Journal of Scientific & Industrial Research
Vol. 80, July 2021, pp. 563-569

VISU: A 3D Printed Functional Robot for Human Pose Replication

Pranav Kompally[1,2], Sibi Chakkaravarthy Sethuraman[1,2]*, Srikar Reddy[1] and Charan Koduri[3], Yaswanth Naidu[3], M Raghavaiah[3], Sashidhar Reddy[3], Namburi Nikhil[3]

[1]Artificial Intelligence and Robotics (AIR) Research Centre, [2]School of Computer Science and Engineering, [3]School of Mechanical Engineering, VIT-AP University

Received 04 July 2020; revised 10 May 2021; accepted 15 June 2021

This paper presents VISU, a novel 3D printed functional robot. VISU is equipped with open-source technologies making it more modular in adapting Internet of Things (IoT) based services. VISU is able to detect and analyze the user's activity and pose. In addition, a simple method to replicate the pose of a user is also proposed. VISU can also perform actions such as Face recognition, Object Recognition among other basic functionalities.

Keyword: Computer vision, Face recognition, Humanoid, Object recognition, Pose replication

Introduction

The project commenced with an idea to create a robot that can work like an appliance that includes applications of knowledge from various domains. To create a machine that is modular and well-equipped with IoT (Internet-of-things) technologies. And a platform of open-source based robots[2] like InMoov[1] to experiment with design.[3,4] VISU has its parts to satisfy the placement of electronics. Altogether, the robot contains over 200 parts that were printed using in-house printers at VIT- AP University with the total time taking over 300 hours. VISU is printed in ABS (Acrylonitrile butadiene styrene). Weighing nearly 19 kgs and equipped with over 25 servos motors, the robot is fully functional. Centrally connected by an Atmega328 microcontroller and activated using Raspberry Pi 3, the robot performs a set of movements inspired by humans.

For remote based operation, VISU is supported on open-source and modular technologies such as Arduino and raspberry pi support IoT technology. One of the eye plates is fitted with a Microsoft LifeCam with a 1080p resolution for high quality input. The recorded media is sent to Raspberry pi, which is the main hub to compute functions such as human pose estimation, face recognition, object recognition etc. The light-weight structure allows smooth motion of many moving parts like biceps, hands, abdomen and head. These movable parts are powered by Hitec Servo motors. Typically, this kind of servos allows only 180 degrees of movement. In order to provide continuous motion to the gears present in joints, the stopper in servos has been removed that restricts a complete rotation. The modification of the removing the stopper from the inner gears of the motors ensures a complete rotation. The microcontroller, Arduino Mega, assigns the angle to which the motors are to be rotated upon receiving certain instructions from raspberry pi.

Apart from performing basic movements of the parts, VISU is capable of replicating basic human poses using image recognition. The original InMoov utilized Microsoft Kinect for gesture and human pose recognition. However, to reduce expenditure, Posenet has been deployed to recognize poses taking input from VISU's camera. Pose replication using Kinect requires additional installment of hardware and configuration apart the added expenditure. Moreover, Kinect requires processing power efficient and robust than that of Raspberry pi. Initially, the program captures a 2 second clip using the camera and saves it in a form of images. Upon identifying a person, the VGG-16 algorithm applies a mask to the detected person and blacks out the remaining area. This helps in better recognition of humans and thus increases accuracy. Once the pose is captured taking reference from a pre-trained model, the 'x' and 'y' values are determined for every joint of the user present in the test video. The initial and final values of 'x' and 'y' are mapped to the initial and final angle positions of the servos respectively. Upon receiving the computed

*Author for Correspondence
E-Mail: chakkaravarthy.sibi@vitap.ac.in

angle between two joints, the respective servo situated in the position is triggered to move to the configured angle. This process is repeated for every position of servos such as head, elbow joints, hands. Thus, human pose replication is achieved. VISU is built with Face Recognition using Eigen face algorithm and image classification using VGG-16 neural network. The key contribution of this paper is given below.

- Deep learning model to detect and recognize objects.
- Pose replication using Computer Vision instead of Kinect.
- Fully 3d printed, advanced low cost humanoid robot.
- Shadow mode is a feature incorporated in VISU that helps in remotely accessing robot's controls with additional sensors such as LeapMotion and Kinect.

Materials and Methods

A. The Skeleton

1. Head

The head and eyes rotating servos are enclosed by 10 plates that form a skull like structure as seen in Fig.1. The inverted servo situated in the head[8,9] allows for 180 degrees of movement situated beside are the horizontal and vertical micro servos that control the movement of the eye plates resembling the action of that of humans as shown in Fig. 2 One of the eye plates is fitted with a Microsoft LifeCam camera.

The same is used as an input for functions such as Image processing discussed later in the paper. Situated below the eye plates is the jaw controlling micro servos that mimics the movement of the lower jaw of humans. Upon startup, all micro servos are initialized to 40 degrees.

2. Biceps

As seen in Fig. 3, two 'nine' plates joined by a common base also houses the gears and servos. The base holds a rotating disc gear and a shaft with gear teeth. On trigger, the servo continuously rotates the shaft gear which rotates the disc gear causing the base to move in prescribed direction. The continuous movement of servo is achieved by removing the stopper attached to the primary gear in the motor, thus, allowing complete rotation of the motor.

3. Hands

Hand and forearm are built on InMoov's design.[7] The forearm holds 5 motors placed closely together. The palm of VISU is attached to a 5 finger like structure that has three movable hinge joints as shown in Fig. 4. Each phalanx in fingers is tightly darned and fastened and connected to respective motors placed in the forearm using a braided fishing line. The motors used here are HS-805BB. As the motors rotate to a 180 degrees, the fingers close into the palm creating a fist like structure and when rotated to 90 degrees, fingers relax and form an open hand like structure.

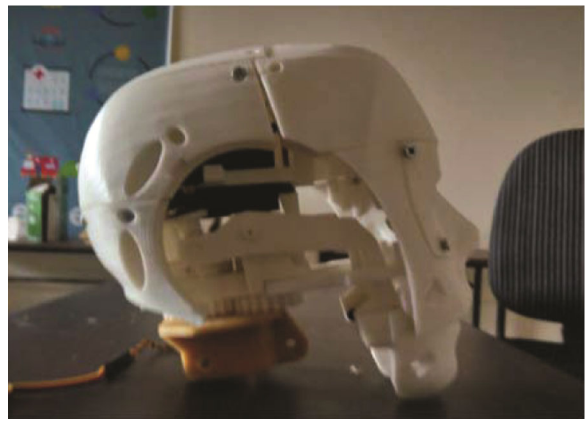

Fig. 1 — Side view of head

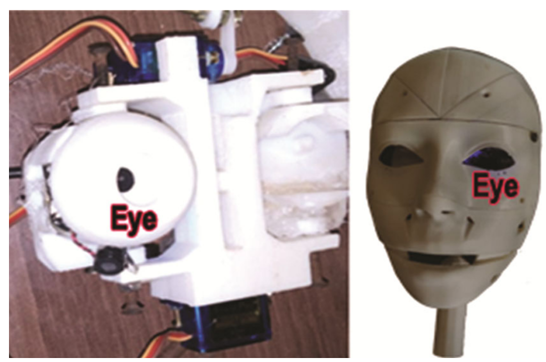

Fig. 2 — Camera placement on the eye plate

Fig 3 — Right and left biceps

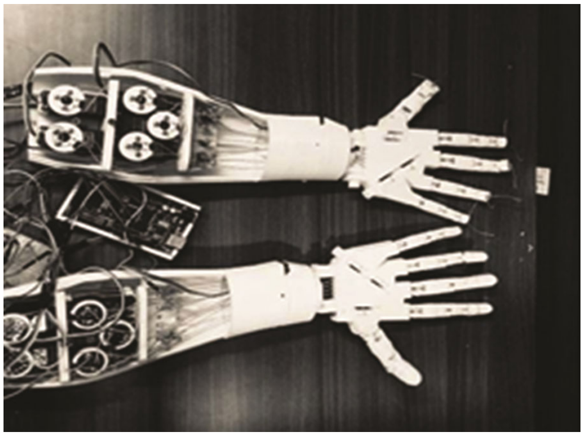

Fig. 4 — Hand structure of VISU

The forearm is attached to the bicep with another servo that performs the function of the elbow. The servo is placed on a shaft gear that pushes into the bicep working like a hinge joint. As the shaft gear rotates into the bicep, the forearm is lifted.

4. Abdomen

The head fits on to a neck like vertebrae that holds a servo which controls the up and down of the movement. The biceps fit on to the shoulder region. The shoulder base is powered by another servo that controls the vertical movement of the bicep. The abdomen rest (refer Fig. 5) on a rotating base. The rotating base consists of two disc gears that are controlled by two synchronous servo.

B. Hardware Architecture

The hardware architecture of VISU is shown in Fig. 6. All the motors and mechanisms that contribute to VISU are connected to a central hub, which holds the Atmega328 Microcontroller as Arduino Mega. The Arduino is uploaded with a set of over 200+ instructions on various movements. Signal pin of each motor is connected to a dedicated Digital Pin on the Mega. The ground pin of each motor is connected in parallel and linked into a single pin that leads to the negative terminal of the SMPS (Switched-mode-Power-supply) attached. The same is done for the power pin as well. All the power pins are connected in parallel and are lined into a single pin leading to the positive terminal of the SMPS. The SMPS used for the purpose is 50A and 6V rated. This provides enough supply to power up 25+ motors at the same time. The following diagram gives a brief idea about the wiring of the motor, Arduino and SMPS. The red, black and blue lines indicate positive, negative and signal lines respectively.

Fig. 5 — Torso fixed on top of abdomen

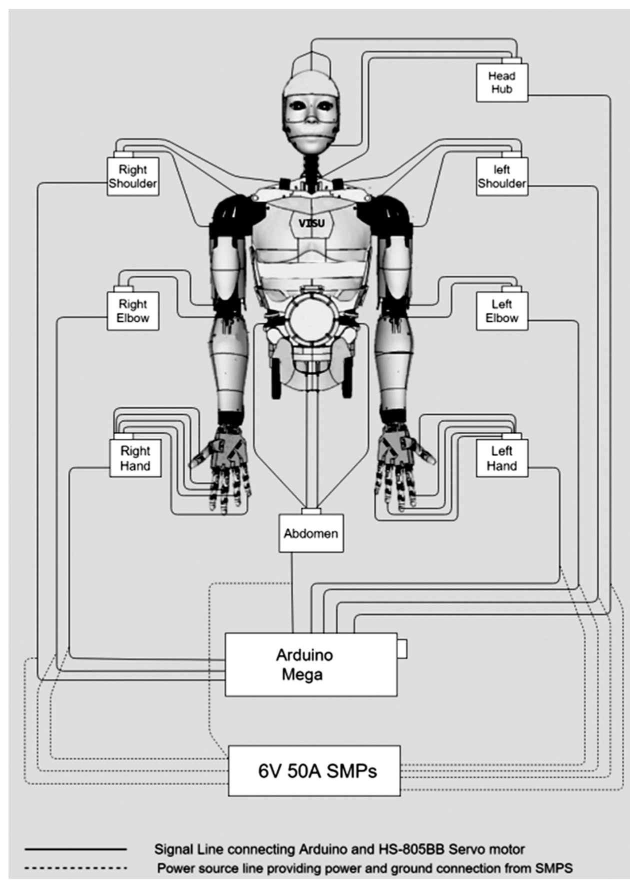

Fig. 6 — Hardware architecture

C. Software Architecture

The workflow of the software architecture is shown in Fig. 7. The software architecture is divided in four layers: The User Layer, Data Layer, Communication Layer and the Execution Layer. The User Layer governs the function of taking inputs either from the camera for image processing based application or from the VISU app and interface. The sole function of this layer is to capture input media and pass it on the Data Layer. The Data Layer holds both the microprocessor and microcontroller, which is Raspberry pi and Arduino Mega respectively. The Raspberry pi performs computations such as Face and object recognition, Crowd counting, human pose detection and IoT service processing. IoT services such as particular face tracking can be integrated using cloud based APIs. Additional IoT service includes i) Remote movement control of VISU over the internet; ii) Alerts the user on the mobile application when a particular face has been tracked and recognized; iii) A log of movement of every joint and its respective angle is pushed to the cloud; iv) Integration with third party web services such as IF This Then That (IFTTT) that allows the users to add in various IoT Services.

Further, all the services that involve the movement of servos are triggered by Arduino Mega. Upon receiving a request from raspberry pi, the Arduino Mega sends respective signals to the hubs in the Communication Layer.

The Communication Layer collects all the commands from the Data Layer and assigns them to the respective servo motors in the Execution Layer. The Execution Layer rotates the motors to the desired position.

1. Image Classification using VGG-16

VISU's main software and hardware controlling hub, Raspberry pi has a VGG-16 neural network-based model that can classify over a half-a-million objects. Unlike other image classification models such as ResNet50 and EfficientNet, which have heavy architecture and consume high processing power, VGG-16 is an optimized and real-time model that can run on minimal architecture. It is a prominent model for performing classification with least amount of processing power.[10,11] LifeCam positioned in the eye plate captures images of resolution 1080p. The images captured at 1080p yield results of higher accuracy. Without going much deeper into the model, brief information is provided. The input image is configured into a 224 × 224 image RGB image. As the name suggests, the model has 16 weights layers including 13 convolutional layers with filters of tiny receptive fields: - 3 × 3 and the convolutional stride is fixed to 1 pixel. All the convolutional layers are divided into 5 groups and each group is followed by a 2 × 2 max-pooling layer with a stride of 2 pixels. The final layer is a soft-max layer which normalizes its input value into a vector of values that follows a probability distribution whose sum totals to 1. The SoftMax function produces raw prediction values as real numbers ranging negative to positive infinity. All hidden layers are backed with the rectification ReLU (Rectified Linear Unit).

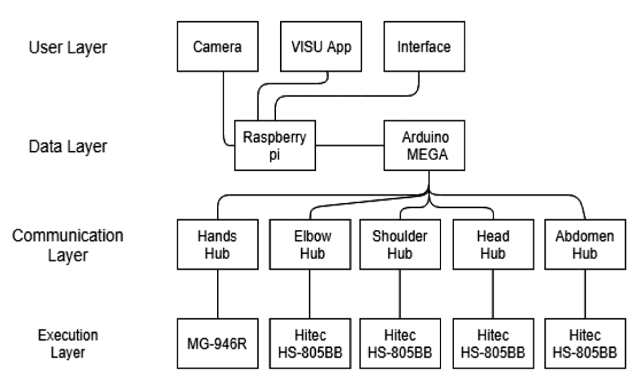

Fig. 7 — Software architecture

2. Face Recognition using Eigen Face Algorithm

Face Recognition is a highlight of VISU. Running on Raspbian OS, the program performing face recognition should be light and yet effective. There are many algorithms that yield great results, such as using Feed Forward Neural Networks (FFNW) algorithm[5], however, raspberry pi doesn't support them due to its computing limitations. To tackle the problem, face recognition using Eigen faces[6] has been used. Face recognition needs pre-defined data sets in order to specify the faces to be recorded. Training data sets of 3 distinct faces has been taken and stored in individual folders. Every dataset contains a minimum 20 images of an individual with a resolution of 480p each.[12]

To yield accurate results, the face is centered in the frame. Next, each folder is feeded to the Eigen face algorithm. Faces in the folder and extracted as T_1, T_2, T_3, . T_n. Each and every image T_i is transformed into a vector and placed in a training set 'S' as shown below in the Eq (1)

$$S = \{T_1, T_2, T_3, ...T_n\} \quad ... (1)$$

where n = number of images in the folders. Average face vector (Ψ) is obtained using the Eq (2)

$$\Psi = \frac{1}{M}\sum_{n=1}^{M} r_n \quad \ldots (2)$$

The average face vector obtained is subtracted from the original/training faces and the result is stored as a variable Φ_i (Eq 3).

$$\Phi_i = T_i - \Psi \quad \ldots (3)$$

Covariance matrix (C) is obtained using the Eq (4 & 5)

$$C = \frac{1}{M}\sum_{n=1}^{M} \Phi_n {\Phi_n}^T \quad \ldots (4)$$

$$C = AA^T \quad \ldots (5)$$

The above mentioned formula Eq. 4 can also be represented as $C = AA^T$ (N2 * N2 matrix where A is $\Phi_1, \Phi_2, \Phi_3 \ldots \Phi_m$)

The M eigenvectors obtained of $L = A^TA$ are used to find M eigenvectors μ_i $(Eq(6))$ of C that form the basis of Eigen face.

$$\mu_i = \frac{1}{M}\sum_{n=1}^{M} \nu_i \Phi_i \quad \ldots (6)$$

where, μ_i is the Eigenface and Eigenvectors,

Φ_i is the stored variable of average face vector subtracted from training faces and ν_i is the M eigenvector of $L = A^TA$

For better results, the eigenvector with the highest eigenvalues are used and the ones with low eigenvalues are omitted. The above said process is taken for every image in all folders and classified with the person/face name. The algorithm for face recognition is adapted from Sibi Chakkaravarthy et al., (2021) and given below.[12]

1. Raspberry pi captures a video clip of 2 seconds.
2. The video clip is then broken down into individual frames or images and saved in a folder specified for testing
3. Five frames are selected and the remaining frames are deleted from the system's memory.
4. Feature vector is obtained for all the 5 frames.
5. This gives a reconstructed face which is compared to the pre-processed faces.
6. If the similarity of equal to or above 0.86 or 86%, it accepts the input as a recognized user.

Due to its computing and storage limitations, raspberry pi can recognize upto 5–6 faces. The above algorithm is implemented in python and can be activated. After recognizing a face, the programs terminates and trigger audio clips saved in the pi that read out a welcome message as shown below:

"Hello user, it's nice to see you"

This message can be altered and can be mapped to different faces.

3. Pose Detection using Posenet

Posenet use has two phases where the captured image using the camera is fed through a convolutional neural network. To achieve decent accuracy and consume optimal computational resources, the output stride is set to 16 with its width resolution, height and key points being 15, 15 and 17 respectively. The pose decoding algorithm generates a heat map and offset vectors of all the joints. Posenet nearly generates over 17 key points that detect the positions of various parts of the body ranging from the head to the ankle through the hip. The offset vectors are generated to detect the exact location of the key points. These offset vectors return 'x' and 'y' coordinates that are added to the resulting heat map position of every joint to generate an exact value of the joint. This provides us with the final key point positions in a 2D space. The 'x' and 'y' coordinates are noted for every joint to measure the angle between them. Upon joining key points, a skeleton is obtained as shown in Fig. 8. The angle at the joints is calculated using the formula given in Eq. (7).

$$\theta = arccos\ arccos \frac{\underline{a}.\underline{b}}{||\underline{a}||\underline{b}||} \quad \ldots (7)$$

where θ is the angle at a joint & $\underline{a}.\underline{b}$ are the vectors of two lines that intersect at a particular keypoint and where arccos is the same as the inverse of the cosine ($\cos^{-1}$). Angle θ is obtained for every joint and stored in an array.

4. Pose Replication using the computed angles

Posenet was trained on Mobilenet. After extracting Key point heat map and offset vectors angles at every joint are computed for obtaining single pose estimation and stored as an array. The angles are passed on to the Arduino and assigned to the prescribed servo through their respective control hubs.

Results and Discussion

A. Image Classification

Before applying image classification, Fig. 9a and Fig. 9c show the test images and Fig 9b and Fig. 9d depicts the end result after classifying objects in indoor conditions. The end result of applying the VGG 16 algorithm in outdoor conditions is shown in Fig. 9e. The VGG-16 algorithm was tested both in indoor and outdoor conditions. In Indoor conditions, the algorithm performed well even in low light conditions and detected almost every object in the frame.

VISU can detect and recognize more than 100 objects. However, in outdoor conditions, the algorithm couldn't detect a few objects as shown in Fig. 9.

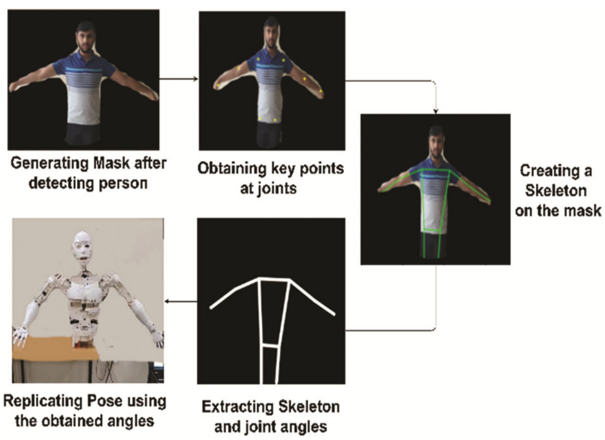

Fig. 8 — Pose replication process and result

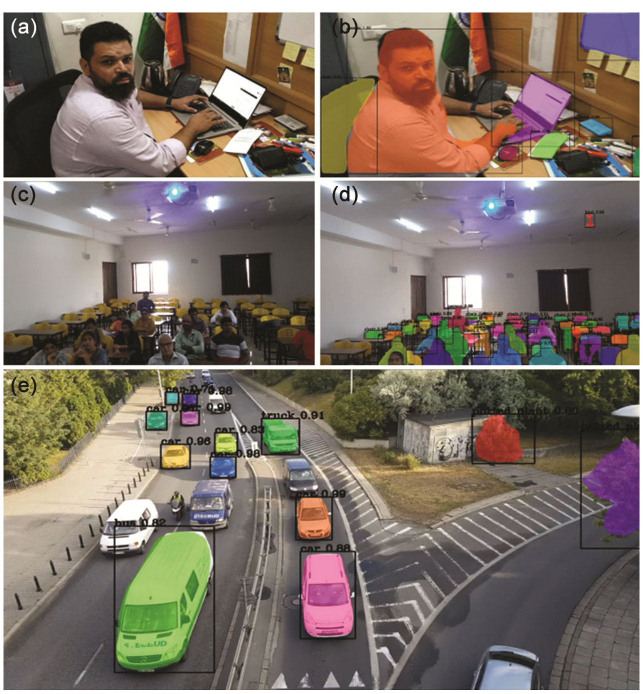

Fig. 9 — Screenshot of VISU image Classification

B. Pose Replication

The stages of pose replication are given in Fig. 8 and full view of VISU is shown in Fig. 10. VISU was able to replicate the pose from the captured image with an accuracy of 88%. However, with over 20 attempts it was noted that the person whose pose was being replicated has to stand between 2 feet to 3 feet from VISU's camera. The benefit developing mask of the person helped in yielding better pose skeleton and reduces computing time by 15%. In Table 1, the comparison of VISU with other platforms such as imNEU[3], El-Greco[11] and LARMbot[10] is done. Potentially, pose replication can be used to study posture and magnitude of the movement of muscles and joints as a part of limb rehabilitation. VISU when paired up with a user can replicate and provide insights on how each joint of the upper body is in motion. Thus, giving the medical instructor insight on development of muscles and limbs

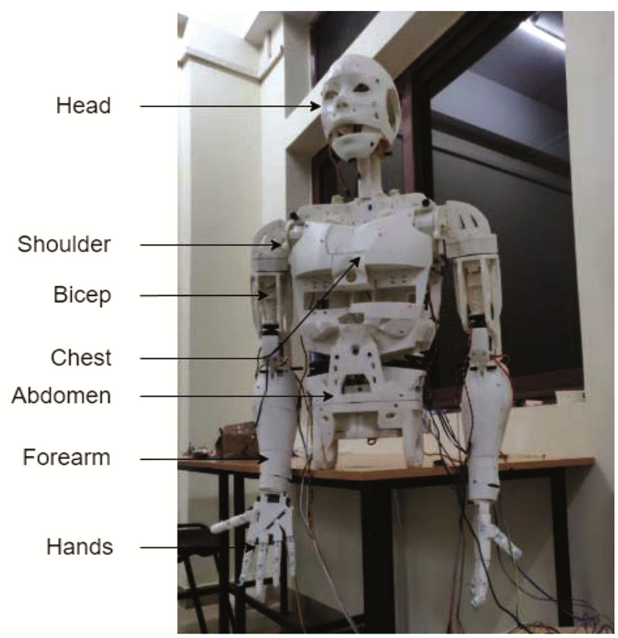

Fig. 10 — VISU at rest in AIR research center at VIT-AP University

Table 1 — Comparison of VISU with the existing humanoid platforms

	VISU	imNEU	El-Greco	LARMbot
Pose Replication	Yes.	No	No	No
Machine Learning on Edge capabilities	Yes.	No.	No	No
Bio-triggered movement	Yes	No.	No	No
Google AI-enabled	Yes.	No	No	No
OS	Raspbian	ROS and Ubuntu	Python CBD	Computer based interface
IoT	Yes	No	Yes	No
Shadow mode	Yes	No	No	No

Conclusion & Future Work

The obtained results make VISU a capable humanoid robot to deploy for crowd counting in both crowded and uncrowded places with minimal production cost. With the help of Face recognition, armed forces can identify and monitor people in public places. The Image Classification program in VISU helps in keeping a track of objects in its field of view in public places. Accurate Pose replication using computer vision reduces the use of Kinect from the original architecture reducing computation resources. Altogether, VISU is a robot that is the result of simple mechanisms, effective electronics, lightweight, and robust algorithms that perform a wide range of tasks. Modular in nature, VISU can be connected to wearable techs such as EEG and EMG to expand its capabilities.

Acknowledgement

We thank Dr S V Kota Reddy, Vice Chancellor, VITAP University for motivating and helping us. A special thanks to the team members of AIR Center, VIT-AP University. Also, an honourable mention and special thanks to Mr Pranav Vempati, CEO, Makers Hive for inspiring us in building the robot.

References

1 Liu S & Deng W, Vesry deep convolutional neural network based image classification using small training sample size, *3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, 2015.

2 Inmoov, "*An Open-source 3D printed Life size Robot*", Online resource, retrieved as on 23.06.2020.

3 Cheng H & Ji G, "Design and implementation of a low cost 3D printed humanoid robotic platform," *IEEE Intl Conf Cyb Tech Auto, Cont, and Intell Syst (CYBER)*, Chengdu, 2016, pp. 86–91.

4 Bergen B van, Toshev R, Heijster O, Sánchez A, Kuhar N, Obbergen B V, Ven J V D, http://eps.novia.fi/assets/Sidor/2/1545/InMoov-Humanoid-spring-2019-Final-report.pdf , 2019. Online resource, retrieved as on 22.06.2020.

5 Lu J, Zhao J & Cao F, Extended feed forward neural networks with random weights for face recognition, *Neurocomp*, **136** (2014) 96–102.

6 Ramadhani A L, Musa P & Wibowo E P, Human face recognition application using PCA and Eigen Face Approach, *Intl Conf Info Comp (ICIC)*, 2017.

7 Bulgarelli A, Toscana G, Russo L O, Farulla G A, Indaco M & Bona B, A low-cost open source 3D-printable dexterous anthropomorphic robotic hand with a parallel spherical joint wrist for sign languages reproduction, *Intl J Adv Robotic Syst*, **13(3)** (2016) 126.

8 Teixeira G & Roque A dos S, "Structure of a framework for the InMoov project in the development of IA and TA applications",https://www.researchgate.net/publication/328345594_Estruturacao_de_um_Framework_para_o_projeto_InMoov_em_prol_do_desenvolvimento_de_aplicacoes_de_IA_e_TA

9 Berra R, Setti F & Cristani M, "Berrick: a low-cost robotic head platform for human-robot interaction," *Intl Conf Sys Man Cyb (SMC)* (Bari, Italy) 2019, 559–566.

10 Cafolla D, Wang M, Carbone G, Ceccarelli M, LARMbot: A new humanoid robot with parallel mechanisms, in *Robot Design, Dynamics and Control* edited by V Parenti-Castelli &, W Schiehlen (ROMANSY 21 -. ROMANSY21 2016) Intl Cen Mech Sci (Courses and Lectures), vol 569, 2016.

11 Skoupras P, Soulountsi M, Khandelwal S, Upadhyay J, Paraskevas K, Zervas G, Fourtounis N, Chartomatsidis M, Katsios D, Papamichalakis G P & Kavallieratou E, El Greco: a 3d-printed humanoid that anybody can afford, *SETN '18*, SETN '18: Proc 10th Hellenic Conf Artif Intell, July 2018 Article No.: 58, Pages 1–2 https://doi.org/10.1145/3200947.3201062 2018.

12 Sethuraman S C, Kompally P & Reddy S, "VISU: A 3-D printed functional robot for crowd surveillance," in *IEEE Consum Electron Mag*, **10(1)** (2021)17–23.