenging works in the field of Computer Vision. The goal of a pose estimation model is to provide the key points in the body of an individual when an image is provided to it. This proves as a very important base for various applications like the sign language interpretation etc. Previously, the use of contours, edges, the histogram of gradient features etc. were extensive. But later, plenty of 2D Pose estimation neural network architectures were created and made available to the people. Training a supervised regression network to obtain joint location directly was a basic strategy. Modern works such as the model architecture by Tekin et al[3], worked on combining 2D and the 3D cues from the images with an improved accuracy percentage.

As we know, tracking hand and face movements are very important with regard to sign language. OpenPose[4] and Google's Media Pipe[5] are the only two applications that can track hand and facial joint positions beyond 2D pose estimation. Although, the hand and the face joints detection is restricted to the dimensions provided by the image. According to an assumption the systematic analysis of the heatmap is often not taken into account. Dark Pose[6] combats this assumption by providing an unbiased method to decode in a principle distribution aware way. But this method is dedicated to the body pose estimation only.

After all the videos are curated, several undesired camera jitters and movements are found which if not rectified might affect the continuity of the estimated pose. 2D strategies are advised to detect the intrinsic properties of the video since the input is a 2D video. Offline motion smoothing is the most commonly used existing technology. To smooth the camera's motion path in [7] with 2D translational, Gaussian filtering is used. Adding black-border constraints naturally to the problem that is solved by optimization has proven to be a huge advantage. Under the 2D Euclidean model, L2 norm of second order difference is used by [8] in camera motion under 2D Euclidean model as the objective function. The Euclidean transformation consists of proper rotations and pure translations that prevent the handedness of the figures. The motion smoothing problem faced can be solved using all of the above together [9].

There are a lot of works that are based on images but only a few previous works dedicated to videos and most of them felt the need to create a densely labeled dataset for the production of effective results. There are a few datasets that are created over the years to resolve this issue. However, only a limited number of works have aimed at the estimation of fine-grain body parts. Previous models have suggested the difficulties of working with videos such as motion blur, flickering, sequence dissimilarity, and wrong detection of joints.

A lot of recent works have proposed the use of 2D to 3D pose estimation to increase accuracy and robustness. This makes it more effective than the basic plan of getting a direct inference of joint locations[10] did not prove to give the best results. This was put forward by the network architecture produced by Tekin et al.[3]. OpenPose is a renowned stick figure estimation model used in this field but it provides only the 2D joint positions for a single camera input video. However, to extract 3D human pose from the 2D pose estimation data from OpenPose, a model named 3d-pose-baseline was proposed by Martinez et al.[11]. This cured the problem related to the missing third dimension with the help of a simple feed-forward network that produces a 3D stick figure output. There exists a VNect model by Mehta et al.[12] that enhances the robustness of the stick figure output with the help of the kinematic information of the human skeleton. This CNN regresses the 2D and 3D joint formulation in real-time without the need for a cropped frame dataset. The issue with the Vnect model is the misinterpretation of 2D key points in the 3D frame. The result of all these models is a human stick figure hence it lacks the estimation of the face and the finger key-points that are crucial for sign detection in an ISL video.

For a video consisting of only one person, LSTM pose machines by Luo et al. provided a solution to all the problems with video pose estimation by converting CNNs to RNNs with the help of a specific weight sharing mechanism introduced inside the network. RNNs are proved to be good at calculating every step by considering the previous result. Also, it provides a way to reduce the number of connections in the transformed network. This decreases the processing time of the videos and the number of parameters used in the network. Even though some of the results of the LSTM model are fast and reliable, they lack precision and are too variant to the dataset split. This issue was resolved by the stacked hour-glass network model.

III. Proposed Method

A. Dataset Used

Lack of annotated dataset is a major problem faced while a Sign Language Translator (SLT) is constructed. The collection and the annotation of the SLT dataset is a very difficult and exhausting task. Therefore, an American Sign Language (ASL) trained data is used.” How2Sign” dataset is a compilation of multiple view and signing videos of the American Sign Language containing around 2500 instructional videos from the already existing “How2” dataset. Detailed 3D reconstruction and pose estimation is enabled by a subset in a geodesic dome setup using hundreds of sensors and cameras. This helps the vision system to understand the 3D geometry of the sign language

ISLRTC provides a standardization to ISL, before this ISL used to have different signs for a single word which is often confusing. The development and testing of this project is entirely based on the standards set by ISLRTC. A video in this dataset typically consists of a signer, signing a word or a phrase with the use of his/her upper body only. Since, the lower body key points are set to a standard in order to avoid the error that occurs due to the missing key point.

B. 2D Pose keypoints

The output of the Open Pose network is a 2D data collection that contains noise and errors. These errors are found in the output due to various reasons such as quick hand movements resulting in capture of a blur frame or the lost tracking joints or occlusions between hands. Hence, to get the desired results, the data cannot be used directly. All the data in the dataset has to be pre-processed before it is used.

Stabilization of a video is firstly done by detecting the key points using FAST. Later, the Euclidean transformation for smoothening is applied on the displacement in x and y directions in every frame as illustrated in Fig 5. Optical flow and global motion parameters are used to detect the flow of transformation from the previous to the current frame. While estimating motion vectors, a local vector based scattered spectrum is obtained. The only two things that remain unchanged are the 2D transformational model angle and their relative lengths.

Smoothening out of the trajectory is done once all the trajectories are collected. This is done using an averaging window. A new transformation is applied to the video only after the generation of a new set of transforms.

Stabilization of video was accomplished by improving key point estimation and acquiring of a non-shaky sequence pose visualization. This has provided better results as compared to when raw video with jitters were used.

In a variety of computer vision-based tasks, corners are detected to perform object tracking and mapping. These corners are detected by a salient feature extractor i.e. the Features from accelerated segment test (FAST). A pixel containing a well-defined coordinate can be detected robustly and is called an interest point Rich local information is present in these points and hence they repeat across different frames. Moreover, the reasons why the results of a key point detector are better are discussed in the results and discussion section.

A salient feature extractor- Features from accelerated segment test (FAST) detects corners which could be used to perform object tracking and mapping in many computer vision-based tasks. Faster than many other popular feature extraction methods, such as Difference of Gaussians (DoG) used by SIFT, SUSAN and Harris, it was devised to find an Interest point in an image. A pixel which has a well-defined coordinate can be detected robustly and is called an Interest point. These points possess rich local information and ideally repeats across various frames.

A plot showing the values taken by the global motion parameters represented by θ (delta angle), Δx (delta x) and Δy (delta y) for the transforms generated for every frame

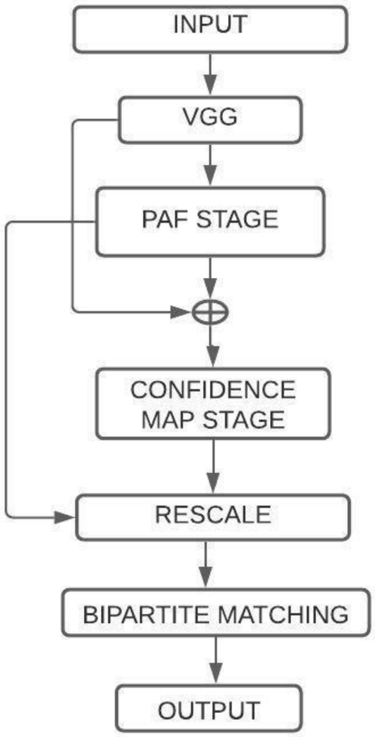
An open-source model called Open Pose is used to detect human key points due to its versatility in detection. Pose information extraction incurred during sign language synthesis is a very exhausting process. Apart from hands, face and body key points are also required for complete regeneration. Therefore, Open Pose is the best option as it detects all the human body key points including the most important parts for determining the sign of a word or phrase i.e. face and hands.

Convolutional Pose Machines (CPM) is used as a multi stage architecture for pose extraction by OpenPose. Receptive fields in CPM provide long-ranged details about the spatial relationships. The problems faced due to the vanishing gradients are solved at the intermediate level with the help of supervision. In Fig2, the base architecture of Open Pose is shown in which the Part Affinity Field (PAF) [15] network is present. Part-to-Part association and the encoding of detection map of confidence is predicted iteratively during the PAF stage. The 2D vectors representing the orientation of one key point to the other are called PAFs, Examining the input image and generation of the heat map is conducted by a convolutional network (VGG-19) at the starting stage. Final set of PAF channels consist of concatenated previous and subsequent stages which are fed into the Φ network. The key point confidence maps are predicted through the PAF channels and the feature maps that are fed into the network ρ. Non-Maximum suppression is applied on the confidence maps to find the discrete sets of body part candidate locations. Lastly, bipartite graphs matching the connections that share the same part detection candidate detect the whole-body poses of each person in the input video.

After the video recordings are passed on to Open Pose's BODY_25,pose key points are attained. The human pose coordinates that include the face, body, left and right-hand key points are saved as JSON files containing the videos mapped with key points and it might contain the background or not as illustrated in Fig3. with key-points with or without the background as shown in Fig. 3.

Although, the output attained by using OpenPose works fine with most of the frames of the input but in some cases, it fails to detect some body joints. The JSON files obtained after processing are further converted into .csv for easy access of data and visualization. When known points are used with known positional values while interpolation of that data takes place, good estimation of the probable location of unknown joints occurs, this is shown in reference [16]. The gaps present in the trajectories in some frames are a result of a variety of reasons like occlusion or failure of the model to estimate them. The gaps in the key points are filled with Akima interpolation techniques as it uses differentiable sub-spline. This is constructed with the help of piecewise cubic polynomials.

The finished pose framework gives an appropriate dataset for the further modules. It helps in work towards text to sign language. The 2D key points obtained can be useful to predict the 3D pose estimation and then it can help in obtaining skeletal animation for a 3D model which gives a sign to a corresponding word input.



C. Pose Data Preprocessing

The obtained raw 2D pose data needs to be preprocessed into a representation suitable for the convolutional network. The JSON file includes frame by frame pose data. However, in order to derive the meaningful pose information and spatial information, we need to fix a coordinate frame for every pose data. A reference frame and a center w.r.t the pose's local framework are defined. Further, points in the pose framework are represented as a coordinate vector to this center using 3 x 3 matrix transformation to obtain relative pose. Since a 2 x 2 matrix representation calls for heavy trigonometric computations, we resort to a 3 x 3 representation. Hence the point (x, y) needs to be represented as homogenous points (x, y, 1). The transformations applied include rotating, scaling and applying sheer in the pose data.

Algorithm to get Transformation matrix:

rot= rotation angle

res= 64

getTransformation (center, scale, rot, res):

$$A = res / scale$$

$$X' = res * (- center[0] / scale + 0.5)$$

$$Y' = res * (- center[1] / scale + 0.5)$$

$$t = (A\ 0\ X'\ 0\ A\ Y'\ 0\ 0\ 0)$$

if rot is not 0:

$$\Theta = - rot * \Pi / 180$$

$$R = (Cos(\Theta)\ - Sin(\Theta)\ 0\ Sin(\Theta)\ Cos(\Theta)\ 0\ 0\ 0\ 1$$

$$T = (1\ 0\ - res/2\ 0\ 1\ - res/2\ 0\ 0\ 1)$$

$$T\_inv = (1\ 0\ res/2\ 0\ 1\ res/2\ 0\ 0\ 1)$$

$$t = ((T\_inv . R) . T) . t$$

t - gives the transform matrix

The resultant matrix t gives the transformation matrix for the given pose framework. This matrix is used to find the point of the other joints in the body pose w.r.t the reference framework.

For a point (x,y) the resultant (X,Y) can be given as

$$(X, Y, 1) = t . (x, y, 1)$$

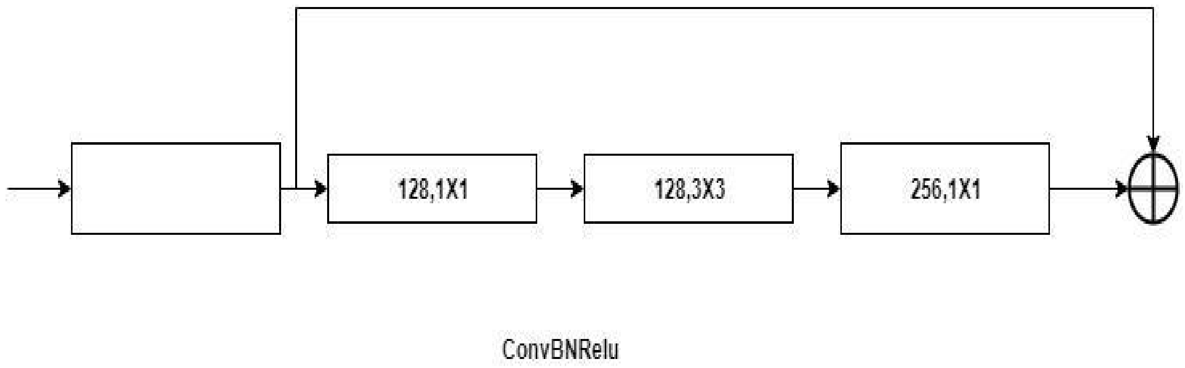
D. Network Architecture

It has been confirmed by several studies that an encoder-decoder architecture can make the designed network lighter and efficient. Hourglass 3D network module used in this document is vividly similar to designs that are fully convolutional and are capable of processing spatial information for depth prediction. The hourglass structure of the model comes from successive convolutional and deconvolutional layers giving it an appearance of an hourglass and hence the name. On different scales, a skip layer with a residual block is introduced to provide a crossover between these layers.

The image resolution is initially reduced and the first convolutional layer or the encoder part of the network extracts the features of the image. This deconstruction using convolutional layers and max pooling of the image provides a feature matrix. We introduce inflation of convNETs used in action recognition applications previously to create 3DconvNET. These conventional 3DconvNETS contain spatial-temporal filters which directly create the hierarchical data representation in the spatial temporal region. Instead of building 3D convNETs from scratch, by simply converting state-of-the-art 2D pose models into 3D ConvNETs we obtain a high performance spatial-temporal model. By starting with the 2D hourglass architecture, all the filters and pooling kernels are inflated to give a z-value as temporal.

Hourglass NET follows a symmetric order. Hence for every encoder layer, we introduce a decoder layer. The succeeding deconvolutional or decoder layers takes this feature matrix with low spatial information and combines it from the rich spatial understanding from the previous layer. The residuals and the skip layers of the network aid this reassembling step. While the output from the skip layer provides a reflection from the previous layers, it also reiterates the high-level spatial knowledge of the features deconstructed by the decoder from one stack (n-1) to the next stack (n).

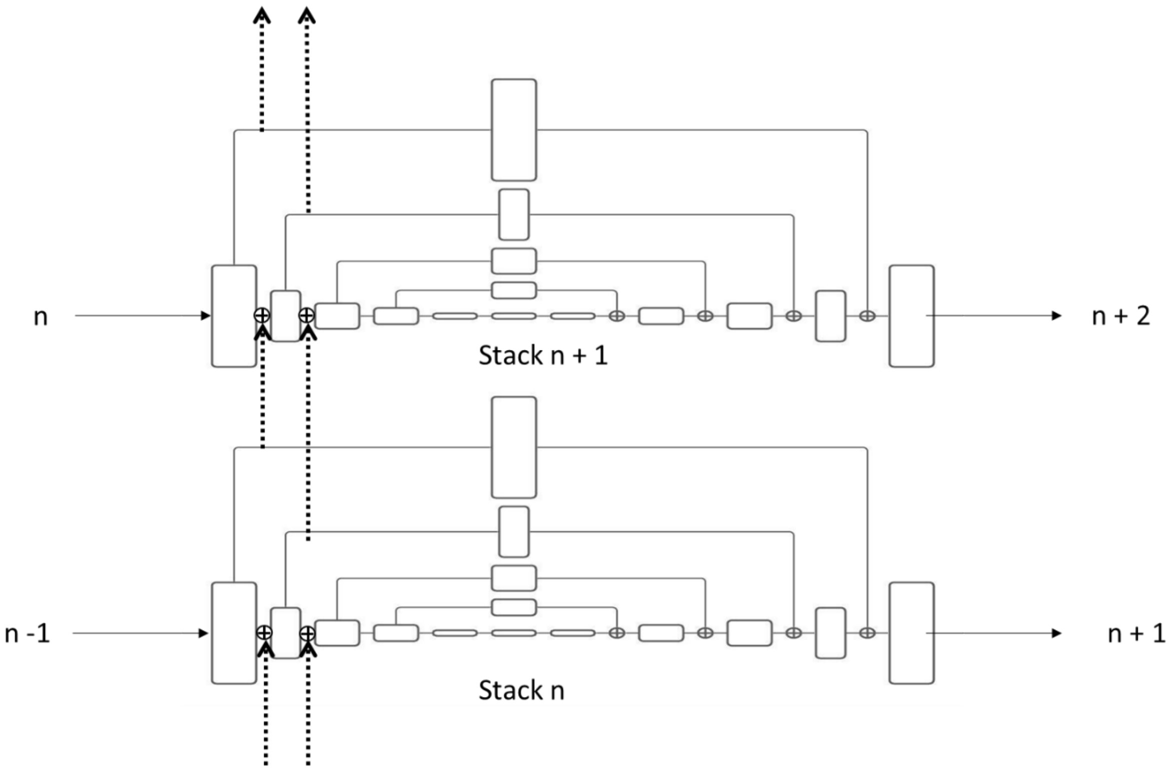
The residual models used in the network provide a single pipeline for spatial information retainment at each resolution. Bottlenecking the blocks to never use a convolutional layer greater than 3 x 3 makes the computations easier and the network runs on lesser memory.



Residual Block

In this document, Hourglass 3D models are stacked one above the other in an end-to-end manner to form a stacked 3D Hourglass model. Each stack consists of 2 Residual3D modules. We have tabulated the results for 1-STACK, 2-STACK and 4-STACK models, this has allowed the network to perform bottom-up and top-down processing repeatedly giving room for re-evaluation of estimates made initially to obtain higher-level spatial and temporal knowledge about the pose in the frame. Pose key points and their information is of sensitively high importance in this application. Since the output is a function of many factors that needs to be integrated to derive a meaningful interpretation of the scenario in the frame. By repeatedly iterating through the network, the spatial and temporal information can update itself by checking and then re-checking the consistency of the features to ultimately obtain high precision.

The input resolution of 256 demands a very high GPU and, thus, the output resolution is set as 64 to obtain smoother processing of data. Pretrained weights of a model trained over NTU Action Database is used and retrained using 15 videos from our database. The 15 videos give an average of 300 frames per video.



E. *DepthRegression3D Module*

In paper Xingyi et. al. innovate approach to integrate the 2D and 3D modules for a similar end-to-end network is seen. We adapt this approach as well as the usage of a loss obtained with a 3D geometric constraint in this model as the DepthRegression3D module.

Integration of 2D and 3D modules.

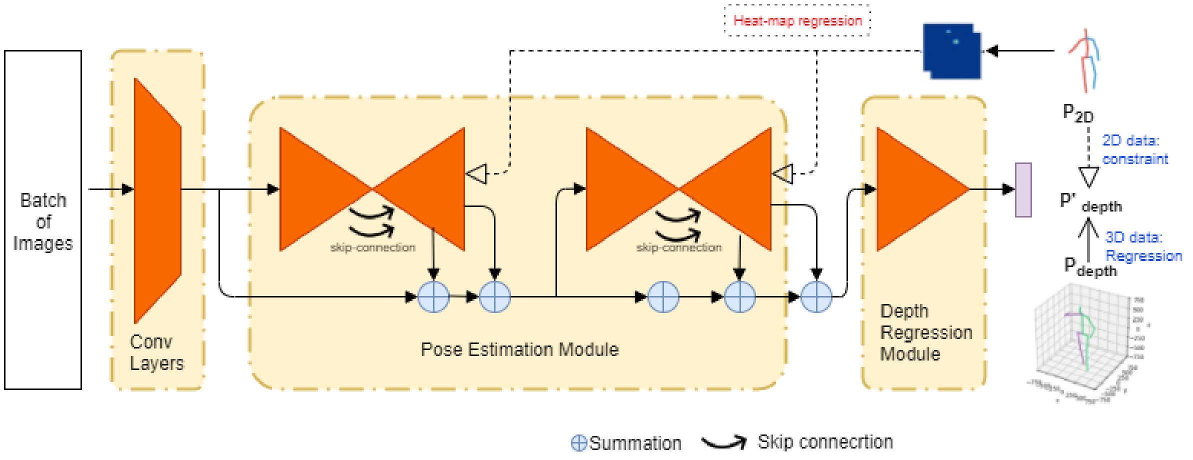
Estimation of the depth parameter from a 2D image faces a crucial bottleneck on knowing ways to exploit existing 2D image features to derive the depth value. If 2D joint locations alone are considered for depth estimation, an inherently highly ambiguous 3D point is obtained since multiple points in the 3D cloud around a single 2D skeleton fits this criteria. Thus, a unique integration of the heatmaps obtained from the 2D joints and the feature matrices, obtained at the intermediate levels of the previous module, leads to a semantically rich data that can be further processed for depth in this module. This fused data contains various other information essential for 3D recovery as mentioned below.

Combining partially labelled and completely labelled images is a major challenge in the Computer Vision domain. For labelled 3D dataset represented as $P_{3D} = (X_{3D}, Y_{3D}, P_{depth})$, using ground-truth depth level training loss can be directly given by the Euclidean Loss. For datasets that are not annotated, $P_{2D} = (X_{2D}, Y_{2D})$, a loss using a geometric constraint is computed. The loss of the depth regression module is

$$\mathrm{Loss}_{dep}(P'_{depth} \mid X, Y_{2D}) = \begin{cases} \lambda_{reg} \| P_{depth} - P'_{depth} \|^2 , \text{ if } X \in X_{3D} \\ \lambda_{geo} L_{geo}(P'_{depth} \mid Y_{2D}), \text{ if } X \in X_{2D} \end{cases}$$

where λ_{reg} and λ_{geo} are the corresponding loss weights.

$L_{geo}(P'_{depth} \mid P_{2D})$ gives the geometric loss. The underlying assumption in this approach is that the length of the bone or length between any two points remain the same in the human body. The length of every bone is considered to be the function of the corresponding joint locations, which are consequently functions of the depth predictions.

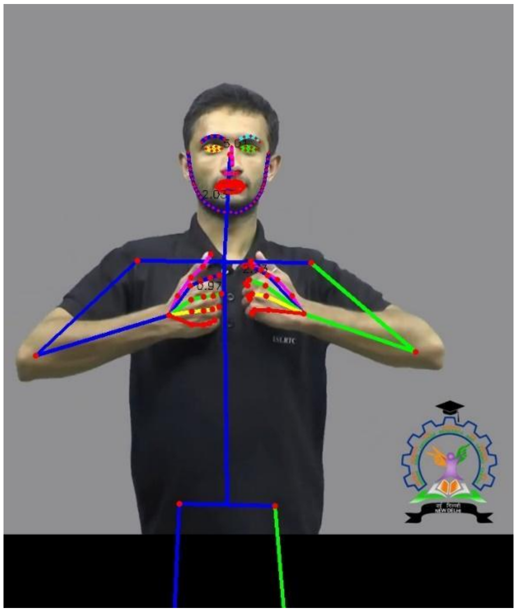


IV. Training Details

Evaluation of the pretrained network was done on NTU dataset. The sample data retrained in this project consists of 15 single-person videos. The hip joints are standardly taken to compute the center of the human. Resizing of all the input frames to 256 x 256 pixels is performed. Rotation is done for 30 degrees and a scale of 0.25 is used. Training is done using a cloud GPU on a 12 GB NVIDIA TitanX GPU with a learning rate of 0.00025 over 300 epochs for about 4 days. Every Convolutional layer is clubbed with a batch normalization layer and a ReLU to enhance the training process. The channel size for these layers was set as 256. RMSProp optimizer is used with an epsilon of 1e-8 and the alpha set as 0.99.

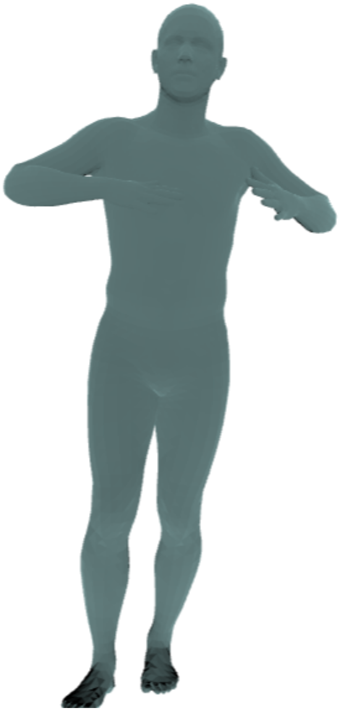
V. Results and Discussions

Video stabilization was performed by first detecting key points using FAST and then applying Euclidean translation for smoothening the displacement in x and y directions in each frame as shown in Fig. 5. This stabilization offered better results compared to using raw video with jitters by improving key point estimation and helping in acquiring a stable non-shaky sequential pose visualization. The resultant JSON file after processing the was converted to .csv for ease of data access and visualization. Fig. 6. shows how Akima was able to predict the nature of the missing point more dynamically than linear interpolation. Thus, the Akima interpolation was chosen and the results against the originally estimated pose by OpenPose and predicted key-points after applying linear and Akima interpolation for three frames with missing joints are tabulated for comparison in Fig. 7.



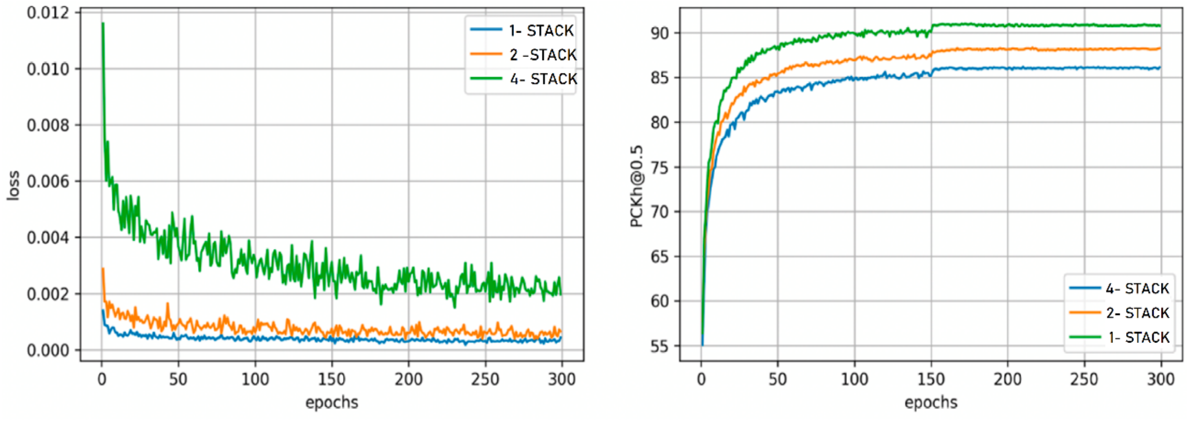
Resulting 3D pose with confidence score

This completed pose framework provides a suitable dataset for further phases of the project of text to sign language. By using these 2D key-points, 3D pose estimation and subsequently helps in obtaining skeletal animation for a 3D model that would output signs for a corresponding word.



3D mesh obtained

Increasing the stacking of the model shows a slight increase in its performance. From 1-stack at 86.5% to 4-stack at 88.2%, the model shows a decent training evaluation. The loss metric used for this model is PCK. PCK (percentage of correct key points) of the model performance is shown in the Figure. 4-stack clearly performs much better than the other two variants where more than 90% of the predicted key points fall under the 0.5 threshold from the true key points. This performance showcases high reliability of the network towards non annotated videos. The 3D key points obtained are then rendered using a SMPL model to produce a mesh for better representation.



(a) (b)

The graph suggests the best suitability of 4- stack hourglass.

VI. Conclusion

At present, since there are no available public datasets of Indian Sign Language, we cannot directly compare our results with previous results in detail. We thus provide a comparison in Fig. 5. showing how the pose estimation improves with the suggested processing techniques as opposed to the pose estimation results obtained on original raw videos. We propose Akima interpolation which proves to be a highly light-weight method to improve the Open Pose's output. In this paper a highly low-cost method compared to all the neural models built for handling missing joints and occlusions is proposed to estimate missing 2D key points from the input RGB videos. This method was run on over 2000 YouTube videos and showed appreciable performance. It is also one of the swiftest and most efficient ways for Indian Sign Language to pose data acquisition which has no previous works dedicated towards it.

Key-points obtained were transformed and sent through the network. This paper showcases the performance of Hourglass network on 3D pose detection of Indian sign language signers. The output obtained is a collection of .bvh files that make animation in future interfaces easy. With this document, we aim to provide an expandable dynamic database for Indian sign language.

VII. References

[1] Disabled Persons in India - A statistical Profile 2016 http://mospi.nic.in/sites/default/files/publication_reports/Disabled_persons_in_India_2016.pdf

[2] Havasi, L., & Szabo, H. M. (2005). A Motion Capture System for Sign Language Synthesis: Overview and Related Issues. EUROCON 2005 - The International Conference on "Computer as a Tool." doi:10.1109/eurcon.2005.1629959

[3] Bugra Tekin, Pablo Márquez-Neila, Mathieu Salzmann, and Pascal Fua. 2017. Learning to fuse 2d and 3d image cues for monocular body pose estimation. In Proceedings of the IEEE International Conference on Computer Vision. 3941–3950.

[4]Tomas Simon, Hanbyul Joo, Iain Matthews, and Yaser Sheikh. 2017. Hand keypoint detection in single images using multiview bootstrapping. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 1145–1153.

[5]Camillo Lugaresi, Jiuqiang Tang, Hadon Nash, Chris McClanahan, Esha Uboweja, Michael Hays, Fan Zhang, Chuo-Ling Chang, Ming Guang Yong, Juhyun Lee, et al. 2019. MediaPipe: A framework for building perception pipelines. arXiv preprint arXiv:1906.08172 (2019).

[6]F. Zhang, X. Zhu, H. Dai, M. Ye, and C. Zhu, "DistributionAware Coordinate Representation for Human Pose Estimation," CVPR, vol. abs/1910.06278, 2019.

[7]S Ertürk, TJ Dennis, Image sequence stabilization based on DFT filtering. IEE Proc. Vision Image Signal Process. 147(2), 95–102 (2000).

[8]C Song, H Zhao, W Jing, Y Bi, in Proc. Intl. Conf. Pattern Recognition. Robust video stabilization based on bounded path planning, (2012).

[9]M Grundmann, V Kwatra, I Essa, in Proc. IEEE Conf. Computer Vision and Pattern Recognition. Auto-directed video stabilization with robust l1 optimal camera paths, (2011).

[10]Sijin Li and Antoni B. Chan. 2014. 3d human pose estimation from monocular images with deep convolutional neural network. In Proceedings of the Asian Conference on Computer Vision. Springer,332–347. http://visal.cs.cityu.edu.hk/static/pubs/conf/accv14-3dposecnn.pdf

[11]Julieta Martinez, Rayat Hossain, Javier Romero, and James J. Little. 2017. A simple yet effective baseline for 3d human pose estimation. In Proceedings of the IEEE International Conference on Computer Vision. 2640–2649. https://openaccess.thecvf.com/content_ICCV_2017/papers/Martinez_A_Simple_yet_ICCV_2017_paper.pdf

[12]Dushyant Mehta, Srinath Sridhar, Oleksandr Sotnychenko, Helge Rhodin, Mohammad Shafiei, Hans-Peter Seidel, Weipeng Xu, Dan Casas, and Christian Theobalt. 2017. Vnect: Real-time 3d human pose estimation with a single rgb camera. ACM Trans. Graph. 36, 4 (2017), 44. https://dl.acm.org/doi/10.1145/3072959.3073596