ISSN No:-2456-2165

Translating Indian Sign Language to Text using Deep Learning

Dr. P. Sivakumar
Assistant Professor (Sr. Gr)
PSG College of Technology
Coimbatore, Tamilnadu

Amrithaa I S, Sandhiya A., Janani T., Karthikeyani S
B.Tech Information Technology
PSG College of Technology
Coimbatore, Tamilnadu

Abstract:- Indian Sign Language (ISL) could be a complete language with its own descriptive linguistics, syntax, vocabulary and a number of other distinctive linguistic attributes. The globe is hardly live while not communication, notwithstanding whether or not it's within the variety of texture, voice or visual expression. The communication among the deaf and dumb folks is carried by text and visual expressions. Gestural communication is usually within the scope of confidential and secure communication. Hands and facial elements are vastly powerful to precise the thoughts of human in confidential communication. Signing is learned by deaf and dumb, and frequently it's not known to traditional folks, thus it becomes a challenge for communication between a standard and hearing impaired person. Its strike to our mind to bridge the gap between hearing impaired and traditional folks to create the communication easier. Signing Language Recognition (SLR) system takes associate input expression from the hearing impaired person offers output to the traditional person within the kind text or voice. The system is split into 3 main parts: the system style, the dataset, and therefore the deep learning model coaching and analysis.

Keywords:- *Indian Sign Language (ISL); Sign Language Recognition (SLR); Deep Learning;*

I. INTRODUCTION

A. INDIAN SIGN LANGUAGE

Communication be a very important activity of people at large to measure, as they'll specific their feeling, encourage cooperation and social bond, share their plan, and work along in society through communication solely. Those who don't seem to be ready to hear or speak (hearing-impaired people) uses linguistic communication as a mean of communication. Like voice communication, linguistic communication conjointly emerges and evolves naturally inside hard-of-hearing persons. It's a visible variety of communication and in every country/region, wherever the hard-of-hearing community exists, this linguistic communication grows severally from the native voice communication of the region. Therefore linguistic communication from every region has distinctive syntax and structure, with one common property that it's perceived visually. Round the world, countries have their linguistic communication. For example: "American Sign Language (ASL), British Sign Language (BSL), Australian Sign Language (Auslan), French Sign Language (FSL), Indian Sign Language (ISL) and many more". ISL is mainly used in India, a country that has a large population of hearing-impaired people. ISL is a language comprises of fingerspelling gestures, single hand gestures, double hand gestures, facial expressions and body movement. In contrast to other prevailing sign languages, ISL has complex representation as it mostly uses double hand shape for sign gestures.

B. SIGN LANGUAGE RECOGNITION

Sign Language Recognition could be a procedure task that involves recognizing actions from sign languages. This can be a necessary drawback to unravel particularly within the digital world to bridge the communication gap by folks with hearing impairments. Finding the matter sometimes needs not solely annotated color (RGB) knowledge, however varied different modalities like depth, sensory data, etc. also are helpful. The ISLR conjointly referred to as word-level SLR is that the task of recognizing individual signs or tokens referred to as glosses from a given section of sign language video clip. This can be ordinarily seen as a classification drawback once recognizing from isolated videos, however needs different things like video segmentation to be handled once used for period applications. In CSLR conjointly referred to as language transcription, given a symbol language sequence, the task is to predict all the signs or glosses within the video. This can be additional appropriate for real-world transcription of sign languages. Betting on however it's resolved, it may generally be seen as associate degree extension to the ISLR task. Language translation refers to the matter of translating a sequence of signs to any needed speech. It's usually sculptured as associate degree extension to the CSLR drawback.

C. DEEP LEARNING

Deep learning is an Artificial Intelligence (AI) operate that imitates the workings of the human brain in process information and making patterns to be used in higher cognitive process. Deep learning may be a set of machine learning in computing that has networks capable of learning unsupervised from information that's unstructured or unlabeled. It's additionally called deep neural learning or deep neural network. Deep learning has evolved hand-in-hand with the digital era, that's has led to AI explosion of information altogether forms and from each region of the globe. This data, celebrated merely as huge information, is drawn from resources like social media, net search engines, e-commerce platforms, and on-line cinemas, among others.

This monumental quantity of information is quickly accessible and might be shared through fine tech applications like cloud computing. Recovering accuracy with deep learning algorithms is either because of a much better Neural Network, a lot of procedure power or vast amounts of information. The recent breakthroughs within the development of algorithms are largely because of creating

them run a lot of quicker than before, that makes it attainable to use a lot of and a lot of knowledge. The foremost vital distinction between deep learning and ancient machine learning is its performance because the scale of information will increase. This is often as a result of deep learning algorithms want an oversized quantity of information to know it dead. But lately, Deep Learning is gaining a lot of quality because of its mastery in terms of accuracy once trained with vast amounts of information. The computer code trade now-a-days moving towards machine intelligence.

D. LONG SHORT TERM MEMORY

After the primary Convolutional Neural Network (CNN) based mostly design (AlexNet) that won the ImageNet LSTMs deal with both Long Term Memory (LTM) and Short Term Memory (STM) and for making the calculations simple and effective it uses the concept of gates. Forget Gate is that the LTM goes to forget gate and it forgets information that is not useful. Learn Gate is that the Event (current input) and STM are combined together so that necessary information that we have recently learned from STM can be applied to the current input. Remember Gate is that the LTM information that we haven't forget and STM and Event are combined together in Remember gate which works as updated LTM. Use Gate also uses LTM, STM, and Event to predict the output of the current event which works as an updated STM.

LSTMs deal with both Long Term Memory (LTM) and Short Term Memory (STM) and for making the calculations simple and effective it uses the concept of gates. Forget Gate is that the LTM goes to forget gate and it forgets information that is not useful. Learn Gate is that the Event (current input) and STM are combined together so that necessary information that we have recently learned from STM can be applied to the current input. Remember Gate is that the LTM information that we haven't forget and STM and Event are combined together in Remember gate which works as updated LTM. Use Gate also uses LTM, STM, and Event to predict the output of the current event which works as an updated STM.

E. MOBILENET

The MobileNet model is designed to be used in mobile applications, and it is TensorFlow's first mobile computer vision model. MobileNet uses depthwise separable convolutions. It significantly reduces the number of parameters when compared to the network with regular convolutions with the same depth in the nets. This results in lightweight deep neural networks. A depthwise separable convolution is made from two operations. Depthwise convolution. Pointwise convolution. MobileNet is a class of CNN that was open-sourced by Google, and therefore, this gives us an excellent starting point for training our classifiers that are insanely small and insanely fast.

The MobileNet model is intended to be employed in mobile applications, and it's TensorFlow's 1st mobile pc vision model. MobileNet uses depthwise severable convolutions. It considerably reduces the quantity of parameters in comparison to the network with regular convolutions with a similar depth within the nets. This ends up in light-weight deep neural networks. A depthwise severable convolution is formed from 2 operations. Depthwise convolution. Pointwise convolution. MobileNet could be a category of CNN that was open-sourced by Google, and thus, this offers U.S.A. a wonderful start line for coaching our classifiers that area unit insanely little and insanely quick.

II. PROPOSED SYSTEM

A. OBJECTIVE

- Sign language to text conversion model are currently in development and are getting popular day by day with advancement of deep learning.
- The objective of this project to create a model which can identify Indian sign language and converts user action to words.
- The main objective of this project is to be able to use this model in real life.
- Sign language is learned by deaf and dumb people, and usually it is not known to normal people, so it becomes a challenge for communication between a normal and dumb person.
- It strikes our mind to bridge the gap between dumb and normal people to make the communication easier.
- Sign language recognition (SLR) system takes an input expression from the dumb person gives output to the normal person in the form text or voice.

B. DATA COLLECTION

The data are collected manually. In this key points are effectively going to form the frame values. First a path is set to the exported data which is the numpy arrays. There are 1000 plus actions that are used to detect. There will be 30 videos for each action which is of 30 frames in length. Many folders created for storing the data which are recorded manually.

C. DATA ANALYTICS

The key point value for training and testing have been collected. For this first a medapipe model is created. Here 3 looping condition is performed. First a loop is performed through actions then a loop is performed through sequence of videos. Then a loop is performed for length of the sequence. Then the values are read and the detection is done. A new wait logic is performed. Before collecting each video, a text stating starting collection and a message stating Collecting frames of this video is displayed. After extraction of key points for all 30 video of each action it ends automatically.

The overall work flow of the proposed system is shown below:

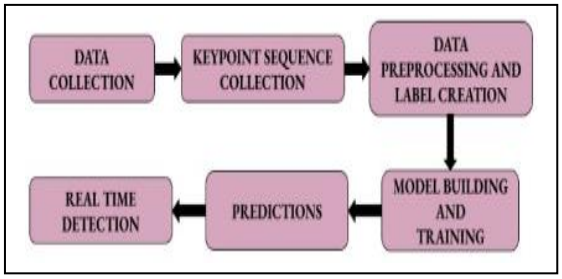

Fig. 1: Overall work flow of the proposed system.

ISSN No:-2456-2165

The above Fig.1 represents the flowchart of the model. First the dataset was collected manually. Keypoints were extracted and the sequence of keypoints were collected. Then the data was pre-processed and a label was created for each data. A model was built and it was trained. At last the results were predicted in a live stream mode.

III. IMPLEMENTATION AND RESULT ANALYSIS

A. LONG SHORT TERM MEMORY MODEL

a) KEYPOINTS USING MP HOLISTIC

Import all the packages needed for the video. Define the Holistic Model and draw the utilities. Using the medapipe function to convert the colour from BGR to RGB and make predictions. And again convert the colour from RGB to BGR. After that draw face, pose, left hand and right hand connections. Define the necessary variables for these landmarks.

Video Capture () function is used for accessing the webcam. Set the initial detection confidence and tracking confidence as 0.5. Cap. Read () is used to read the current frame from the webcam and CV2.imshow () is used to show that to the user. Wait key is used for waiting and if the 'q' is pressed it will stop capturing the video and used to close the CV feed.

b) EXTRACTING KEYPOINT VALUES

The features such as pose landmark, right hand landmarks, left hand landmarks, face landmarks were extracted in a way that to concatenate these into a numpy array. Using flatten every landmarks were converted into one big array because it wanted to be in a particular manner. These were combined to do the sign language prediction. The keypoints value for training and testing have been collected. For this first a medipipe model was created. Here 3 looping condition was performed. First a loop was performed through actions then a loop was performed through sequence of videos. Then a loop was performed for length of the sequence. Then the values are read and the detection was done. A new wait logic was performed. Before collecting each video, a text stating starting collection and a message stating Collecting frames of this video was displayed. After extraction of keypoints for all 30 video of each action it ends automatically.

c) DATASET COLLECTION

All the data collected as video from live stream mode are stored in MP_Data folder. Inside that 1000 plus actions were stored. Each action have 30 videos which was of 30 frames in length.

Fig. 2: Dataset collected for training

Fig.2 were the folders created for storing the data which were recorded manually.

d) DATA PREPROCESSING

For this some dependencies were imported. Next a label map was created to represent each one of the different actions. In this two array was created one is sequences which represents the x data and the next one is the label which represents the y data. Then it was looped through videos and then the sequence of frames and a blank array called window is created which going to represent the different frames for the particular sequence. Then using np.load to load the frame which goes through 4 paths.

e) MODEL BUILDING

This was the LSTM network and for this the tensorflow and keras was used. First the dependencies were imported and a log directory was created and tensorboard callbacks was set. Then the neural network model was created. Here there were 3 layers of LSTM and 3 layers of fully connected dense layer. In model summary there were 3 LSTM layers, 3 dense layers, Total number of parameters and trainable and non-trainable parameters.

```
model.summary()

Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_6 (LSTM)                (None, 30, 64)            442112
_________________________________________________________________
lstm_7 (LSTM)                (None, 30, 128)           98816
_________________________________________________________________
lstm_8 (LSTM)                (None, 64)                49408
_________________________________________________________________
dense_6 (Dense)              (None, 64)                4160
_________________________________________________________________
dense_7 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_8 (Dense)              (None, 3)                 99
=================================================================
Total params: 596,675
Trainable params: 596,675
Non-trainable params: 0
```

Fig. 3: Model Summary

Fig.3 shows the summary of the LSTM model. Here use the layer type as LSTM and dense, output shape as none and declared the parameters.

f) TRAINING THE LSTM MODEL

In training phase the model was complied. It passes to x train, y train specified to epoch equal to 2000 and a call back was set.

```
model.compile(optimizer='Adam', loss='categorical_crossentropy', metrics=['categorical_accuracy'])

model.fit(X_train, y_train, epochs=2000, callbacks=[tb_callback])

Epoch 1/2000
2/2 [==============================] - 11s 2s/step - loss: 1.2179 - categorical_accuracy: 0.3333
Epoch 2/2000
2/2 [==============================] - 1s 287ms/step - loss: 3.9508 - categorical_accuracy: 0.3158
Epoch 3/2000
2/2 [==============================] - 1s 326ms/step - loss: 21.1879 - categorical_accuracy: 0.3509
Epoch 4/2000
2/2 [==============================] - 1s 305ms/step - loss: 7.0661 - categorical_accuracy: 0.3333
Epoch 5/2000
2/2 [==============================] - 1s 292ms/step - loss: 7.4311 - categorical_accuracy: 0.3509
Epoch 6/2000
```

Fig. 4: Training Result

The Fig.4 shows that after training, run the epochs and calculate the loss and categorical accuracy.

g) EVALUATION METRICS

Using the model.predict() function to predict the model and a accuracy of the model was near to 95%.

h) TEST IN REAL TIME

At last the testing phase, in that the threshold was declared as 0.8. A medipipe model was set. The feed was read and the detection was done. Then the landmarks were drawn. Variables such as colour, font, size of the words were set and the values had been set. After the prediction it breaks gracefully. When the result of np.argmax(res) is greater than the threshold, the words of action will be printed on the screen as a s

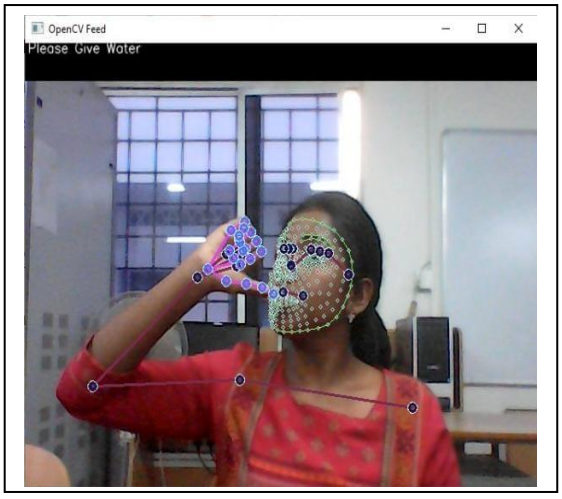

Fig. 5: Testing

Fig. 5 predicts the output of deaf and dumb people's communication with normal people in the screen.

B. MOBILENET

a) COLLECTING IMAGES

Import the main packages and libraries. Declare the IMAGE_PATH directory link and the sign action words. VideoCapture() function is used for accessing the webcam. Give the sleep time as 5. cap.read() is used to read the current frame from the webcam and CV2.imshow() is used to show that to the user. Wait key is used for waiting and if the 'q' is pressed it will stop capturing the video and used to close the CV feed.

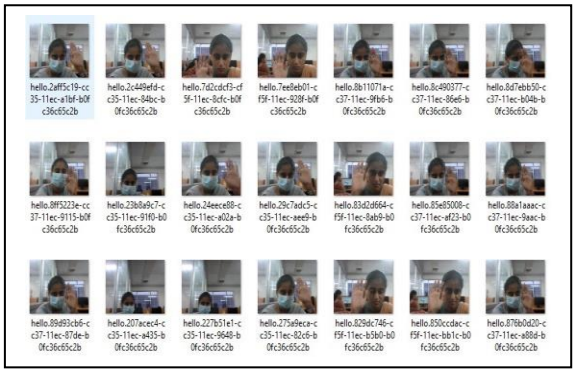

Fig. 6: Dataset Collection

Here the Fig.6 shows that the sign action is recorded as many images and frames.

b) DATA PREPROCESSING

The path had been set to the corresponding directories. Create a label map in which a label was declared. Then a Tensorflow Record was created. A tensorflow model and the pre-trained models from tensorflow model zoo. Then the model was. Cloned. Copy the model configuration to the training folder and the model was updated for transfer learning.

c) TRAIN THE MODEL

The model was trained. For this the dependencies were imported. Load the pipeline configuration and a detection model was built. Then the checkpoint was restored. Fig.7 explains the result of training.

```
python Tensorflow/models/research/object_detection/model_main_tf2.py --model_dir=Tensorflow/workspace/models/my_ssd_mobnet --pi
peline_config_path=Tensorflow/workspace/models/my_ssd_mobnet/pipeline.config --num_train_steps=5000
```

Fig. 7: Training Result

d) DETECT IN REAL TIME

Here the dependencies are imported. The values were assigned to the category index in the annotation path and the feed was released. The capture was setup using the video capture function and the width and height of the video had been captured. The detection classes had been declared in ints. A wait key was performed and the detection was done in the real time. Here in this model we got the accuracy as 92.1%.

C. COMPARISON OF THREE MODELS

In this study, an assessment of state-of-the-art pre-trained models for the task of prediction of sign language using images was done. The objective of this research was to compare the Cnn, LSTM and MobileNet models by evaluating the accuracy and validation loss. All of the models showed a statistically significant performance. Starting with the accuracy metric CNN had the lower results with 89%, followed by MobileNet with 92% and LSTM with 96% representing an almost excellent classification. On the other hand, in the validation loss metric, LSTM obtained the lower result with 2.96%, followed by MobileNet with 3.32% and CNN with 4.01%. Indeed, as is shown, all of them achieved statistically significant performance in each measure, but the LSTM implementation achieved the highest percentage.

MODEL NAME	ACCURACY %	VALIDATION LOSS
LSTM	96.58	2.96
MOBILENET	92.1	3.32
CNN	89.2	4.01

Table 1: Performance measure(%) for pre-trained model

D. HISTOGRAM FOR THREE MODELS

a) ACCURACY

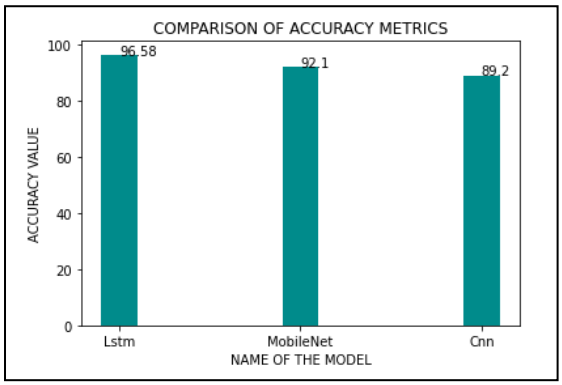

Fig. 7: Accuracy of three models

b) VALIDATION LOSS

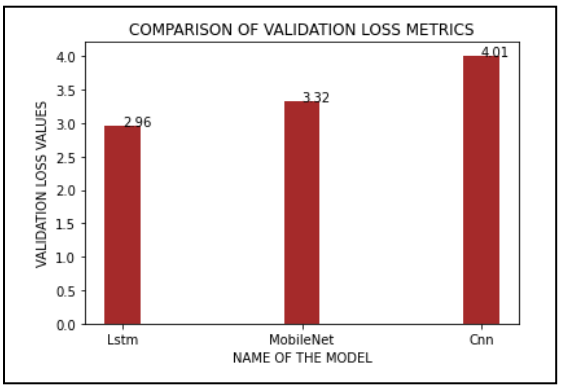

Fig. 7: Validation loss of three models

IV. CONCLUSION AND FUTURE ENHANCEMENT

Indian language may be a complete language with its own descriptive linguistics, syntax, vocabulary and a number of other distinctive linguistic attributes. It's employed by over five million deaf folks in Asian country.

We obtain to involve the DHH community all told our choices. By conducting surveys and interviewing deaf participants, we have a tendency to learn that:

The DHH community faces the biggest challenges at the geographic point, with fifty six of our survey participant news difficulties in act with workplace colleagues. Geographic point difficulties rank over the other scenario - like banks, whereas looking, at restaurants, with family etc.

Communication is that the biggest challenge for the deaf community at work. By conducting interviews with deaf participants, we have a tendency to learn that they realize it troublesome to speak to their hearing managers and colleagues. The community is especially solely ready to ask associate degree interpreter or to different deaf colleagues at work, and this prevents them from having the ability to voice considerations concerning serious work problems like appraisals, promotions, pay hikes and different work-related matters.

In future, aim to build an AI powered translator,that allows the DHH community to continue conversing in ISL and enables the hearing community to understand them easily. With the rise of video calling applications and work-from-home (78% of our survey respondents work from home), we believe that integrating a sign language interpreter into video calling apps like Microsoft Teams should should ease the problem of workplace communication, and level the playing field for the DHH community at work.

REFERENCES

[1.] Ashok Kumar and Sahoo, “Indian Sign Language Recognition using Machine Learning Techniques,” 17 June 2021.
[2.] Aditya Das, Shantanu Gawde, Khyati Suratwala, Dr. Dhananjay Kalbande, “Facialexpression recognition from video sequences: temporal and static modelling. Computer Vision and ImageUndertaking”,Feburary 2018.Atabay, H. A. (2017). Deep residual learning for tomato plant leaf disease identification. J. Theor. Appl. Inform. Technol. 95, 6800–6808.
[3.] Zafar Ahmed Ansari and Gaurav Harit, “Nearest Neighbour Classification of Indian Sign Language Gestures using Kinect Camera”,in Sadhana, Vol. 41, No. 2, February 2018.
[4.] KumudTripathi, Neha Baranwal, G. C. Nandi,“ Continuous dynamic Indian Sign Language gesture recognition with invariant backgrounds”, 2016 Conference on Advances in Computing, Communicationsand Informatics (ICACCI).
[5.] S. Tamura and S. Kawasaki, “Recognition of Sign Language Motion Images In Pattern Recognition”, volume 21, 2019.

[6.] Vogler, C. & Metaxas.D, “A framework for recognizing the simultaneous aspects of American Sign Language”, 2020.
[7.] Garcia, B., & Viesca, S. A., “Real-time American Sign Language Recognition with Convolutional Neural Networks”, 2020.
[8.] Starner, T., Weaver, J., & Pentland, A., “Real-time american sign language recognition using desk and wearable computer based video”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.
[9.] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. “Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks. ACM Transactions on Graphics”, 33(5):1–10, Sept. 2018.
[10.] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, “Joint training of a convolutional network and a graphical model for human pose estimation”, pages 1799–1807, 2019.
[11.] Toshev and C. Szegedy, “Deeppose: Human pose estimation via deep neural networks”, pages 1653–1660, 2018.
[12.] J. Zhang, J. Jiao, M. Chen, L. Qu, X. Xu, and Q. Yang.,”3d Hand Pose Tracking and Estimation Using Stereo Matching”, arXiv preprint arXiv:1610.07214, 2016.
[13.] R. Zhao, Y. Wang, and A. Martinez,” A Simple, Fast and Highly-Accurate Algorithm to Recover 3d Shape from 2d Landmarks on a Single Image”, arXiv preprint arXiv: 1609.09058, 2017.
[14.] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei,”Model based deep hand pose estimation”, pages 2421–2427, 2016.
[15.] D. Tome, C. Russell, and L. Agapito,” Lifting from the Deep: Convolutional 3d Pose Estimation from a Single Image”. arXiv preprint arXiv:1701.00295, 2017.
[16.] N. Sarafianos, B. Boteanu, B. Ionescu, and I. A. Kakadiaris,”3d Human pose estimation: A review of the literature and analysis of covariates”, 2016.
[17.] T. Sharp, Y. Wei, D. Freedman, P. Kohli, E. Krupka,A. Fitzgibbon, S. Izadi, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, and A. Vinnikov,” Accurate, Robust, and Flexible Real-time Hand Tracking”, pages 3633–3642. ACM Press, 2017.
[18.] Shih-En Wei, Varun Ramakrishna, Takeo Kanade, and Yaser Sheikh,”Convolutional pose machines”, pages 4724–4732, 2016.
[19.] M. Oberweger, P. Wohlhart, and V. Lepetit,” Hands deep in deep learning for hand pose estimation”, arXiv preprint arXiv: 1502.06807,2019.
[20.] M. Oberweger, P. Wohlhart, and V. Lepetit,”Training a feedback loop for hand pose estimation”, pages 3316–3324,2017.