Hand Gesture Based Human-Robot Interactions

Arpit Bahety, Ankur Dengla, G.C. Nandi

Robotics and Machine Intelligence Laboratory
Indian Institute of Information Technology, Allahabad
Allahabad, India
Email: iit2015089@iiita.ac.in, gcnandi@iiita.ac.in

***Abstract*— Human-robot interaction is a vast and ever developing field of robotics and AI. In this paper we showcase an implementation of hand gesture recognition in real time by the Nao robot for interaction with humans. The gestures being recognized are not static images but videos i.e. the movement of the hand with respect to time is also taken into consideration (Example - Swiping left, zooming in, etc). The problem also incorporates the fact that the environment in which the gestures are being performed is not constrained by any factors i.e. the domain for this system is hand gesture videos captured by any standard webcam. The major contribution of this paper is (1) Training a 3D Convolutional Neural Network for the task of hand gesture classification task. (2) Integrating this trained model with Nao robot for real time gesture recognition. (3) Tackling the problem of limited computing power of Nao by creation of an interactive pipeline between Nao and an external system performing all the computations.**

I. INTRODUCTION

The Nao robot has the following modules in-built as mentioned in its documentation -

- Motion
  - Omnidirectional walking
  - Whole body motion
  - Fall Manager
- Vision
  - Track, learn, and recognize images and faces
- Audio
  - Voice recognition and text-to-speech capabilities
  - Sound Source Localization

But the Nao robot still lacks a gesture based communication module. This is exactly what this paper intends to implement.

A gesture can be defined as any physical movement, large or small, by a person that is intended for communicating a specific message or instruction. In other words, gestures are mostly non-verbal means of communicating. According to this simple definition, we can say that hand gestures are gestures performed by only using the hands. The problem of hand gesture recognition basically involves designing a system that can recognize the gestures and react accordingly. This is an important and popular problem because of the applications that it can have such as in VR Systems, surgery using robots, and sign language interpretation systems to name a few.

The underlying algorithm for gesture recognition is a 3D convolutional neural network (3D CNN)[1] model (which will be discussed in detail in section II.A). The model is trained using 20bn Jester dataset [1], which is a dataset created by large number of crowd workers. By using video gestures instead of images, we also add a temporal dimension to the problem. This facilitates recognizing a much wider range of gestures. Also, the aim is to be able to use this system irrespective of the background i.e. at any location. Since the 20bn dataset is made up of videos collected from the community, it has a variety of backgrounds, lighting and other varying factors.

Due to the shortcomings of the Nao processor, the model cannot be loaded onto the robot itself for prediction. Thus, we developed a communication pipeline between the Nao robot and the neural network model. (this is further discussed in section II.C)

After an effective model and the pipeline was obtained, the entire module was integrated with Nao robot for the purpose of hand gesture controlled communication with it. This communication refers to controlling the movement of the robot with the help of hand gestures. This hand gesture based communication, if achieved effectively, can be developed further to enable people without any knowledge of programming to communicate with the Nao robot remotely and thus, help towards the deployment of such robots for a much wider range of applications than they are currently being used for.

II. RELATED WORK

The motivation for the idea of using 3d CNN for the task of gesture recognition for Nao came from the work of Maryam et al. in their paper "Deep Learning for Action and Gesture Recognition in Image Sequences: A Survey".[2] It is a survey of current deep learning based methodologies for action and gesture recognition in sequences of images. They have done a comparitive study of four types of models for the task of action and gesture recognition. The four types of models are 2D models, motion-based input model, 3D models, and temporal models. They've also argued that 3D networks over a long sequence are able to learn more complex temporal patterns which is why 3D CNN was chosen by us.

[1] 20BN Jester Dataset: https://20bn.com/datasets/jester

Another paper by Raveesh et al. titled "Integration of Gestures and Speech in HumanRobot Interaction"[3] talks about incorporating a module in Nao to enhance the interaction abilities of the Nao humanoid robot by extending its communicative behavior with non-verbal gestures. This paper shows us how with the limited computation power of Nao processor, incorporating external modules to enhance performance of Nao is possible.

Ningbo Yu et. al. in their paper[4] have proposed a gesture-based telemanipulation scheme to control the Nao humanoid robot using Leap Motion Contoroller. The Leap Motion Controller[5], introduces a new gesture and positon tracking system with declared sub-millimeteraccuracy, and a frame rate of up to 200 fps. They have successfully telemanipulated the Nao Robot for Ground Walking which includes going forward or backward, turning left or right with the full body, and rotating clockwise or anticlockwise only with the the upper body. But the drawback of leap motion controller is the hassle of an external device. The solution that we are trying to provide for gesture recognition in Nao is without the use of any external heavy device that needs to be carried around.

Ajili Insaf in their paper "Gesture Recognition for Humanoid Robot Teleoperation" [6]have worked on an online gesture recognition system for natural and intuitive communication between Human and Nao robot using the Laban Movement Analysis technique to describe high level gestures for Nao teleoperation. The Laban Movement Analysis (LMA) is one of the most famous theories of body expressions that was originally developped by a dancer, architect, choreographer and painter, Rudolf Laban[7]. Seeing the challenges of gesture based communication, here we proposed a technique for communicating with humanoid robots using video gestures with the help of a 3-D convolutional neural network for video recognition and integrating this model with Nao robot for action performance in real time which has significantly low computational resources. Our system based on 3-D CNN is characterized by robustness, accuracy and reliability.

III. METHODOLOGY

This section consists of 3 parts. The first part talks about the model used for the classification task. The dataset used to train the model is described in the second part and the third part discusses about the communication pipeline between the Nao robot and the model.

A. Model

Before introducing the architecture of the model we would like to discuss about the pre-processing of the input to the model as this step is necessary to get best results. Frames of the input video at discrete intervals of time are taken, where each frame is saved as a 2-D image of size 176 x 100. Each image is cropped at the center to output an image of size 84 x 84. 18 frames of each video are stacked sequentially along the depth to output a 3D tensor of shape 84 x 84 x 18. This 3D tensor is normalized using mean 0 and standard deviation of 1. This processed input is then fed to the 3D CNN.

The architecture of the 3D Convolutional Neural Network consists of four convolutional layers followed by a 3 layered fully connected neural network. The details of the architecture can be seen in Fig. 1

B. Dataset

The dataset being used for the purpose of this project is the 20BN-Jester dataset [2] which comprises of labeled videos of humans performing pre-defined actions. The dataset was collected from the community. The actions were performed in front of a standard computer webcam. There are video samples spanning 27 classes. We used only 8 classes out of the 27 for this project. The classes chosen were the ones relevant to the actions that were needed to be performed. Each sample is a set of frames (JPG images) obtained from the video submitted at 12 frames per second. The height of each image is 100 pixels with variable width from 150 - 176 pixels. There are total 43,797 training samples and 5144 validation samples (with labels).

C. Communication Pipeline

The specifications of Nao are as follows:

- ATOM Z530 1.6 GHz CPU
- 1 GB RAM
- 2 GB Flash memory
- 8 GB Micro SDHC

The main challenge with Nao was the limited computation power available on it. Thus, loading the neural network on the robot was out of the question as the 1 GB RAM available on it would be exhausted by allocating the weights of the network. Also the relatively low processing power of Nao would lead to a lot of latency in the prediction. Hence, there was a need to perform all the computations of predicting the gesture externally. This created the need of establishing a communication pipeline between Nao and the model. The pipeline is described in Fig.2

- The first step of the pipeline was to get the live feed from the Nao camera to the server to be processed. This was done via the PyQt4[3] python library. Even though Nao can capture images at 30 fps, the server was limited by processing power and thus, the practically obtained fps on the server is low. The feed is kQQVGA (160*120 pixels).
- In the second step, this feed obtained at server is pre-processed according to to preprocessing methodology talked about in section III.A to generate the input for the model. The server provides the model with the input and the output of the model is the recognized gesture label.
- The third step sends the action corresponding to the label recognized by the model to Nao using Naoqi API [4]. Nao performs the action and is then ready for the

[2]20BN Jester Dataset: https://20bn.com/datasets/jester
[3]PyQt4: https://www.riverbankcomputing.com/software/pyqt/
[4]Naoqi API: http://doc.aldebaran.com/2-1/Naoqi/index.html

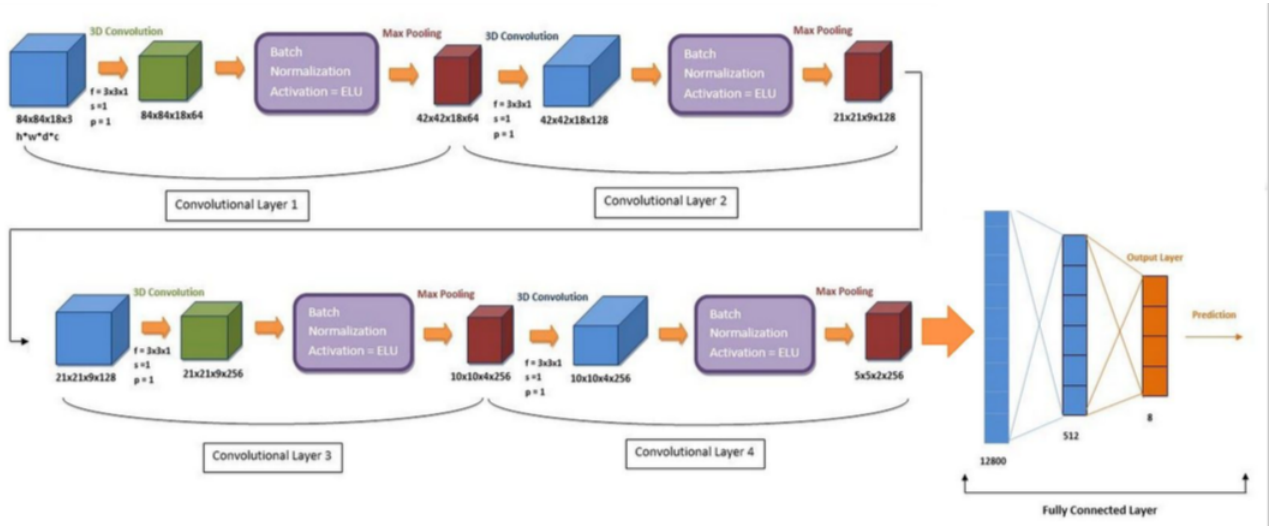

Fig. 1. Architecture of 3D CNN

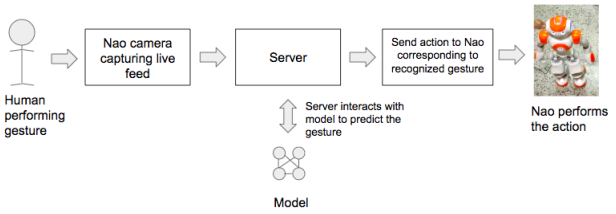

Fig. 2. Communication pipeline between Nao and the model

next gesture. The algorithm for the entire process is described in Algorithm 1. The output 'action' is sent to Nao and then Nao performs the action. A list of actions corresponding to the label is shown in the Table I

IV. RESULTS

The results of the present research were impressive with high training, validation and testing accuracy. We have divided the results into two subsections for clarity. The first being Training Model Results and the second Testing Model Results.

A. Training Model Results

The training process took a total of 5 hours for 11 epochs. It was trained on a Nvidia 1080Ti GPU with 10 GB RAM. A learning rate of 0.001 was used with categorical cross-entropy loss function and stochastic gradient descent as the optimizer to update the weights. The top 1[8] and top 3 accuracy values are shown in Table II. The top 1 and top 3 validation values are shown in Table III. A maximum of 98.898 % top 1 training accuracy and 97.975 % validation accuracy was achieved. A low bias is achieved as the training accuracy is quite high. Also generalization is good as well as the validation accuracy is high too, thus a significantly low variance is achieved. Therefore, it can be inferred that we have achieved a good bias-variance tradeoff.

Total number of frames of video, N = 18
Input : $\{f_k \mid k = 1, 2..., N\}$
where f_i is the i^{th} frame
shape(f_i) = 176 x 100

Output: action

F ← **new** tensor;
for $i \leftarrow 1$ **to** N **do**
  $f_i \leftarrow$ *84 x 84 pixels cropped from center of* f_i;
  F.append(f_i*)*;
end
F ← Normalize F using mean 0 and standard deviation 1;
filterSize ← 3 x 3 x 1;
stride ← 1;
padding ← 1;
activationFunction ← ELU;
$F^{'}$ ← Pass F through the four convolutional layer
label ← Pass $F^{'}$ through the fully connected layer
action ← **map**(label)

Algorithm 1:

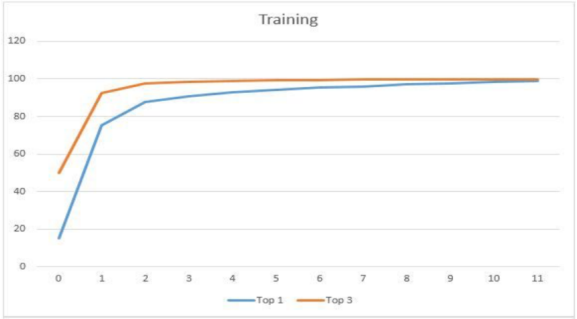

Fig. 3. Training Top 1 and Top 3 Accuracy Graph (Number of Epochs vs Accuracy)

TABLE I

ACTIONS BY NAO CORRESPONDING TO THE LABEL RECOGNIZED BY MODEL

Label	Action by Nao
No Gesture	Do nothing
Swiping two fingers up	Stand Up
Swiping two fingers down	Sit Down
Pulling two fingers in	Move Forward
Push hand back	Move Backward
Swiping two fingers left	Move left
Swiping two fingers right	Move right
Shaking hand	Say Hi

TABLE II

TRAINING TOP 1 AND TOP 3 ACCURACY

Epoch	Top 1 (Accuracy)	Top 3 (Accuracy)	Loss
0	15.000	50.000	2.0500
1	75.129	92.330	0.7332
2	87.750	97.561	0.3862
3	90.880	98.357	0.2862
4	92.715	98.810	0.2275
5	94.119	99.202	0.1775
6	95.348	99.488	0.1430
7	96.012	99.580	0.0970
8	97.237	99.766	0.0824
9	97.547	99.857	0.0713
10	98.538	99.909	0.0478
11	98.898	99.968	0.0363

B. *Testing Model Results*

The two major attributes that define the effectiveness of this project are:

- Testing Accuracy: Percentage of correct outcomes
- Latency: Time gap (in seconds) between completion of gesture by human and start of corresponding action by Nao

A total of 50 tests were done. Of these 48 times Nao responded with the expected action. i.e.
Total number of outcomes = 50
Number of correct outcomes = 48
Testing Accuracy Achieved = 96 %
Average Latency Achieved = 1.3 seconds

Fig. 5, 6, 7, 8 show some of the results of our experimentation.

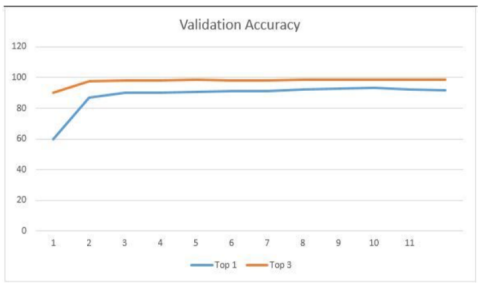

Fig. 4. Validation Top 1 and Top 3 Accuracy Graph (Number of Epochs vs Accuracy)

TABLE III

VALIDATION TOP 1 AND TOP 3 ACCURACY

Epoch	Top 1 (Accuracy)	Top 3 (Accuracy)	Loss
0	60.000	90.000	0.9927
1	86.779	97.104	0.4020
2	89.908	97.688	0.3179
3	90.989	98.341	0.3180
4	91.790	98.096	0.3040
5	92.968	98.556	0.3240
6	93.444	98.060	0.3280
7	94.103	98.065	0.3170
8	94.900	98.430	0.2693
9	95.145	98.776	0.2869
10	97.975	98.504	0.3082
11	96.784	98.803	0.3800

Fig. 5 shows the ”Swiping two fingers up” gesture being performed by the human subject in front of Nao. Fig. 6 shows the corresponding action taken by Nao (standing up) after successful recognition of the hand gesture performed. Similarly, Fig. 7 shows another hand gesture, ”Swiping two fingers left” being performed. The result after successful recognition is that Nao turns to the human’s left (i.e. Nao’s right). The same is shown in Fig. 8.

Fig. 5. Swiping two fingers up

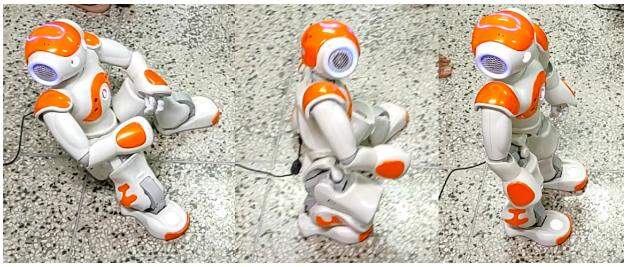

Fig. 6. Nao standing up

V. CONCLUSIONS

The important contribution of this paper is the introduction of a gesture recognition module in Nao using a novel approach of integrating a 3D convolutional neural network with an external server by the communication pipeline mentioned in Fig. 2. The gesture recognition module is important to

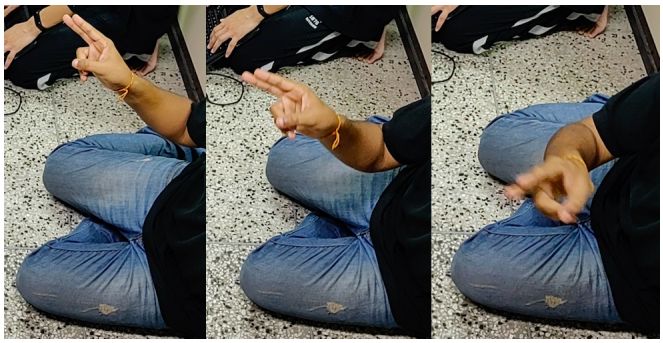

Fig. 7. Swiping two fingers left

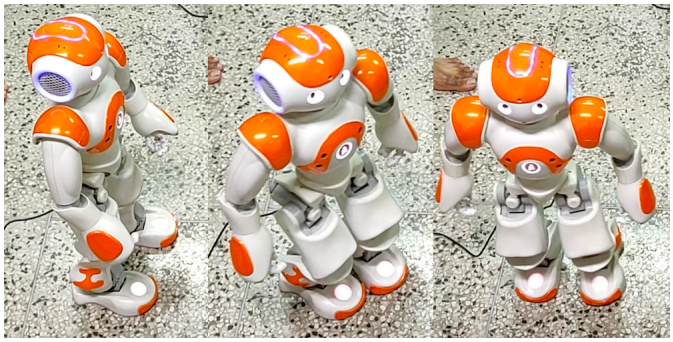

Fig. 8. Nao turning to the human's left (Nao's right)

Nao or for that matter, to any robot, for making the robot more interactive, for social robots and possible for the tasks of surgery using robots, and sign language interpretation systems to name a few. An important conclusion drawn from this research is that even though Nao is limited by its computation power, an efficient and effective system can be developed using an external server that enables human-robot interaction using hand gestures. The latency levels achieved by this system are encouraging but still not the best that can be achieved. Further research work into developing a lighter neural network architecture with the same level of accuracy can make the communication process a lot more efficient.

Once a good balance is obtained between the accuracy and the latency of the system, it can be deployed for further applications with Nao having the capability of handling more number of gestures.

REFERENCES

[1] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ser. ICCV '15. Washington, DC, USA: IEEE Computer Society, 2015, pp. 4489–4497. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2015.510

[2] M. Asadi, A. Claps, M. Bellantonio, H. J. Escalante, V. Ponce-Lpez, X. Bar, I. Guyon, S. Kasaei, and S. Escalera, *Deep Learning for Action and Gesture Recognition in Image Sequences: A Survey*, 07 2017, pp. 539–578.

[3] R. Meena, K. Jokinen, and G. Wilcock, "Integration of gestures and speech in human-robot interaction," in *2012 IEEE 3rd International Conference on Cognitive Infocommunications (CogInfoCom)*, Dec 2012, pp. 673–678.

[4] N. Yu, C. Xu, K. Wang, Z. Yang, and J. Liu, "Gesture-based telemanipulation of a humanoid robot for home service tasks," 06 2015.

[5] L. Motion. Leap motion controller. [Online]. Available: http://www.leapmotion.com

[6] I. Ajili, M. Mallem, and J. Didier, "Gesture recognition for humanoid robot teleoperation," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Aug 2017, pp. 1115–1120.

[7] R. P. Tsachor and T. Shafir, "A somatic movement approach to fostering emotional resiliency through laban movement analysis," *Frontiers in Human Neuroscience*, vol. 11, p. 410, 2017. [Online]. Available: https://www.frontiersin.org/article/10.3389/fnhum.2017.00410

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.