Deep gesture interaction for augmented anatomy learning

Abstract

Augmented reality is very useful in medical education because of the problem of having body organs in a regular classroom. In this paper, we propose to apply augmented reality to improve the way of teaching in medical schools and institutes. We propose a novel convolutional neural network (CNN) for gesture recognition, which recognizes the human's gestures as a certain instruction. We use augmented reality technology for anatomy learning, which simulates the scenarios where students can learn Anatomy with HoloLens instead of rare specimens. We have used the mesh reconstruction to reconstruct the 3D specimens. A user interface featured augment reality has been designed which fits the common process of anatomy learning. To improve the interaction services, we have applied gestures as an input source and improve the accuracy of gestures recognition by an updated deep convolutional neural network. Our proposed learning method includes many separated train procedures using cloud computing. Each train model and its related inputs have been sent to our cloud and the results are returned to the server. The suggested cloud includes windows and android devices, which are able to install deep convolutional learning libraries. Compared with previous gesture recognition, our approach is not only more accurate but also has more potential for adding new gestures. Furthermore, we have shown that neural networks can be combined with augmented reality as a rising field, and the great potential of augmented reality and neural networks to be employed for medical learning and education systems.

Keywords: Neural network, augmented reality, 3D reconstruction, medical education, mobile cloud

1. Introduction

In pursuit of immersive human-machine interaction, researchers have explored the different interacting method with new input sources other than the traditional mouse and touchpad. In augmented reality, gesture control is often considered as an ideal interacting method, while leaving gesture recognition as a crucial problem to study [1, 2, 3, 4]. In contrast to the traditional mouse, the gestures to convey the same instruction are different in every person. Gestures are also a dynamic process so that the duration of the gestures cannot be fixed. Plus, except for the starting and ending positions, every other position in the duration must be detected and track. In the previous work on gesture recognition, gestures cannot be detected and tracked with satisfactory accuracy. Meanwhile, the neural networks have achieved many unprecedented results in deriving meaning and recognizing an object from complicated and vague time in various fields. With this remarkable advantage, neural networks can be used to extract patterns and detect the trends which used to be considered too hard for computers. Thus, a trained neural network can be used to analyze the gestures, such gestures are often more complicated comparing to click the mouse.

Successful and accurate gesture recognition can significantly improve the sense of immersion and user experience. It especially has great potential in medical teaching and learning where require high immersion to simulate the real environment of operations or anatomy courses. Heard from medical school, the professors and students are facing such dilemma that the number of anatomical specimens is limited while the number of students keeps increasing. Students have lacked the opportunity to have a close study of the specimens. Therefore building a three-dimensional object can help understanding which cannot require the textbook.

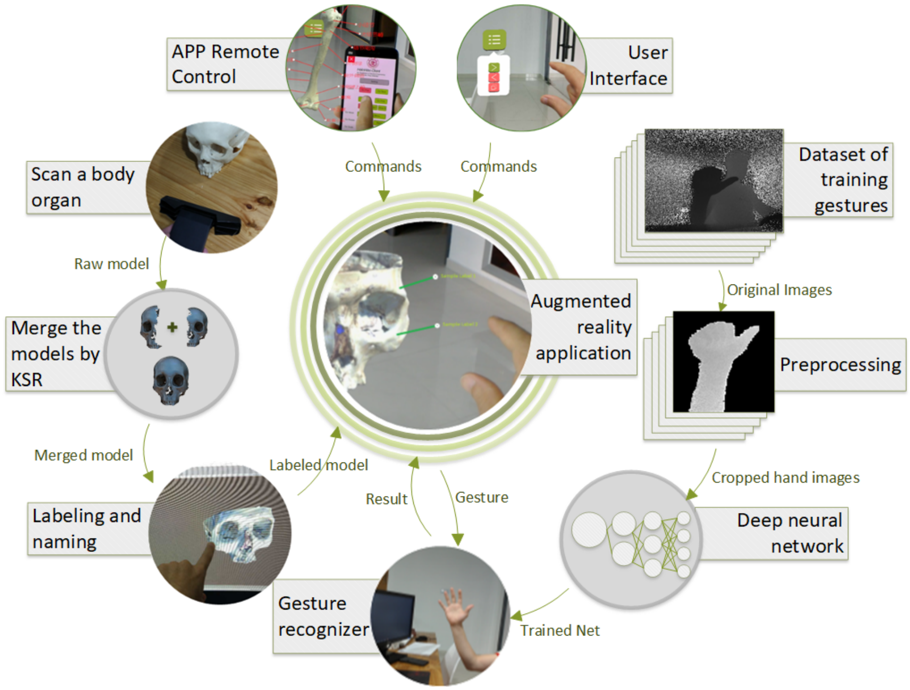


Figure 1: General idea to utilize interactive learning based on augmented reality glasses.

In this paper, we propose a user interface of augmented anatomy learning with gesture interaction based on the deep convolutional neural network. It offers the functions which teacher can use it to replace the common anatomy teaching process and students can use it to review anytime and anywhere. Also as a tutor, it can be the best choice, for example, medical students by listening to the recorded audio of their professors and using proposed application can learn anything without joining to those courses directly. Fig. 1. shows the application interfaces displaying human Humerus supporting labels on/off, scale, move, and rotate function by gestures controlling. Students can operate the application by Pan, Pinch, Fist, and Tap gestures which are recognized by a trained deep convolutional neural network. The networks include a 3D convolutional neural network to merge and analyze the information from the depth camera and RGB image. Furthermore, by using a cloud, the procedure of learning is getting faster. In this method, the learning process should divide into some separated server and the results returned back to the main server. Besides the gestures recognition, we have used the mesh reconstruction to 3D reconstruct the specimens. We scanned the model from different view angles and then merge and re-mesh the scanned meshes by the key points surface representation (KSR) algorithm.

We have shown that neural networks as a rising field can be applied to augmented reality for improvement. We also demonstrate the great potential of augmented reality and neural networks to be employed for medical and educational usage. Before 2012, people mostly use principal component analysis (PCA)to reduce dimension then feed to support vector machine (SVM) to recognize the hand gestures. After 2012, convolutional neural network (CNN) has become an important tool for object recognition since ImageNet of Krizhevsky *et al.* [5] have excelled results on the ILSVRC12 challenge. With high-performance GPUs, CNN's show great power on image recognition. Compared with other neural networks, CNN's take fewer parameters with better feature extraction quality which are easier for training. Our architecture contains 26 layers except for the Relu activation. The only sizes of filters here are 1x1 and 3x3. We alternately use these two kinds of filters. All convolutional layers are followed by fully-connected layers. The achievements made by this research include:

- To provide a low-cost and efficient way to reconstruct a 3D model from divided meshes. Scanning objects are done by the rangefinder camera. From the point, scanning of the whole part of a body organ is not possible or at least the quality will decrease. We have divided the scanning into small parts and merged them with KSR method;

- To provide a user-friendly interface, which meets the demands of medical education. The improvement of the user interface is not only gesture recognition, but also a new user interface is designed to interact with an operator in an easy-using manner;

- To provide an efficient way to recognize human's gestures by neural networks. By utilizing the convolutional neural network, the accuracy of the gesture recognition is improved and the operator is able to send a proper command to the augmented reality application.

The remaining parts of the paper are organized as follows: Section 2 is about related work. In Section 3, the details of our approach are presented. In Section 4, the experiments and analysis of our design are presented. The conclusions of the paper go in Section 5.

2. Related work

The origin of human-computer interaction (HCI) to other areas of study such as computer interface design, human factors, usability and specifically to educational environments are examined [6] and now its a time for progress this way to make it as convenient as possible. For the recent years, many approaches have been proposed for the immersive human-machine interface and augmented reality. We briefly review some studies related to mesh reconstruction, gestures reconstruction, and neural networks.

Mesh Reconstruction. There are lots of methods to reconstruct a mesh model by range sensors or scanners. RealSense camera is one of them that can do the scanning with its SDK and tools. However, the result of the scanners is not acceptable in some fields of usage; also it cannot build object just by one try. Using poison to reconstruct a surface from oriented point samples acquired with 3D range scanners is one of the famous methods in 3D reconstruction, but it runs a risk that the data will be over smooth of Kazhdan *et al.* [7]. Calakli and Taubin [8] made efforts in incorporating positional constraints by using poisson reconstruction algorithm. Furthermore, Kazhdan and Hoppe *et al.* [9] proposed screened poison surface reconstruction. It is one of the best surface reconstruction and already implemented in some tools and library such as Meshlab and PCL. Another method to combine all mesh parts together is about using iterative closest point (ICP) algorithm by Holz and Behnke [10]. They had proposed registration of non-uniform density 3D point clouds using approximate surface reconstruction that it can be used to merge all parts from different angles and extract the full object mesh with reasonable quality. The paper that is the main method of proposed approach is keypoints-based surface representation by Shah *et al.* [11] that it got comparable results with all 3D key points detectors.

Gestures Reconstruction. Gesture recognition has aroused considerable interest in research with the increasing demand for human-machine interaction. Many different models based on the spatiotemporal scheme have been used to solve the problem of gesture detection and tracking. Nowozin and Shotton [12] proposed hidden Markov model, which is used to track the movement. Wang *et al.* [13] realized a more efficient way to recognize gestures in hidden conditional random fields for gesture recognition. However, their model does not achieve the goal of extracting higher-level features of hands. Also [14] have focused on using Hierarchical Bayesian Neural Networks and active learning to personalize the human gestures.

Neural Networks. Since the introduction of RGBD depth cameras like Kinect, people not only try to use RGB data but also to use depth data for gesture recognition. Wang *et al.* [13] attempt to use RGB-D data and one-shot learning to train a gesture recognition model. Wan *et al.* [15] similarly, have applied another method to apply deep learning to gesture recognition. In the article of Neverova *et al.* [16], adaptive multi-modal gesture recognition is established by using convolutional neural networks and deep learning. 3D convolutional neural networks for human action recognition are then proposed by Ji *et al.* [17], they applied calibrated and supervised videos to train 3D CNN to automatically identify human actions. This model is able to get many high-level features effectively from the video [18, 19]. Convolutional learning of spatiotemporal features by Taylor *et al.* [20] also used 3D CNN to get Spatio-temporal features. Wu *et al.* [21] then proposed deep dynamic neural networks for multi-modal gesture segmentation and recognition.

In the convolutional network, CNN's proposed by Krizhevsky *et al.* [5] for processing ImageNet promoted the advancement in image recognition. The architecture of [5] consists five convolutional layers, some layers followed with max-pooling layers, and fully-connected layers. They also refer to neurons with nonlinearity as rectified linear units (ReLUs). In 2014, A CNN structure to further improve the original architecture for processing ImageNet is proposed by Simonyan and Zisserman [22] to make use of the deep representation of a network with small 3x3 filters for all convolutional layers. GoogLeNet by Szegedy *et al.* [23], a 22 layers deep network, won ILSVRC14, this architecture uses 12x fewer parameters than the architecture of Krizhevsky *et al.* [5], while being more accurate (6.67 vs 16.4 top-5 error rate). Also, it removes FC layers completely. ResNet by He *et al.* [24] is ILSVRC 2015 winner, 8x more layers than VGG nets [22] but faster at runtime. In this architecture, they introduce a deep residual learning framework with batch normalization after every Conv layer.

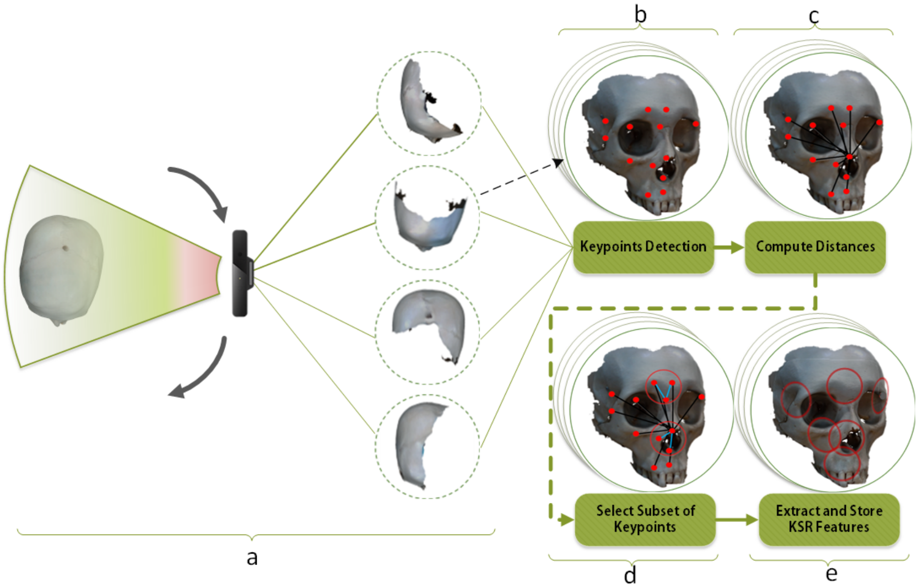


Figure 2: The overview of extracting the KSR features from the scanned results. (a) Scanning the object from different view angle. (b) Detecting the keypoints. (c) Calculate the distances from one feature to another (d) Selecting the KSR features. (e) The results of KSR.

3. Approach

3.1. Mesh reconstruction

3.1.1. Scan by range finder

We have used RealSense camera to scan our objects. The overview of extracting the KSR features from the scanned results is shown if Fig. 2. Regular photo and video will be recorded by a standard 2D camera but RealSense camera is used an infrared camera and an infrared laser projector to detect its distance to every point of faced objects, and it can separate its target from the background. This device comes in three different types: front-facing, rear-facing, and snapshot. These days, the front-facing cameras are most common type because it supports more operating system and allowing all kinds of games and application to use it. However, in this experiment, we have selected the front-end version to scan the human's skeleton. We had to move the camera around our object to scan one side of it. It is an impossible task to make a complete model by just single time scanning. Thus, in this paper, we propose to use scanner more than one time and merge all parts together. In the first step, the model should have scanned from different view angle, save them and make them ready for the step of mesh registration. A demonstration of scanning an object from a single side is shown in Fig. 3.

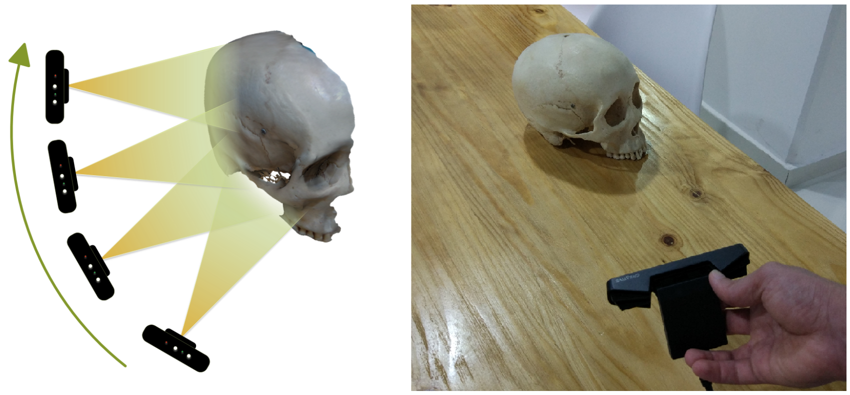


Figure 3: Scan an object from single side. The camera should be moving around our object as much as possible (less than losing the quality).

3.1.2. Merge the divided point clouds

There will be some meshes that have scanned from different view angles. Therefore, they need to be merged and become a complete object mesh. The state-of-the-art KSR method by Shah *et al.* [11] is one of the best methods to merge our meshes into a full object's mesh. Through this algorithm, the geometrical relationship of detected 3D key points for local surface representation will appear. It will output the transformation between point clouds and reference surfaces. KSR is computationally efficient and fast. It has done its task by detecting the key points and then making subsets from them. These subsets are used to make KSR vector that is the main part of merging task. Meanwhile, the scanned parts are important to be clear and have some common feature with other parts. After achieving the KSR features and attaching the local reference frame (LRF) to the features, DOG keypoints detector by Darom and Keller [25] has been used to get the matched point cloud. Fig. 4 has briefly shown the procedure of merging the separated meshes of an object to full mesh.

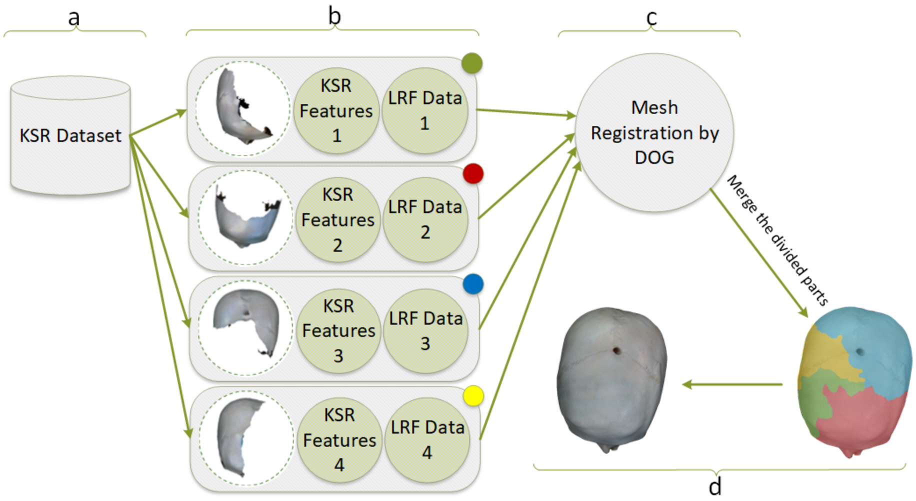


Figure 4: Mesh registration procedure: (a) Dataset from the previous process of extracting KSR features and LRF data (b) The input of the registration process (c) DOG have been used to match the input KSR features. (d) Put the meshes together to get the complete mesh. Each color is connected with one of the mesh parts.

3.2. User interface design

3.2.1. Overview

After research on the current anatomy course in medical school, teachers usually, follow such process: (1) Introducing the general information of a certain part; (2) Decomposing the object into several components and show them with real specimen or picture; (3) Labeling out all the names of each part on the component; (4) Explaining each part and show their position by virtual specimen or pictures. To cover the medical professor needs, the labeling should be precise and the labels have to be in 2 languages: English and Mandarin languages. In order to scan the body organs and to pointing the labels on the right part, a group of medical students has been helping us. At the end of making a complete 3D model, we have a painted model by vertex colors that helped us to recognize right parts and pair them with connected labels.

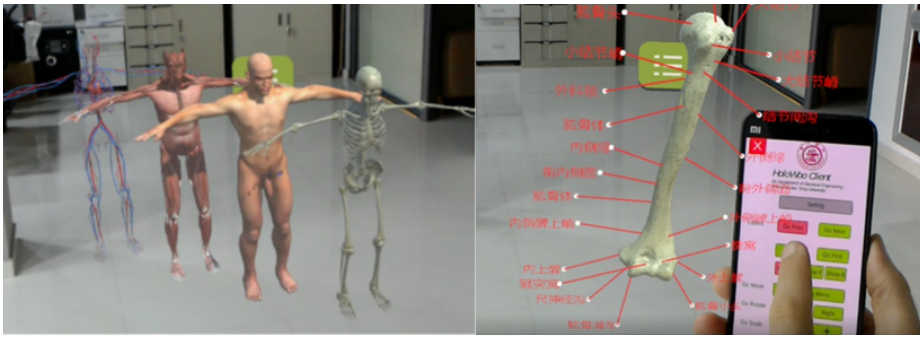


Figure 5: User interface designed to reach our needs in medical school. Left: Whole body menu; Right: The user interface displaying Human Humerus supporting gestures controlling and cellphone APP. The menu will appear in front of the operator based on initial pattern position at setup time.

3.2.2. Existing problems and our design

The real specimens amount is limited and some specimen must be kept in a certain preservation solution. Not all students can have a close look at it and none of them can take one home for review. On the other hand, if the teachers use pictures to conduct the education, although students can observe the picture as much as they want. Therefore, the user interface must meet the following requirements: (1) Provide the general information in text format first; (2) Show the model which is 3D Reconstructed from the real one; (3) Enable people to Zoom, Move, Rotate the model by gestures; (4) Label out all the names of each part of the component; (5) Hide the labels for quiz mode. The user interface is built by Unity3D and realized all the function mentioned above. Meanwhile, the gesture recognition is left for the trained model of neural networks to achieve more user-friendly interaction for aged and experienced teachers.

3.3. Deep convolutional neural networks

3.3.1. Preprocessing

One of the most important parts of learning method is preprocessing part but to use deep learning, in most of the cases the preprocessing is done inside of the network. On the other hand, with the use of convolutional layers, we can recognize the best features to get the acceptable result. In order to decrease the learning process and improve the rate of recognition, we have done some steps of preprocessing to make the gestures ready to proposed deep neural network. Without doing the preprocessing the position of the gesture and also noise had affected to our network and caused to make us use more layer and consume more time to get the favorite results.

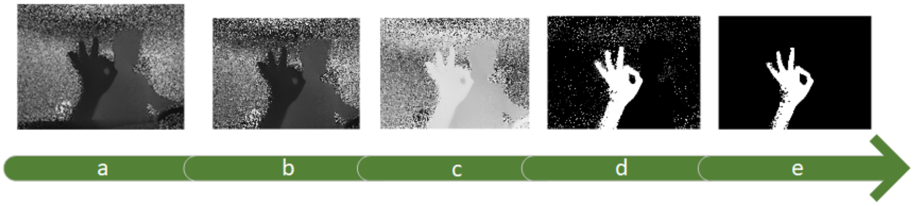


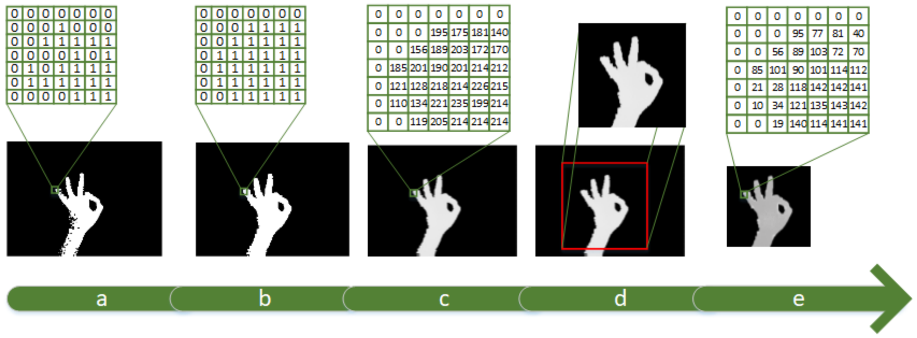
Figure 6: The preprocessing procedure. (a) Original image (b) Cropping image (c) Making negative of gray values (d) Thresholding (e) Eroding.

As it has shown in Fig. 6, we get images directly from the dataset and edit them to achieve the gesture with as less as possible noise. At the first step, our specific dataset has the tiny noises on its borders. Sometimes because of the sitting the operator behind a desk, we have some irrelevant objects around our depth images which affected to our results; such as desk, camera cable, PCs and etc. To remove these things from the images we just have to remove the borders. The size of borders is depending on the dataset. In our experiment, we set it to 50 pixels. The raw result of the real sense camera is different from the others; the nearest point is darker than the far points. To make it such as other depth camera and make it more understandable to human vision, we have made it negative; nearest points are brighter than the far points and then the image is ready for thresholding. As it's clear in Equation 1.

$$
\begin{aligned}
&A = v_{Max} - I \\
&B = \begin{cases} 0 & , A_x < t \\ A_x & , A_x \geq t \end{cases}
\end{aligned}
\tag{1}
$$

where, v_{Max} is the maximum gray level of our images, I is the original image,

A is the negative image, A_x is pointed to each pixel of the negative image, t is the threshold and B is the result of thresholding. On thresholding step, the minimum and maximum of the threshold level must be defined. But these parameters are directly related to the distance of the camera from the user hand gestures [26]. Therefore, the parameters must be set based on our target dataset. You can follow the sample in the fourth step of Fig. 6. It has shown the thresholding results. In this result, there are so many tiny white pixels that appeared because of the noise in the depth camera. The main emphasis is to use morphological methods to remove this kind of noise. Thus erosion has been used to get the fifth step of Fig. 6.



0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	1	1	1	1
0	0	0	0	1	0	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
0	0	0	0	1	1	1

0	0	0	0	0	0	0
0	0	0	1	1	1	1
0	0	1	1	1	1	1
0	1	1	1	1	1	1
0	1	1	1	1	1	1
0	1	1	1	1	1	1
0	0	1	1	1	1	1

0	0	0	0	0	0	0
0	0	0	195	175	181	140
0	0	156	189	203	172	170
0	185	201	190	201	214	212
0	121	128	218	214	226	215
0	110	134	221	235	199	214
0	0	119	205	214	214	214

0	0	0	0	0	0	0
0	0	0	95	77	81	40
0	0	56	89	103	72	70
0	85	101	90	101	114	112
0	21	28	118	142	142	141
0	10	34	121	135	143	142
0	0	19	140	114	141	141

Figure 7: Step 2 of the preprocessing procedure. (a) Eroded image (b) Dilating image (c) Multiply the result to original image (d) Cropping the max area (e) Result of subtracting min value from the gesture pixels.

The Step 2 of the preprocessing procedure is illustrated in Fig. 7. At this step, the white pixels are disappeared but still, some of the important parts of user's hand are noisy. In this part, we have to fill the empty part and pixels of the gestures. Hence by applying the dilation method with disk size 1, the clearer gesture has appeared, also you can follow this procedure in the Equation 2.

$$C = (B \ominus H_E) \oplus H_D \tag{2}$$

where H_E is the structuring element of erosion, H_D is structuring element of dilation and C is the result of morphological noise removing's part. On this step, we have a clear black and white gesture image but the proposed network needs to have the depth data of our gestures. Also cropping the gesture area was one of our goals to have a better input of proposed CNN. Accordingly, in this step, we multiply the back and white image to the negative of the original image. In the result, the black side is disappeared and white side is replaced with original negative pixels (Fig. 8 step 3).

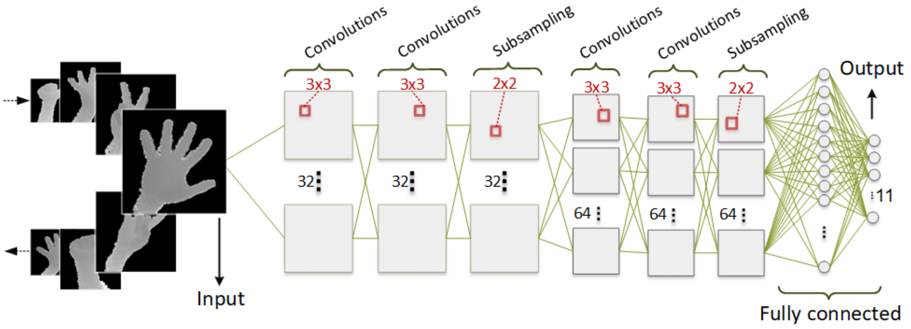


Figure 8: Convolutional network architecture. It should train for every gesture separately. It starts from 32x32 image and by decreasing the size through the network, the number of the Conv layer is increased.

In order to get the gesture area, the maximum value of the gesture has been found and then window size 112x112 is cropped from this area that the Equation 3 is shown this method.

$$D = Max(C \times I)_{area} \tag{3}$$

where D is the result of preprocessing image. It has been shown in step 4 of Fig. 8. Step 5 just makes it more clear to human vision and is a step of the normalizing value of gesture's pixel. To do this, the minimum value was subtracted from the whole gesture points. After doing these steps, the preprocessed dataset is ready to import as an input to our proposed neural network.

3.3.2. Network architecture

The input of our network is 32x32 depth images and We have used 3x3 convolutional layers. The stride for all convolutional layers is 1 pixel. The first layer in our architecture has 32 filters with size 3x3 and padding 1. The next convolutional layer has channel size 64 and the filter size is 3x3 and no padding. Filters of the same amount are used to maintain the dimension. In this architecture, the stride of max-pooling layers is 2 and after max-pooling layer, the feature map size is halved. We have doubled the filter number to preserve the whole size per layer. After convolutional layers, a fully-connected layer with 11 channels is followed. Finally, a SoftMax layer has used in calculating the loss. Except for those linear transformation, we add rectification (ReLU) non-linearity as the activation function [5]. ReLU is very computationally efficient and converges much faster than sigmoid or tanh.

One of the challenges of the learning method and especially deep learning is being time-consuming. Sometimes training for a small dataset takes more than a day. In order to improve the speed of our training, the cloud server is used. Fig. 9 is shown the proposed procedure of using cloud computing. In fact, the input and model of each gesture should send to one specific device to train. In our cloud part, not only the PCs but also Android devices have been used. This task was down by installing Tensorflow library on windows and android devices such as mobile phones and tablets and etc. Therefore, at the first stage, the cloud server is categorizing the train procedures and then related Tensorflow commands and related inputs will be sent to connected PCs or mobile devices. Thus the proposed cloud has divided into 2 main parts: console cloud and mobile cloud.

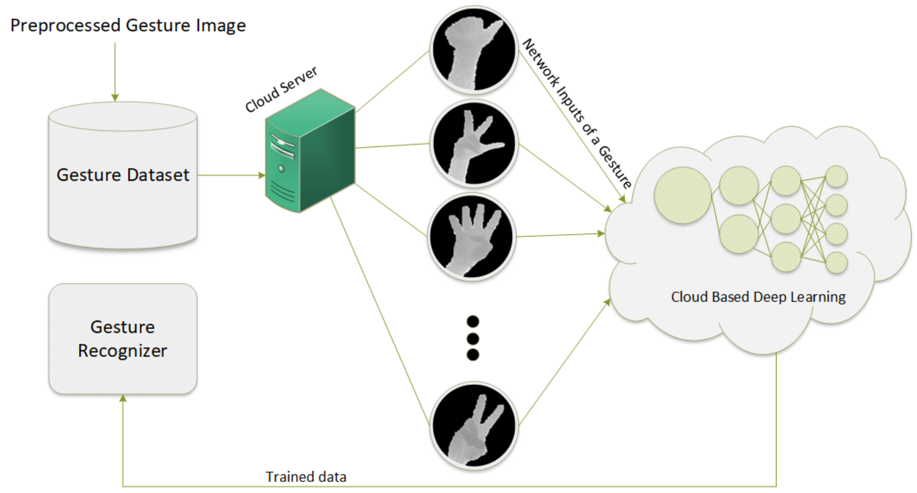

Preprocessed Gesture Image
Gesture Dataset
Cloud Server
Network Inputs of a Gesture
Cloud Based Deep Learning
Gesture Recognizer
Trained data

Figure 9: Procedure of using cloud in order to increasing the speed of training level.

3.3.3. Difficulties and challenges

However, in the implementation, we have found that the proposed network has good performance in a wide range gestures. In the case of several handshaking or rapid movement, it lost the hands' position and also cannot recognize the gestures well. This may be due to very little training data about the blurred situation that it appeared because of the motion. Therefore, if hands move rapidly, the neural network cannot correctly identify the location of the hand. However, in the interaction of augmented reality programs, we often need a wide range of hands movement. Also losing the hand's position or any failure is not acceptable. That is the reason that makes us improve the existing model. On the other hand, to load many meshes just in one application of AR glasses is another challenge. To overcome this step, the application should be a low process and sometimes use some other applications to optimize our reconstructed mesh is necessary. Finding a more optimized way to render or making the low-poly meshes without losing quality is the other challenges in this research.

4. Experiments and analysis

4.1. Gesture recognition: Deep convolutional neural network

The 11 gestures such as Pan, Pinch, Fist and etc. have been trained in our network which has been made in the dataset of Memo *et al.* [27]. In this part of the project to recognize the hand gestures, the RealSense camera has been used. We should mention that Hololens has a range camera but its quality was not enough to the proposed method. The preprocessing was done and then the gestures were set as standard inputs and tested in different layers and trains of the model respectively. Also, we record the recognition statistics in a real scene. Results of two kinds of testing are listed below. Table 1 is a result of cross-subject method, and Table 2 is result of cross-validation method. To cross subject testing every group of the samples should separate into 2 parts: Half for training and the half for testing. But in cross-validation method, we have to divide the samples into 4 parts, 3 for training and 1 for testing. Also in cross-validation after first training, the results are stored and train should be started with 3 other groups. Thus the train must repeat 4 times and at the end, the mean of these 4 trains is our final result. As in Table 1 and Table 2, each row shows the results of testing the network on a specific gesture. For instance, Table 1, row 1 is shown the average rate of estimation with 97 percent accuracy.

In order to get the superiority of proposed method, the results are compared with the results of Memo *et al.* [27]. Proposed method accuracy was interesting but the advantages of the proposed method are more than a just recognition rate. By performing the preprocessing, our proposed approach is trustable and more extendable in comparison with [27]. The output of preprocessing layer is small and cropped image that is affected by the speed of the network in both training and testing process. Also, this method is removed the background pixels hence the network doesn't need to process the ineffective data. In the CNN part, the result of [27] was fine but we have got extremely better results which are shown in Table 3 and Fig. 10.

Table 1: Cross subject testing results. The gestures should be divided into two parts, one for training and one for testing. This table shows the results of the networks for every gesture.

	G1	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**	**G8**	**G9**	**G10**	**G11**
G1	0.97	0	0	0	0.02	0	0	0	0	0.02	0
G2	0	0.98	0	0	0	0	0	0.02	0	0	0
G3	0	0	1	0	0	0	0	0	0	0	0
G4	0	0	0	0.98	0.02	0	0	0	0	0	0
G5	0.02	0	0	0.05	0.92	0	0	0	0.02	0	0
G6	0	0	0	0	0	0.97	0	0	0.02	0	0.02
G7	0	0	0	0	0	0.02	0.98	0	0	0	0
G8	0	0	0.1	0	0	0	0	0.87	0.03	0	0
G9	0	0	0	0.02	0	0	0	0.08	0.9	0	0
G10	0.13	0	0	0	0	0	0	0	0	0.87	0
G11	0	0	0.05	0	0	0	0	0.07	0	0.12	0.77

Table 2: Cross validation testing results. The gestures should be divided into four parts, three for training and one for testing. This procedure has been performed for four times, and the results is the mean of the four testing outputs.

	G1	**G2**	**G3**	**G4**	**G5**	**G6**	**G7**	**G8**	**G9**	**G10**	**G11**
G1	1	0	0	0	0	0	0	0	0	0	0
G2	0	0.93	0.01	0	0	0.01	0.01	0.02	0	0.03	0
G3	0	0	0.99	0.01	0	0	0.01	0	0	0	0
G4	0	0	0	0.99	0	0	0	0	0.01	0	0
G5	0	0	0	0	1	0	0	0	0	0	0
G6	0	0	0.02	0	0	0.98	0	0	0	0	0
G7	0	0	0	0.03	0	0	0.94	0	0	0.03	0
G8	0	0	0	0.02	0	0.01	0	0.93	0.02	0	0.03
G9	0	0	0.01	0	0.01	0	0	0	0.99	0	0
G10	0.01	0	0	0	0	0	0	0	0.02	0.96	0.02
G11	0	0	0	0	0.01	0.01	0	0	0.01	0.06	0.92

Table 3: Comparison between 3D array from Memo *et al.* [27] and our proposed CNN approach with the cross-validation and the cross-subject testing methods.

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	**mean**
3D array	0.94	0.84	1	0.88	0.88	0.88	0.8	0.93	0.82	0.92	1	**0.9**
Ours (Cross Subject)	0.97	0.98	1	0.98	0.92	0.97	0.98	0.87	0.9	0.87	0.77	**0.93**
Ours (Cross Validation)	1	0.93	0.99	0.99	1	0.98	0.94	0.93	0.99	0.96	0.92	**0.97**

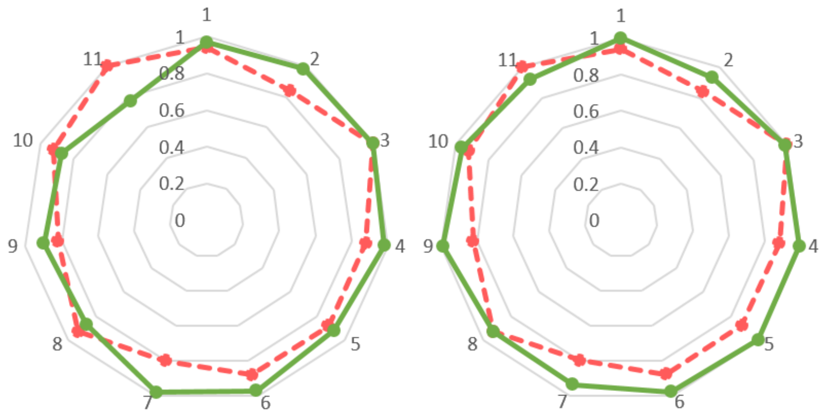


Figure 10: Correctness rate with different model, the solid line is our proposed method and the dotted line is from 3D array method. The left one is the proposed method's result by cross subject testing method and the right one is come from cross validation testing method.

4.2. Interactive learning parts

To implement a real application in AR glasses, we need to have a camera to get gesture directly from the operator and recognize the meaning of that gesture by passing it through CNN. We have implemented 4 applications to achieving the goal of interactive learning in medical schools: 1- first of all, our CNN network that has implemented in Tensorflow library by using Python language that is already explained in section 3. 2- An application in HoloLens augmented reality glasses. This application has divided important parts to doing the interactive learning: First of all, we can mention our novel user interface design. Fig. 5 is shown the parts of this design. The second feature shows the 3D body organ mesh with all labels. The third is our gesture recognition method that helps the operator to interact with the application by their gestures. 3- Interaction with the application is one of the important parts, thus to make it easier, the cellphone APP has been made that it can control every parameter and action, by sending the commands through the UDP network protocol. 4- Scanning the body organs are done by real sense camera, but to merging every part of the scanned mesh, the mesh reconstruction was made that it already explained in section 2.

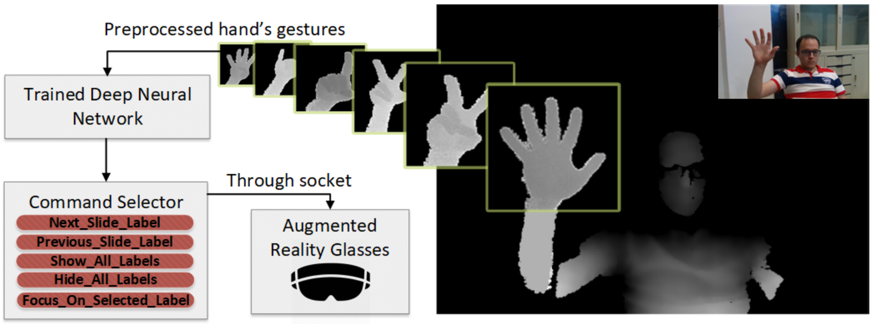


Figure 11: The hand area should be found and crop from the main depth image. By sending the cropped image to trained CNN, and recognizing the gesture, target command is selected. The commands are sent through the socket programming on the active wireless network to the target AR Glasses.

The learning method includes some parts which are easy to use and clear to understand to both professors and students. As an instance, we can mention displaying a human body, muscles, vessels, skeleton as user interface menu. Professors of medical school can wear the HoloLens glasses and see the starting menu. To see the details of one of the parts of human body, they have to select that part with their gestures. If they want to select a part they just have to do the Tap gesture that is already defined in HoloLens glasses. But if the body organ was selected and the labeling is shown, they are able to change the labeling to the next or previous one by using our predefined gestures. An illustration of gesture processing is shown in Fig. 11. At the end of implementing the proposed method, we have honored to hold the first interactive learning course at Shanghai Jiao Tong University.

5. Conclusions

In this paper, we discuss the future trends that combining neural networks and augmented reality to achieve immersive experience and user-friendly operation. It is obvious that augmented reality has much better performance in displaying three-dimensional view than any traditional methods. Considering the demands of realistic experience from medical school when teaching anatomy and the specimen shortage, we try to apply augmented reality in medical anatomy learning. In the process of practice, we try to solve the mesh reconstruction and UI interaction design. At the same time, we no longer want to use the traditional devices as an input source, and hope to use gesture in 3D space to interact. For this purpose, we use the trained neural networks with an RGB-D camera to recognize hand's position and tracks three-dimensional path. Finally, we achieve a good result. In the future of augmented reality development, the reality and virtual interaction will become increasingly important, and this must involve a large variety of people's behavior patterns to detect and track, and this is where we need to combine the augmented reality and neural networks. Up to this point of our research, for thresholding step of the operator have to set the threshold parameter manually, based on the distance of camera and operator. As a future plan, we need to improve this method and remove any interfere in preprocessing methods. Also shaking the hand and rapid movement was another weak point of this method that pushing us to continue this research until achieving the best performance.

References

[1] S. Mitra, T. Acharya, Gesture recognition: A survey, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 37 (3) (2007) 311–324.

[2] L. Pigou, S. Dieleman, P.-J. Kindermans, B. Schrauwen, Sign language recognition using convolutional neural networks, in: European Conference on Computer Vision Workshop, 2015, pp. 572–578.

[3] S. S. Rautaray, A. Agrawal, Vision based hand gesture recognition for human computer interaction: A survey, Artificial Intelligence Review 43 (1) (2015) 1–54.

[4] K. Qian, J. Niu, H. Yang, Developing a gesture based remote humanrobot interaction system using kinect, International Journal of Smart Home 7 (4) (2013) 203–208.

[5] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, in: International Conference on Neural Information Processing Systems, Vol. 1, 2012, pp. 1097–1105.

[6] G. A. Berg, Human-computer interaction (hci) in educational environments: Implications of understanding computers as media, J. Educ. Multimedia Hypermedia 9 (4) (2000) 349–370.
URL `http://dl.acm.org/citation.cfm?id=374674.374847`

[7] M. Kazhdan, M. Bolitho, H. Hoppe, Poisson surface reconstruction, in: Eurographics Symposium on Geometry Processing, 2006, pp. 61–70.

[8] F. Calakli, G. Taubin, SSD: Smooth signed distance surface reconstruction, Computer Graphics Forum 30 (7) (2011) 1993–2002.

[9] M. Kazhdan, H. Hoppe, Screened Poisson surface reconstruction, ACM Transactions on Graphics 32 (3) (2013) 29:1–29:13.

[10] D. Holz, S. Behnke, Registration of non-uniform density 3D laser scans for mapping with micro aerial vehicles, Robotics and Autonomous Systems 74 (2015) 318–330.

[11] S. A. A. Shah, M. Bennamoun, F. Boussaid, Keypoints-based surface representation for 3D modeling and 3D object recognition, Pattern Recognition 64 (2017) 29 – 38.

[12] S. Nowozin, J. Shotton, Action points: A representation for low-latency online human action recognition, Tech. Rep. MSR-TR-2012-68, Microsoft Research Cambridge, Cambridge, U.K. (July 2012).

[13] S. B. Wang, A. Quattoni, L. P. Morency, D. Demirdjian, T. Darrell, Hidden conditional random fields for gesture recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 1521–1527.

[14] A. Joshi, S. Ghosh, M. Betke, S. Sclaroff, H. Pfister, Personalizing gesture recognition using hierarchical bayesian neural networks, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2017.

[15] J. Wan, G. Guo, S. Z. Li, Explore efficient local features from RGB-D data for one-shot learning gesture recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (8) (2016) 1626–1639.

[16] N. Neverova, C. Wolf, G. Taylor, F. Nebout, Moddrop: Adaptive multimodal gesture recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (8) (2016) 1692–1706.

[17] S. Ji, W. Xu, M. Yang, K. Yu, 3D convolutional neural networks for human action recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (1) (2013) 221–231.

[18] F. Porikli, F. Bremond, S. L. Dockstader, J. Ferryman, A. Hoogs, B. C. Lovell, S. Pankanti, B. Rinner, P. Tu, P. L. Venetianer, Video surveillance: past, present, and now the future, IEEE Signal Processing Magazine 30 (3) (2013) 190–198.

[19] M.-C. Roh, B. Christmas, J. Kittler, S.-W. Lee, Gesture spotting for low-resolution sports video annotation, Pattern Recognition 41 (3) (2008) 1124–1137.

[20] G. W. Taylor, R. Fergus, Y. LeCun, C. Bregler, Convolutional learning of spatio-temporal features, in: European Conference on Computer Vision: Part VI, 2010, pp. 140–153.

[21] D. Wu, L. Pigou, P. J. Kindermans, N. D. H. Le, L. Shao, J. Dambre, J. M. Odobez, Deep dynamic neural networks for multimodal gesture segmentation and recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (8) (2016) 1583–1597.

[22] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, CoRR abs/1409.1556 (2014) 1–14.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[24] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[25] T. Darom, Y. Keller, Scale-invariant features for 3-d mesh models, IEEE Transactions on Image Processing 21 (5) (2012) 2758–2769.

[26] S. Reifinger, F. Wallhoff, M. Ablassmeier, T. Poitschke, G. Rigoll, Static and dynamic hand-gesture recognition for augmented reality applications, in: International Conference on Human-computer Interaction: Intelligent Multimodal Interaction Environments, 2007, pp. 728–737.

[27] A. Memo, L. Minto, P. Zanuttigh, Exploiting silhouette descriptors and synthetic data for hand gesture recognition, in: Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference, 2015, pp. 1–9.