Research on Technology of Human Pose Estimation and Robotic Arm Linkage Based on Computer Vision

Xue Yang[1,a] , Hongyang Yu[1,b], Wanjun Huang[1,c]

[1]*Research Institute Electronic Science and Technology,*
University of Electronic Science and Technology of China n, Chengdu, China
a. zephyryx77@163.com, b.hyyu@uestc.edu.cn, c.wanjun_huang2017@163.com

Keywords: MobileNetV2, Person pose estimation, 3D human arm information recovery, Human-computer interaction, style, styling, insert.

Abstract: At present, the restoration of 3D human information based on a single picture or video mainly uses 3D human body reconstruction in view of deep learning, which has the problems of long calculation time and high hardware requirements. For ordinary monocular cameras, 3D human arm information recovery method based on image ranging and spatial geometry is proposed. First of all, MobileNetV2 neural network has the characteristics of lightweight and low latency. The MobileNetV2 is used for 2D human pose estimation, and the original activation function is modified to maintain a low amount of calculations and parameters compared to traditional networks. It also improves the recognition rate. Then, based on the principle of camera imaging and spatial geometry, the method of estimating the three-dimensional information from the two-dimensional information of the human body is studied to obtain the three-dimensional information of the human right arm. Next, the joint angle is calculated by the space vector method to realize the consistency mapping from arm to manipulator and determine the motion of the wrist joint. Finally, relevant experimental research based on the UR manipulator was carried out to complete the human-machine natural interaction experiment, which verified the feasibility and effectiveness of this scheme.

1. Introduction

The vision-based human pose estimation mainly detects the position, direction and scale information of various parts or joints of the human body from the image. It brings a completely new interactive way to automatically recognize human action gestures. Gestures and actions are used to convey the user's meaning. In noisy environments, such as airports and factories, human-computer interaction technologies such as human gesture recognition can provide more accurate information than voice recognition. It has extensive application prospects in the fields of human activity analysis and video surveillance [1].

Human-Machine Inclusion Robot is an emerging research object with the core characteristics of the same natural space of human-machine, natural interaction of human-machine, learning of human skills, and coordination and complementarity with human. Direct teaching of human-machine natural interaction is a non-programmed teaching technology that uses human motion as teaching content, which is relatively difficult, the robot can reproduce the movements of the human body in a manner consistent with the habits and cognition of the teacher. The method can not only improve human-machine collaboration efficiency, but also provide data support for human advanced action semantic division [2]. In addition, the technology that can determine and control the motion trajectory of a wrist joint is also widely used in industrial production environments. Therefore, it is of practical significance and application prospect to study human-computer interaction technology that can reproduce human arm movement.

Firstly, this paper introduces the official MobileNetV2 network structure. And the network has been changed to ensure that the recognition rate increased with a low amount of calculations. Secondly, the two-dimensional coordinates of human right arm is obtained through the human key-point information, which estimated by the neural network. Finally, the method of image ranging and spatial three-dimensional geometry is used to restore the three-dimensional information of the human right arm, and the three-dimensional information is mapped to the robotic arm, so that the robotic arm moves with the human arm, and the quasi-real-time reproduction of the human arm motion trajectory and the control of the robot arm wrist joint motion trajectory are completed.

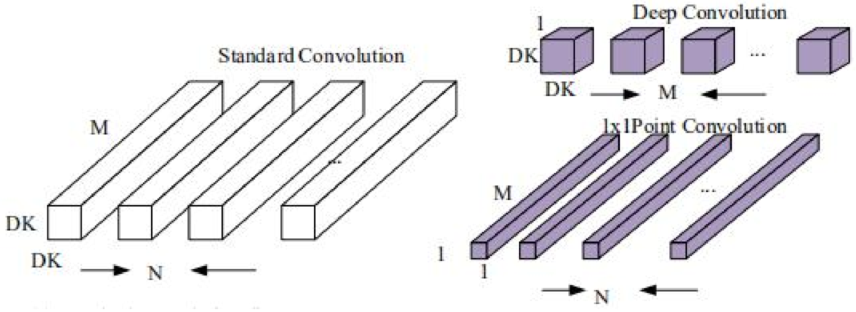
2. Mobilenetv2 Network

MobileNetV2 is a lightweight convolutional neural network [3], which makes two improvements based on the MobileNet network [4]. The first point is to learn from the structure of ResNet [5], including the process of increasing the dimension of the input image, deep decomposition convolution, and then reducing the dimension. The second point is to improve the depth separable convolution, a 1x1 convolution before the deep convolution is added to ensure that the convolution process is performed at high latitudes. In order to reduce the damage of the activation function to the feature at low latitudes, the second activation function after 1x1 convolution is removed.

2.1. Deep Decomposition Convolution

Deep separable convolution [6] as the main structure of lightweight networks, and its key role reduce network parameters and speed up network operation. It is assumed that the number of channels of the input feature map and the output feature map are M and N, respectively. As shown in Figure 1.

Standard convolution uses multiple convolution kernels with the same number of input data channels as the summation after channel-by-channel convolution. The core of the deep separable convolution factorize the standard convolution operation into a deep convolution operation and a point convolution operation. Deep convolution performs convolution on each channel of the input feature map separately, and including 1 convolution kernel channel. The point convolution is responsible for performing 1x1 convolution on the M feature maps obtained from the deep convolution, and including N convolutions.



(a)Standard convolution diagram (b)Deep separable convolution diagram

Figure 1: Comparison of standard convolution and depth separable convolution structures.

The traditional convolution layer takes a set of feature maps of size $D_F\times D_F\times M$ as input, and generates a set of feature maps of size $D_F\times D_F\times N$ as output. D_F means width and height of the input feature map, M means the number of input channels, and N means the number of output feature channels. The convolution kernel K in the convolution layer contains $D_K\times D_K\times M\times N$ parameters, D_K means the side length of the convolution kernel. The calculation of a convolution kernel when processing input data is $D_K\times D_K\times M\times D_F\times D_F$. While convolution kernels increasing N, the calculation of the convolution layer is $D_K\times D_K\times M\times D_F\times D_F\times N$. The depth separable convolution is depth convolution and point convolution. The cost of depth convolution is $D_K\times D_K\times M\times D_F\times D_F$ and the calculation of point convolution is $M\times D_F\times D_F\times N$. So, the total calculation of this combination is $D_K\times D_K\times M\times D_F\times D_F$ $+$ $M\times D_F\times D_F\times N$. The ratio of the calculation of the depth separable convolution compared to the standard convolution shows in Equation 1.

$$\frac{D_K\times D_K\times M\times D_F\times D_F+M\times D_F\times D_F\times N}{D_K\times D_K\times M\times D_F\times D_F\times N}=\frac{1}{N}+\frac{1}{D_K^2} \qquad (1)$$

2.2. MobileNetV2 Network Structure

Based on streamlined architecture, MobileNetV2 uses deep separable convolutions to build lightweight deep neural networks. Its main work is using the deep separable convolution to replace the standard convolution to solve the computational efficiency and parameter of the convolutional network. It solves the standard convolution integration into a deep convolution and a point convolution. Deep convolution applies each convolution kernel to each channel, while point convolution used to combine the output of the channel convolution. The network structure is shown in Figure 2.

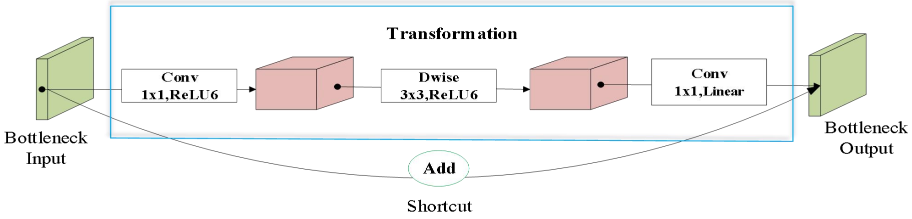


Figure 2: MobileNetV2 network structure.

Linear Bottlenecks, removing the non-linear activation layer behind the second point convolution output layer, in order to ensure the expressiveness of the model. According to the activation function[7] $\mathrm{Re}lu(x)=\max(0,x)$, for the non-zero output of the ReLU layer, it plays a linear transformation role. According to experiments [3], the original input dimension increased to 15 or 30, which used as the input of ReLU, the output loss a little after the output restored to the original dimension. If the original input dimension increased to 2 or 3 only, for the input of ReLU, the information is lost a lot after the output restored to the original dimension. Therefore, in MobileNetV2, the non-linear activation layer like ReLU is removed after the convolution layer performed dimension reduction.

According to the network structure, the calculation distribution[3] of MobileNetV2 distributed in Table 1. And t means the multiplication factor of the input channel, n means the number of repetitions of the module, and Cout means the number of output channels.

Table 1: mobilenetV2 calculation distribution.

Input	Operator	t	Cout	n	Stride
224^2x3	Conv2d	-	32	1	2
112^2x32	bottleneck	1	16	1	1
112^2x16	bottleneck	6	24	2	2
56^2x24	bottleneck	6	32	3	2
28^2x32	bottleneck	6	64	4	1
14^2x64	bottleneck	6	96	3	2
14^2x96	bottleneck	6	160	3	2
7^2x160	bottleneck	6	320	1	1
7^2x320	Conv2d 1x1	-	1280	1	1
7^2x1280	avgpool 7x7	-	-	1	-
1x1x1280	Conv2d 1x1	-	K	-	

According to MobileNetV2， the network ascended dimension firstly and then reduced, which can enhance the propagation of gradients and significantly reduce the memory footprint required during inference. Removing the activation function of the second point convolution, using linear superposition, retaining the feature diversity, and enhancing the expression ability of the network [8], so the model can adapt to images of different sizes. Using ReLU6 (the highest output is 6) activation function makes the model more robust under low-precision calculations, and its complexity is calculated by Equation 2.

$$Complexity:\frac{Depth-wise\ Separable\ CONV}{S\tan dard\ CONV}=\frac{1}{K^{2}}+\frac{1}{C_{out}}\quad\frac{1}{K^{2}} \tag{2}$$

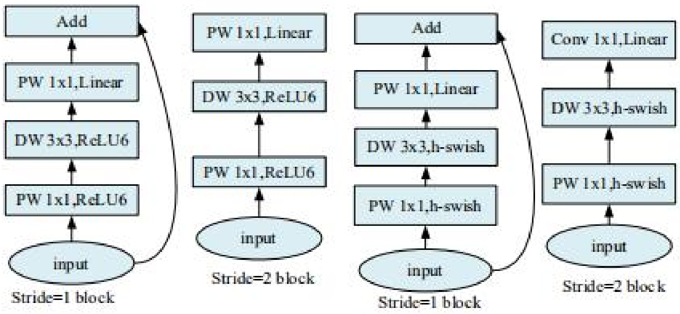
K means the size of the convolution kernel and C_{out} represents the number of output channels.

2.3. MobileNetV2 Network Structure

MobileNetV2, $\mathrm{Re}lu(x)=\max(0,x)$ as the default activation functioncan effectively increase non-linearity in high-dimensional space, and destroy features in low-dimensional space. This paper adjusts the activation function of the network to h-swish function [9,10]. As shown in Equation 3.

$$h-swish[x]=x\frac{\mathrm{Re}\,LU\,6(x+3)}{6} \tag{3}$$

Comparison of network structures is shown in Figure 3;



(a) MobileNetV2 Official (b) MobileNetV2 implemention of this paper

Figure 3: Comparison of network structures.

DW: Deep convolution, applying a single-channel lightweight filter to each input channel. PW: Point convolution, which is responsible for calculating linear combinations of input channels to construct new features. H-swish, this non-linear function brings many advantages [10]. Firstly, ReLU6 can be implemented in many software and hardware frameworks. Secondly, it avoids the loss of numerical accuracy during quantization. And the network has a positive effect on accuracy and delay by this change [11].

2.4. Human arm 3D Information Recovery

At present, the restoration of 3D human information based on a single picture or video mainly uses deep learning to reconstruct 3D human body. The neural network is used to fuse image features of different scales into 3D space to helps restore precise surface geometry through stereo feature transformation [12,13], but this method takes a lot of time to calculate and requires high hardware equipment. So, this paper proposes a method based on image ranging and spatial geometry to recover the three-dimensional information of the human right arm from the two-dimensional coordinate information of the human body in a single picture.

MobileNetV2 is used to estimate the two-dimensional position information of human joints, for solving the joint angle, we need the three-dimensional position information of these joint points on the base of the human arm. Therefore, firstly, the coordinate system of the known two-dimensional information is modeled by the method of spatial solid geometry. At the same time, the coordinate transformation is performed to the human right arm base system, and the three-dimensional coordinate information of the right arm is restored. This paper takes the human right arm as the research object and numbers these joint points of right arm for description. 1 represents the right shoulder, 2 represents the right elbow, 3 represents the right wrist. We set the monocular camera coordinate system fixed named O_{xyz} and human arm base coordinate system named O_{uvw} .The relationship between two coordinate system is shown in Figure 4. In the experiment, please make sure that the human body facing the monocular camera.

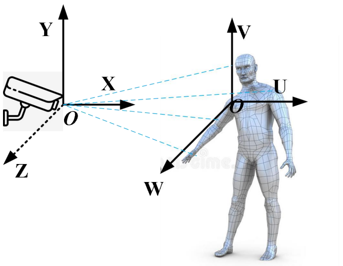


Figure 4: Coordinate system transformation map.

There are steps for coordinate transformation, (1)Initialization, obtaining the length of the arm. We set right arm flat as initial state, assuming the length from the right shoulder to the right elbow is L1 and length from right elbow to right wrist is L2. W * H means image size (W represents the width of the picture, H represents the height of the picture), As shown in Figure 5. Rotating right arm, according to Equations 4 and 5, a series of values can be obtained. The top ten percent value is selected, the average value is used as the actual length of the arm, which can reduce the calculation error of the arm length.

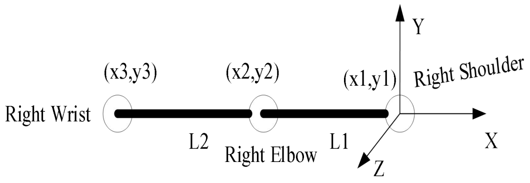


Figure 5: Calculating the actual length of the arm.

$$L1=\sqrt{\left(W*(x_2-x_1)\right)^2+\left(H*(y_2-y_1)\right)^2} \tag{4}$$

$$L2=\sqrt{\left(W*(x_3-x_2)\right)^2+\left(H*(y_3-y_2)\right)^2} \tag{5}$$

(2)Restore the 3D coordinate information of the right arm. According to the imaging principle of the camera, the projection length of the arm, S1 and S2, is displayed on the image screen, and the absolute positions of the joint points in the human arm coordinate system are shown in Figure 6;

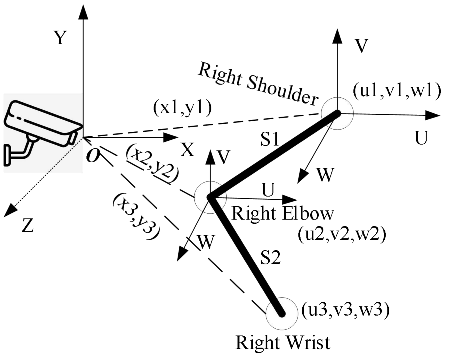
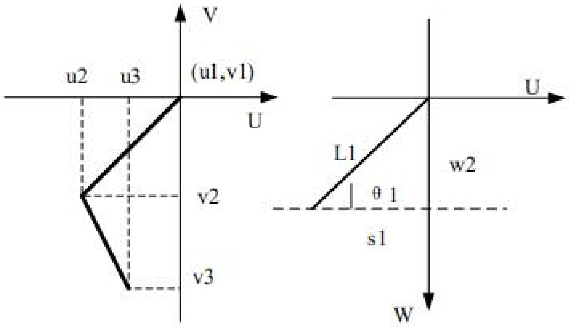


Figure 6: Calculating three-dimensional coordinates.

In picture, the right shoulder acts as the zero point of the base standard system, so, each joint point based on the length of the coordinate system in the U and V axes is calculated by Equation 6 and Equation 7. S1 represents the projection length and is calculated by Equation 8.



(a) UV-axis plan view (b) UW-axis plan view

Figure 7: Calculating the W-axis plane map.

$$u_2=(x_2-x_1)*W \quad v_2=(y_2-y_1)*H \tag{6}$$

$$u_3=(x_3-x_2)*W \quad v_3=(y_3-y_2)*H \tag{7}$$

$$S_1=\sqrt{(W*(u_2-v_1))^2+(H*(u_2-v_1))^2} \tag{8}$$

$x_1, x_2, x_3, y_1 y_2, y_3$ represent the two-dimensional coordinate values of each joint point in the image coordinate system. Therefore, when the arm moves to a certain angle, for the segment from the right shoulder to the right elbow, calculating the W axis value according to the geometric algorithm is shown in Figure 7;

W2 means the length of W axis, L1 means the actual length of arm, S1 represents the projection length, θ_1 means the angle between L1and S1, and it can be calculating by the cosine theorem. As shown in Equation9.

$$L_1\cos\theta_1=s_1 \quad \cos\theta_1=\frac{s_1}{L_1} \quad \text{.........................} \tag{9}$$

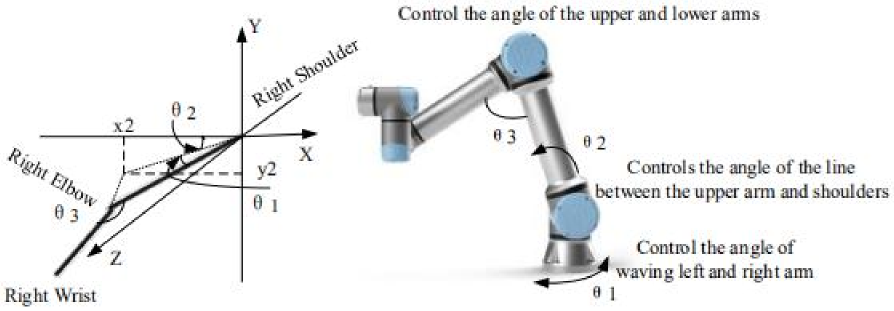
Because our movement is guaranteed to be in front of the camera, all angle values are less than π. After that, the value in the W axis is $w_2=L_1\sin\theta_1$ .So, the coordinate value of the right elbow is(u_2, v_2, w_2), Similarly, translating the base standard system to the right elbow, and we can obtain the coordinate information of the right wrist, (u_3, v_3, w_3) . At this point, three-dimensional information of the three key points of the right arm is obtained.

After the coordinate transformation is completed, the following description of the position of the origin of each joint point will be based on the human arm base system connected to the human body as a fixed coordinate system O_{xyz}, and as the moving coordinate systemO_{uvw} is related to the joint to be solved, the subsequent calculations are independent of the camera coordinate system of the monocular camera.

2.5. Human-Machine Joint Mapping

A human's right arm consists three joints: shoulder, elbow and wrist. The shoulder has 3 degrees of freedom, including abduction-adduction, flexion and extension, and internal and external rotation of the arm. Elbow has 1 degree of freedom, which is the flexion and extension movement between the boom and forearm. The wrist has 3 degrees of freedom, including abduction and adduction, flexion and extension, and forearm rotation. The joint mapping between human arm and the humanoid robotic arm based on the tandem rotation joint can realize one-to-one mapping. However, each joint of the UR robot arm in this experiment has only one degree of freedom, it cannot realize all the movements of the arm. So, under the conditions of UR manipulator motion constraints, the base frame of the right arm of the human body is overlapped with the base frame of the robotic arm, and the joint angle is calculated by the space vector method to drive the robotic arm to reproduce the movement of the human arm. The corresponding relationship: (1) The UR manipulator's 1 and 2 joints simulate the abduction-adduction and flexion and extension movements of the shoulder joint, that is, the angle of the left and right waving corresponds to the base degree of freedom of the robotic arm, and the angle between the connection between the upper arm and the shoulders corresponds to the second degree of freedom of UR. (2) The third joint of the robot arm is used to simulate the elbow joint, the angle between the boom and forearm corresponds to the third degree of freedom of the robot. (3) The fourth joint of the robotic arm simulates the wrist joint, and the spatial position of the wrist joint can be obtained because of experimental data. The other two degrees of freedom can be placed at the zero position. The comparison of the man-machine motion model in this paper is shown in Figure 8;

θ_1 indicates the projection angle (the angle between the upper arm and the XY plane), θ_2 indicates the position of the arm on the XY plane, and θ_3 indicates the angle between the upper arm and the lower arm.


Y
Right Shoulder
θ 2
x2
X
y2
θ 1
Right Elbow
θ 3
Z
Right Wrist
Control the angle of the upper and lower arms
θ 3
θ 2
Controls the angle of the line between the upper arm and shoulders
Control the angle of waving left and right arm
θ 1

(a) Calculate joint angle diagram (b) picture of UR arm

Figure 8: Human-machine motion model comparison.

When UR arm reproduced human motion, the position information of each joint point has been calculated by spatial geometric methods. The angle value of the robotic arm can be obtained from Equation 10, 11, and 12, thereby determining the motion trajectory of the robotic arm.

$$L_1 \cos\theta_1 = S_1 \quad \cos\theta_1 = \frac{S_1}{L_1} \tag{10}$$

$$\cos\theta_2 = \frac{x_2}{\sqrt{\left(x_2^2 + y_2^2\right)}} \tag{11}$$

$$\begin{cases} \cos\theta_3 = \dfrac{\left(l_1^2 + l_2^2 - l^2\right)}{2 l_1 l_2} \quad l_1 = \sqrt{\left((x_2 - x_1)^2 + (y_2 - y_1)^2\right)} \\ l_2 = \sqrt{\left((x_3 - x_2)^2 + (y_3 - y_2)^2\right)} \\ l = \sqrt{\left((x_3 + x_2 - x_1)^2 + (y_3 + y_2 - y_1)^2\right)} \end{cases} \tag{12}$$

After obtaining the angle value of each joint, the trajectory of the UR manipulator is controlled by TCP communication to realize the manipulator to reproduce the human arm movement.

3. Experimental Results

3.1. Data Set Introduction

The COCO 2017 dataset released by Microsoft can be used in image detection, semantic segmentation, instance segmentation, key point detection, and image-speaking scenarios. In this paper, the COCO data set is used for human key-point detection, and the main idea of evaluating key-point detection is to simulate the evaluation indicators used for target detection, named average precision (AP) and average recall (AR) and its variants [14]. The core of these metrics are similarity metrics between actual real objects and predicted objects. In the case of object detection, IoU (intersection-over-union) is used as this similarity measure (for boxes and fragments). IoU implicitly defines the match between the actual real object and the predicted object, and allows the calculation of the precision recall curve. We perform key point detection by defining object key point similarity (OKS). With the open source of the COCO dataset, great progress has been made in the field of human pose estimation in recent years. At the same time, the COCO data set has become a recognized "standard" data set for the performance evaluation of key point detection algorithms. This paper uses MSCOCO 2017 for model training.

3.2. Human-Computer Natural Interaction Results

The CPU of this experimental environment is i7-4790k, and the GPU is GTX 1080 Ti. The official implementation of MobileNetV2 based on Caffe2. This article uses TensorFlow, network parameters and main hyper-parameters are official. In this paper ,the activation function was changed. Compared with the official implementation, the modified MobileNetV2 has 1.1% improvement in recognition rate. In this experimental environment, the running speed of the network's is basically same between activation function before and after modification, which can reach about 300ms to process a picture, and the recognition results are shown in Figures 9 and 10;

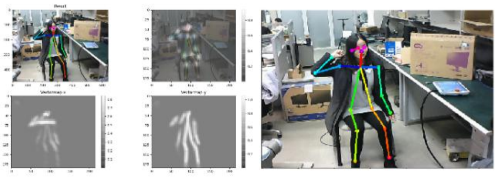


Figure 9: Recognition result and vector map.

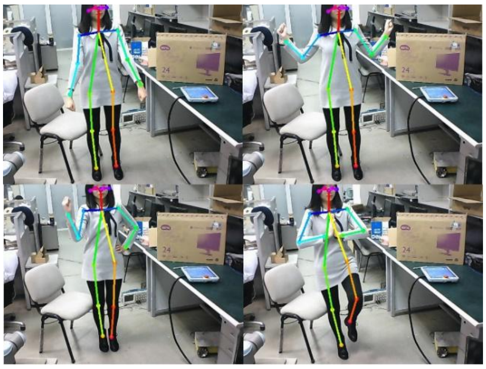


Figure 10: Improved MobileNetV2 recognition effect diagram.

The network performance on the COCO dataset shown in Table 2;

Table 2. MobileNetV2 performance comparison chart (COCO，AP at IoU=0.50:0.05:0.95)

Implemention	AP	AP_{50}	AP_{75}	AP_M	AP_L
MobileNetV2	28.1	55.6	24.7	28.0	28.4
Implementation of this paper	29.2	56.3	25.9	29.1	29.3

From the table, it can be seen that the overall effect of the improved MobileNetV2 is better than traditional MobileNetV2. As Threshold value increased, its accuracy (the ratio of identifying key points of the human body to the total identified objects) decreased.

Using the algorithm proposed in this paper, the 3D coordinate information is calculated based on the 2D coordinates of the joint points, which estimated by the convolution network. And coordinate information are mapped to the UR to complete the quasi real-time interaction between the robotic arm and the human body. The results are shown in Figure 11;

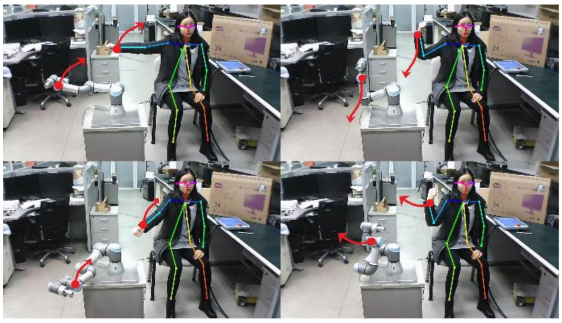


Figure 11: Interaction between the right arm and the robotic arm.

Compared with the standard MobileNetV2, In this paper, its recognition rate increases, less affected by the outside world, and in the quasi-real-time interaction with the robotic arm, the delay (the time difference between the action of the human arm and the corresponding action of the robotic arm) reaches about 500 ms.

3.3. Human-Machine Pose Similarity Evaluation

The robotic arm imitate according to the movement of the human arm, and the similarity of the movement between the robotic arm and the human arm can be evaluated by two ways: the positional relationship or the angular relationship between the two poses. This paper measures the similarity of movement imitation by the angle difference between two poses. At the same time, the angular difference between the robot arm attitude R and the arm attitude G is $\varepsilon(R,G)$, and the similarity is defined as shown in Equation 13;

$$f(R,G)=\frac{1}{1+\varepsilon(R,G)} \tag{13}$$

We assume that λ represents the angle difference of each joint of the right arm in the adjacent posture of the real human arm. γ represents corresponding angle difference of each joint in the adjacent pose of the robotic arm estimated from the posture of the human arm. The angle difference calculation $\varepsilon(\ \lambda,\ \gamma)$ is shown in Equation 14. Obviously, $\varepsilon(\ \lambda,\ \gamma)$ is greater than or equal to 0. The smaller the angle difference, the greater the similarity, so the value of $f(R,G)$ is(0, 1], the maximum similarity is 1.

$$\varepsilon(\Delta\lambda,\Delta\gamma)=\sum_{i=1}^{3}\left|\frac{\Delta\lambda_i-\Delta\gamma_i}{\Delta\lambda_i}\right| \tag{14}$$

These values of $\varepsilon(\ \lambda,\ \gamma)$ are different under different poses. The similarity is converted to the degree of difference according to the angle difference. The similarity curve obtained as shown in Figure 12.

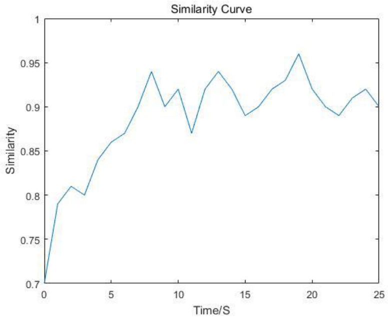


Figure 12: Human-machine attitude similarity curve based on joint angle.

In industrial applications, posture simulation of the wrist joint is more common. However, in this experiment, the position of the wrist joint can be determined by the angles of the three joints from the bottom to the top of the robotic arm. The freedom of the wrist joint is not considered, but the position of the fourth joint of the robotic arm is used to simulate the wrist joint, and the posture of the UR is not changed in all movements due to UR`s structure, so the angle relationship of the wrist joints in this experiment is 1.

The experimental results show that the algorithm proposed in this paper based on the method of image ranging and spatial geometry to recover the three-dimensional information of the human right arm from the two-dimensional coordinate information of the human body in a single picture can make the robotic arm reproduce the motion trajectory of the human arm (Constrained by the structure of the UR manipulator, some actions cannot be reproduced). And the synchronization between right arm and UR is high.

4. Conclusions

This paper proposes a quasi-real-time system that robotic arm mimics human arm movement based on a lightweight network MobileNetV2, and optimizes the MobileNetV2 model. At the same time, the three-dimensional information of the human arm was recovered from a single picture or video and linked with the robotic arm. The interaction time difference was within 500ms. The experimental results show that the method proposed in this paper realizes the complete real-time motion mapping and motion reproduction between human and robotic arms. This paper provides an informative method for the natural human-computer interaction based on the monocular camera.

References

[1] Du Yonghui. Research on two-dimensional whole body pose estimation based on monocular video sequence [D]. Shandong University, 2015.

[2] He Yuqing, Zhao Yiwen, Han Jianda, et al. Harmony with people——the new trend of the development of robotics [J]. Robot Industry, 2015 (5): 74-80.

[3] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen. MobileNetV2: Inverted Residuals and Linear Bottlenecks[J]. The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510-4520

[4] Howard, Andrew G, Zhu, Menglong, Chen, Bo. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J]. Computer Vision and Pattern Recognition (cs.CV)

[5] Targ S , Almeida D , Lyman K . Resnet in Resnet: Generalizing Residual Architectures[J]. 2016.

[6] Francois Chollet. Xception: Deep Learning with Depthwise Separable Convolutions[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017.

[7] C. Zhang, P. C. Woodland. DNN speaker adaptation using parameterised sigmoid and ReLU hidden activation functions[C]// 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016.

[8] Ohn I , Kim Y . Smooth function approximation by deep neural networks with general activation functions[J]. Entropy, 2019.

[9] Eger, Steffen, Youssef, Paul, Gurevych, Iryna. Is it Time to Swish? Comparing Deep Learning Activation Functions Across NLP tasks[J].2019

[10] Howard A , Sandler M , Chu G , et al. Searching for MobileNetV3[J]. 2019.

[11] Chieng H H , Wahid N , Ong P , et al. Flatten-T Swish: a thresholded ReLU-Swish-like activation function for deep learning[J]. 2018.

[12] Kolotouros N , Pavlakos G , Daniilidis K . Convolutional Mesh Regression for Single-Image Human Shape Reconstruction[J]. 2019.

[13] Zheng, Zerong, Yu, Tao, Wei, Yixuan. DeepHuman: 3D Human Reconstruction from a Single Image[J]. 2019

[14] Lin T Y , Maire M , Belongie S , et al. Microsoft COCO: Common Objects in Context[J]. 2014.

[15] Yu Le. Research on human arm movement simulation by six-axis robotic arm [D] .2018.