Simulating Hand Gesture Data by Translating a Random Sample of Moving Hand

Arpit Goel[1]

[1] U.G. student, Computer Science and Engineering Department,

at Thapar University, Patiala, Punjab, India

agoel2_be22@thapar.edu

Abstract: Data gathering can be a challenge while training artificial intelligence models for new tasks like sign language interpretation. In this case, there is limited data available for Indian Sign Language and it is therefore very difficult to build an interpreter for it. This technique hopes to reduce that challenge by allowing simulation of new data based on present data and some random sample of hands. This is done by using a random sample with a small variance in gestures, on basis of which changes can be made to the gathered data to simulate new training data built around the gathered data having the complexity in movements of the random sample.

Keywords: Artificial intelligence; Data generation; Simulation; Hand gesture; mediapipe; opencv

1. Introduction: The development of Artificial Intelligence is largely dependent on data collection and the usability of the collected data. Most models are trained on publicly available data sets which aren't very difficult to access. Alternatively, models are also trained on gathered data which is either publicly available and can be converted into a

figure 1. 1

dataset or data gathered by other methods specifically for the model. The latter is usually, a very time, money and resource intensive task. One such case is seen when attempting to acquire data for Indian Sign Language (ISL) to Text conversion. The limited number of resources on the internet include one example for every phrase or alphabet in the vocabulary. This paper tries to reduce this challenge; faced when gathering such types of data. This approach hopes to translate patterns followed by landmarks during subtle movements on a hand into gathered data samples in hopes of creating new samples such that these patterns are reproduced. This is a more space friendly, low complexity alternative to fine tuning pretrained models on said examples.

Feeding an image of a open hand like shown below (figure 1.1) will output a set of 21 landmarks labelled as P_i where $i \in [0,20]$ (figure 1.2).

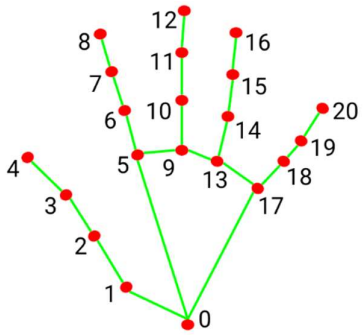

figure 1. 2

We may consider P_0 to be the position of the hand. i.e. in a two-dimensional coordinate system, P_0 indicated the coordinates in the form of (x_0, y_0) depicting where the hand is, referring to the bottom left corner of an image as (0,0). We may move the hand towards any direction (without affecting the rotation of the hand) by calculating the change in its P_0 coordinates represented by ΔP_0 and then adding

the difference to all the remaining points. Using this point of reference, we enable ourselves to exploit other patterns that occur when hands move.

3. Method: To simulate examples based on movements, the determination of the type of translation needed to carry the process out. Transformations used can either be spatial; involving movement of hands while maintaining gesture in 3 dimensions or gestural; indicating any change in the gesture or sign made by the hand.

Consider a hand given by the set $H_0 = \{j_i \mid j_i = (x_{0,i}, y_{0,i}, z_{0,i}) \wedge i \in [0,20]\}$ placed at $j_0 = (x_{0,0}, y_{0,0}, z_{0,0}) \in H_0$ in a three-dimensional cubical space of L_0 x B_0 x H_0 units. The two-dimensional notation of this set may be represented by the vector $p_i = (x_{0,i}, y_{0,i})$ where $x_{0,i}, y_{0,i} \in H_0$.

Consider another hand $H_1 = \{k_i \mid k_i = (x_{1,i}, y_{1,i}, z_{1,i}) \wedge i \in [0,20]\}$ placed at $k_0 = (x_{1,0}, y_{1,0}, z_{1,0}) \in H_0$ in a three-dimensional cubical space of L_1 x B_1 x H_1 units at any time-stamp t_0. The two-dimensional notation of this set may be represented by the vector $q_i = (x_{1,i}, y_{1,i})$ where $x_{1,i}, y_{1,i} \in H_1$.

H_0 is our base example and H_1 is the simulator. Any changes measured in the simulator occurred during the time of observation, are to be translated into the base case, therefore recreating the movements of H_1 in the hand H_0. We do this by calculating the change in position of each point in the set H_1 and then normalising it. This would be constructed into a 1x21 matrix $T(t_0, t_n)$ which is the translation of H_1 from time-stamp t_0 to any time-stamp in the simulator assuming, that the gesture doesn't change all throughout the time of observation.

$$\Delta\vec{k_i} = \overrightarrow{k_{i,t_n}} - \overrightarrow{k_{i,t_0}}$$

This is the difference in positions of i^{th} point between the two time-stamps. This is calculated for all 21 vectors in the set and stored in a 1x21 matrix $K_{t0,tn}$

$$K_{t0,tn} = \begin{matrix} \Delta\vec{k}_0 \\ \vdots \\ \Delta\vec{k}_{20} \end{matrix}$$

The transformation T comes from the following normalisation:

$$T_{0,n} = K_{t0,tn} \times N_1$$

The matrix N_1 is the normalisation matrix thich is diagonal matrix of size 21x21 which containing the normalisation factors for each of these vectors that are given by:

$$\vec{n} = \begin{matrix} n_0 \\ \vdots \\ n_{20} \end{matrix}$$

Where,

$$n_i = \left|\vec{k}_{i,t_0} - \vec{k}_{j,t_0}\right| \exists\, j = i-1 \;\forall\; i \in (1,2,3,4,6,7,8,10,11,12,14,15,16,18,19,20),\; j = 0 \;\forall\; i \in (5,9,13,17)$$

And,

$$n_0 = l_1 \times b_1$$

$$N_1 = \left[1/x_{ij} \,\middle|\, x_{ij} = 0 \forall i \neq j, x_{ij} = n_i \;\forall\, i = j \;\exists\, n_i \in \vec{n}\right] \exists\; 20 \geq i, j \geq 0$$

$$N_1 = \begin{bmatrix} 1/n_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1/n_{20} \end{bmatrix}$$

Which means, N_1 is the diagonal matrix of elements in $\vec{n}$.

The result of these calculations is the matrix $T_{0,n}$, which contains 21 vectors with three dimensions each representing the changes in each of the corresponding vectors. These changes may be simulated into the base example by denormalization of these values and summing the denormalized values with a matrix of all vector points in H_0.

$$H = J + \left(T_{0,n} \times D_0\right)$$

Where,

$$J = \begin{matrix} j_0 \\ \vdots \\ j_{20} \end{matrix}$$

$$d_j = |j_i - j_l| \;\exists j = i-1 \;\forall\; i \in (1,2,3,4,6,7,8,10,11,12,14,15,16,18,19,20),\; j = 0 \;\forall\; i \in (5,9,13,17)$$

And,

$$d_0 = l_0 \times b_0$$

D_0 is the diagonal matrix of the values d_j

$$D_0 = \begin{bmatrix} d_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_{20} \end{bmatrix}$$

H is a new sample of hand gesture data. The movement of H_0 from the base example and the new

sample is the same as that of H_1 from time t_0 to time t_n.

This technique can be used in two dimensions by substituting all k vectors with q vectors and all j vectors with p vectors. There also can be different methods to normalize the differences by either ignoring the position of the hand, (the zeroth point) in all calculations and operating on only 20 points to avoid shifting the position of hand in base example. It is important to note that this technique works very well when the movement captured can be approximated to linear motion without much curvature in trajectories, as seen in any space and is being translated into a space of similar dimension.

4 Implementation: The experiment performed for this method utilizer the variations of normalisation techniques mentioned in the previous section of this article. The base example is from an Indian Sign Language dataset representing the ISL symbol for the number

3.1: Spatial Transformations: are classified for use in the type of simulation involving the movement of hands from one place to another, or the rotation of hands, or scaling. This is possible because the translations of points in the three-dimensional space can easily be translated into two dimensions by simply dropping the third dimension. The working principle used here is demonstrated as follows: