machines

MDPI

Article

Skill Acquisition and Controller Design of Desktop Robot Manipulator Based on Audio–Visual Information Fusion

Chunxu Li [1], Xiaoyu Chen [2], Xinglu Ma [2,*], Hao Sun [2] and Bin Wang [2]

[1] College of Mechanical and Electrical Engineering, Hohai University, Changzhou 213000, China
[2] Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China
* Correspondence: maxinglu@qust.edu.cn

Abstract: The development of AI and robotics has led to an explosion of research and the number of implementations in automated systems. However, whilst commonplace in manufacturing, these approaches have not impacted chemistry due to difficulty in developing robot systems that are dexterous enough for experimental operation. In this paper, a control system for desktop experimental manipulators based on an audio-visual information fusion algorithm was designed. The robot could replace the operator to complete some tedious and dangerous experimental work by teaching it the arm movement skills. The system is divided into two parts: skill acquisition and movement control. For the former, the visual signal was obtained through two algorithms of motion detection, which were realized by an improved two-stream convolutional network; the audio signal was extracted by Voice AI with regular expressions. Then, we combined the audio and visual information to obtain high coincidence motor skills. The accuracy of skill acquisition can reach more than 81%. The latter employed motor control and grasping pose recognition, which achieved precise controlling and grasping. The system can be used for the teaching and control work of chemical experiments with specific processes. It can replace the operator to complete the chemical experiment work while greatly reducing the programming threshold and improving the efficiency.

Keywords: desktop experimental manipulators; skill acquisition; motion control; motion detection; speech recognition; information fusion; pose recognition

Citation: Li, C.; Chen, X.; Ma, X.; Sun, H.; Wang, B. Skill Acquisition and Controller Design of Desktop Robot Manipulator Based on Audio–Visual Information Fusion. *Machines* **2022**, *10*, 772. https://doi.org/10.3390/machines10090772

Academic Editor: Raffaele Di Gregorio

Received: 14 July 2022
Accepted: 31 August 2022
Published: 6 September 2022

1. Introduction

Modern production is subject to a great deal of uncertainty since new items are being introduced at an increasingly rapid rate, particularly those with multiple varieties and a limited lifespan. Therefore, the ability to create flexible and reconfigurable production systems is highly desired. When it comes to flexible tasks, human abilities such as quick perception and the processing of different types of information or adaptability and improvisation can be crucial success factors [1–3]. The most sensitive jobs, including operating chemical experiments, still rely entirely on physical labor, despite automated robots playing a large role in modern manufacturing lines. These lines require significant labor, and a human worker may have to spend hours drilling in screws or wheels without stopping. Modern business needs collaborative robots that can effectively aid human employees because labor is becoming more expensive due to an aging population. Therefore, giving full play to the advantages of robots and using them to complete some cumbersome and high-risk experiments will become the main direction of the intelligent development of chemical laboratories. In the same process, the use of intelligent devices such as robots and robotic arms requires the mastery of complex programming and control techniques, which is still difficult for chemical experimenters. This raises the question of how to control robots, robotic arms and other intelligent devices to complete chemical experiments in a simpler and faster way.

Chemical experiment is an indispensable link in the process of research, study and production in the chemical industry [4]. At present, the intelligence of chemical engineering procedures is not very high; the experimental process that can be completed by using intelligent mechanical equipment such as manipulators to carry out chemical experiments is relatively simple, and the programming is relatively complex. In addition, various chemical reagents are used in chemical experiments. These reagents interact with each other during the experiment to produce various harmful substances, and unpredictable dangers may occur as a result of experimental errors. Therefore, it is necessary to use more intelligent manipulators and other equipment to replace the operator to complete the relevant complex tasks.

Today's technologies enable us to send a robot to land on Mars, but not to properly control a robot shaking hands with us. The performance of most advanced robot control systems at present still cannot match that of humans' adaptability, flexibility and cooperative ability, which are urgently required by the flexible manufacturing systems used to facilitate mass customization in the context of Industry 4.0 [5,6]. Modelling human skills from a robot control view is still challenging, especially for high level versatile and collaborative skills. Robots have recently been finding their way into human industrial and daily life, an example of which is by learning motor skills from human tutors through demonstration, and then generating these learned skills [7]. Obviously, a skilled robot would be more efficient in interacting with humans and industrial productions. It is increasingly expected that robots should be capable of flexible skills in order to adapt to more complex situations. Teaching by demonstration is seen as one of the most effective ways for a robot to learn motion and manipulation skills from humans [8]. In this paper, inspired by human mechanical intelligence adaptivity to variations of tasks, both position trajectory and oral interaction are achieved for robot motion control to realize a more completed skill transfer process.

With the development of machine learning and machine vision technology, the deductive programming of the manipulator provides a new solution for human-computer interaction, which is an important way to reduce the difficulty of acquiring skills for the manipulator [9–11]. The teaching programming of the manipulator is a process of automatically learning the motion trajectory by watching and learning the teaching actions of people; teaching robots by visual signals such as motion tracking-based teleoperation, or by audio inputs such as oral command interface, plays an increasingly important role. Researchers such as Haage of Lund University used an RGB-D camera-based sensing module to track human movements when implementing robot teaching. The robot is mainly used to install industrial parts and inspection equipment [12]. Li C et al. designed a LeapMotion sensor-based controller for tracking the operator's hand movements to achieve the real-time robot teaching. The end-effector of the robot is actually held for demonstration to teach the robot action. Meanwhile, a neural network (NN)-based adaptive controller has successfully been developed for the remote manipulation of the DLR-HIT II robot hand [13]. In [14], the authors developed a robot learning method by modelling the motor skills of a human operator using dynamic motor primitives (DMP) and integrating the speech recognition, wherein people could easily teach the robot by speaking. However, only a few works [15,16] take advantage of the visual signals- and audio inputs-based robot teaching by combining them, which results in a negative effect on the data transmission latency and the diversity of motor skills.

In this paper, the arm motion detection and speech recognition are combined, and at the same time, with the help of information fusion and pose recognition, a teaching control system for the manipulator is designed [17] so that the manipulator can understand the skills taught by the experimental staff. It can complete the human-robot interaction more accurately and quickly, and then replace the operator to complete the experimental work. Compared with other robot learning methods, our proposed system possesses the following features:

1. The teaching process is simple and does not require operators to have programming skills, which is suitable for direct use by chemical operators.
2. The self-developed desktop manipulator is small in size and suitable for simple desktop chemical experiments.
3. Chemical experimental actions are decomposed into a combination of multiple basic actions, and a combination of visual action recognition and speech recognition is used to improve the accuracy of teaching.
4. Compared with the manipulator used in traditional research, our proposed method employs a low-cost manipulator with lower power consumption but embeds advanced robot learning algorithms.

2. Overall System Design

The teaching control system in this design takes the desktop experimental manipulator as the physical carrier and the Raspberry Pi operating system as the platform. It is mainly aimed at teaching the experimental robot arm to imitate the experimental movements of the human arm through the teaching of the operator in the chemical analysis experimental scene. This is then combined with the grasping pose recognition algorithm of a certain experimental instrument to assist the control, teaching and motion control. The two are combined to complete a set of experimental process combinations.

2.1. System Logical Architecture

As shown in Figure 1, the overall logical architecture of this design can be divided into three levels: From the bottom up, they are the hardware composition, device driver and application software layer. The hardware consists of a desktop manipulator (with three basic degrees of freedom, similar to a human arm), a Raspberry Pi 4B core board, a motor and motor control board, a high-resolution camera, a microphone and a robotic claw. The device driver part includes an audio driver for microphone, USB_Cam camera driver, motor driver and other related programmable logic. The application software layer is the software program running in the Raspberry Pi desktop operating system and the set of motion parameters of the manipulator, which is similar to the Linux operating system. The software program part includes a visual interface program for motor control and teaching process grasp, speech recognition [18] function module, motion detection module and the fusion and matching part of motion information.

2.2. System Function Process

The overall workflow of the teaching control system is shown in Figure 2. In the teaching process of this whole set of experimental procedures, a complete set of experimental procedures is composed of many simple experimental actions, which are called action primitives. The operator needs to dictate his movements while performing the movements. The manipulator obtains information by listening and seeing, and performs fusion verification, so as to understand the movements that need to be performed. That is, the speech recognition module and the action recognition module, respectively, combine the action primitives recognized by the whole set of experimental process actions into the set in order, and then match them with the action groups in the manipulator action set stored in the local action library where the action information is displayed. The identification information in the process is obtained through the information fusion algorithm to obtain a final set of action groups, which are saved or handed over to the manipulator to run and reproduce.

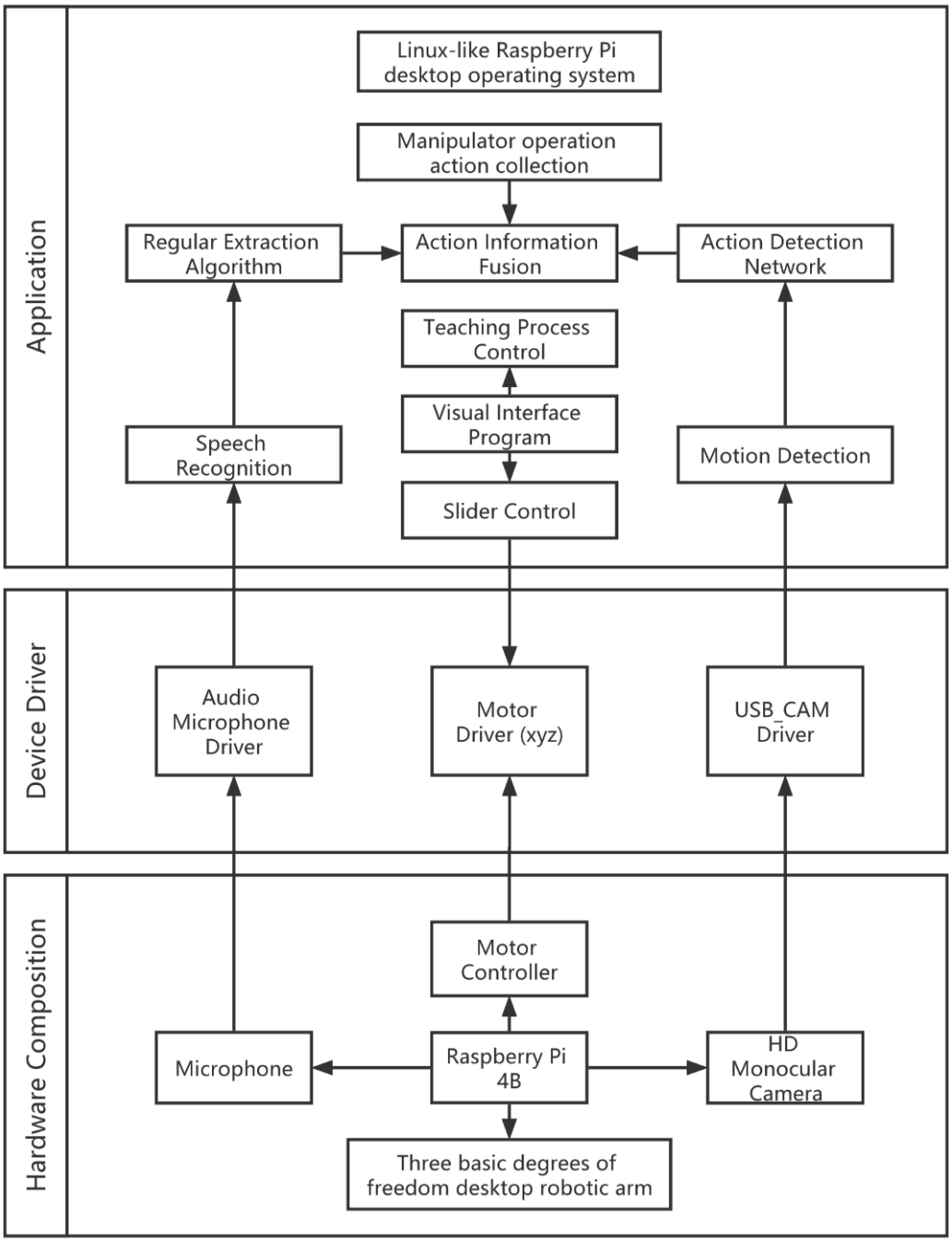


Figure 1. System logical architecture.

The system visualization interface is compatible with Windows and Linux operating systems. As shown on the right side of the figure, the action parameters are stored in an xml file, and the format is neat and simple, which is easy to read and store the action parameters. In the visual interface, three Scale and Radio buttons are set to control the steps and directions of the three motors (x, y, z, respectively), and the action group is defined and stored by sliding and adding actions. Click Start Live Teaching to turn on the camera, and then start the teaching function.

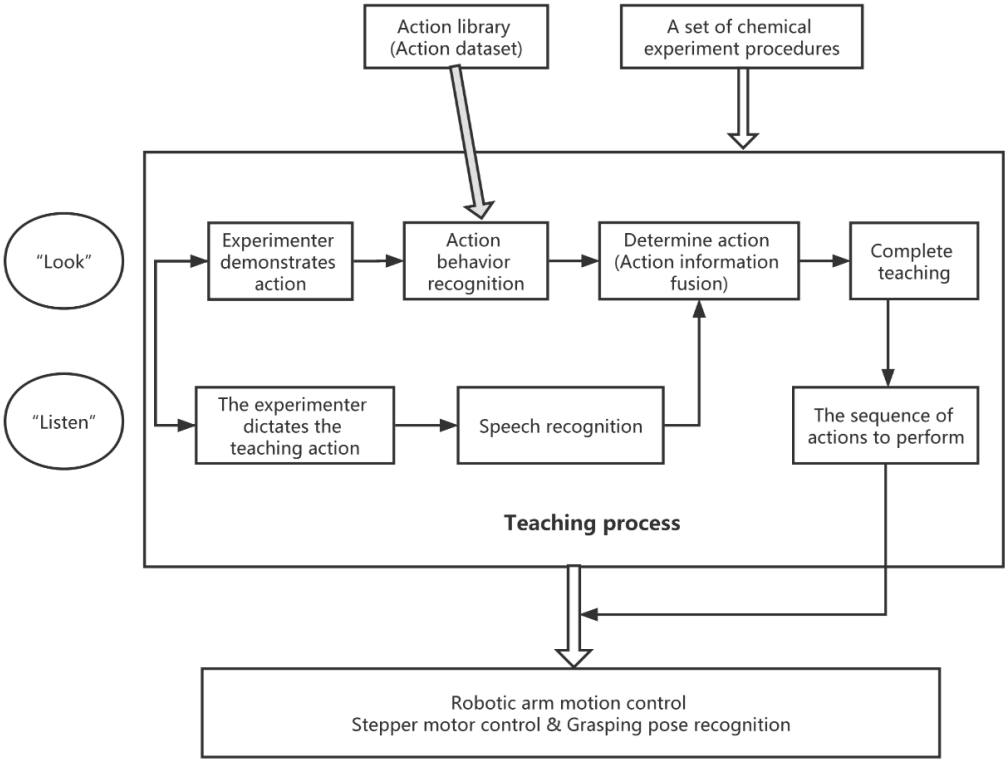


Figure 2. System function flow.

3. The Composition and Performance Parameters of the Manipulator

As shown in Figure 3, the desktop robotic arm in this design has three basic degrees of freedom and can rotate in the space above the desktop. The robotic arm is composed of a Raspberry Pi 4B as the core control board, a motor drive expansion board, a stepper motor drive module, three stepper motors, a 12 V power adapter, and a camera with a microphone. The operations can be picked and placed using robotic arm universal grippers. The maximum payload of the robotic arm is 500 g, and the max reach is 320 mm. The stepper motor adopts a high-torque 42 planetary deceleration stepper motor with a step angle of 1.8°. In the stepper motor drive module, each step (1.8°) of the motor is subdivided into 16 steps for finer control.

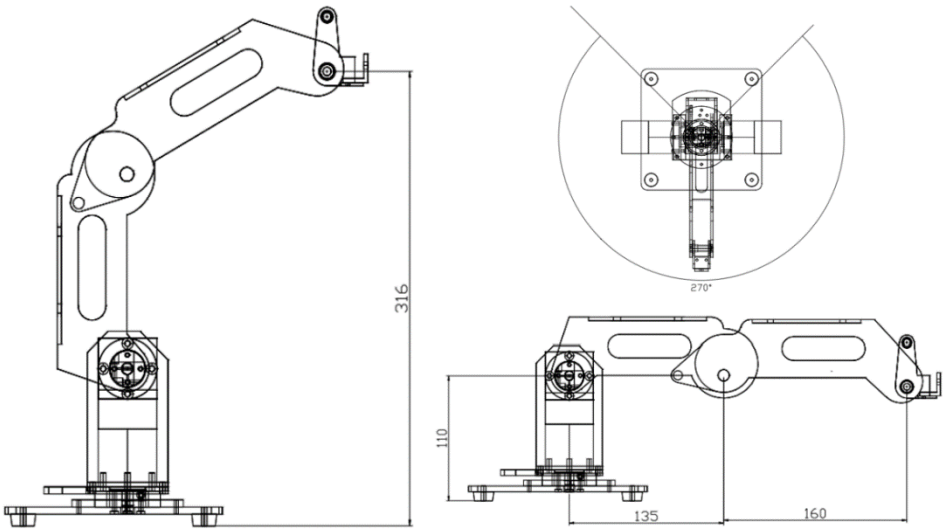


Figure 3. Structure diagram of the manipulator.

3.1. Core Control Panel

The control system runs on a non-customized version of Raspberry Pi 4B, which is essentially a tiny, embedded PC. The Raspberry Pi used in this article has a 64-bit, 1.5 GHz quad-core CPU, 1 GB of memory, and two USB 3.0 and two USB 2.0 communication ports. The programming system is a desktop Raspbian System based on the Linux operating system [19]. The control system controls the microphone and camera through the USB communication port of the Raspberry Pi and controls the self-developed stepper motor and drive module through the GPIO port.

3.2. Stepper Motor and Drive Module

The key element used to control the movement of the manipulator is the motor. There are three stepping motors used in this manipulator, which realizes the motion control of three basic degrees of freedom. The manipulator adopts a high-torque 42 planetary deceleration stepping motor. The theoretical deceleration ratio of the horizontal motion motor is 1:5.18, and the actual measurement is 11:57, which is close to the theoretical value; the deceleration ratios of the stepping telescopic motors of the two arm parts are both 1:19 (the actual measurement is 187:3591). The current of the motor is 1.7 A, the step angle is 1.8°, and its step accuracy is 5%.

As shown in Figure 4, the motor driver adopts an A4988 driver, which can drive the motor voltage of 8–35 V. This manipulator uses a 12 V power supply to power the motor. Among them, each stepper motor drive module outputs two control signals, STEP and DIR, respectively, which are connected to the Raspberry Pi pins to realize the control of the stepping pulse and direction, respectively; the MS1–3 pins are used in this design. Both are connected to a high level, and the corresponding parameters of the interface level and the number of steps is shown in Table 1. Each step (1.8°) of the motor is subdivided into 16 steps to achieve more precise control.

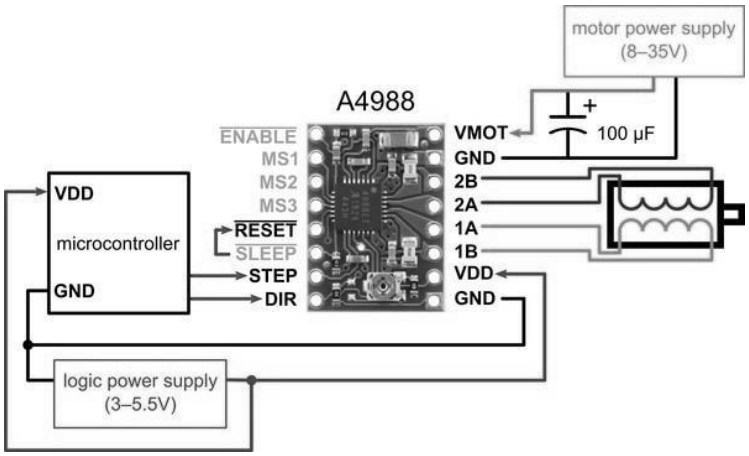


Figure 4. Stepper motor driver module.

Table 1. Corresponding parameters of MS interface.

MS1	MS2	MS3	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	Quarter step
High	High	Low	Eighth step
High	High	High	Sixteenth step

4. Skill Acquisition Module Design

4.1. Motion Detection

At the beginning, we selected the action detection of the skeleton network, but the effect was not obvious, and the earliest dual-stream network also had a very good accuracy in action detection. Based on the dual-stream network, we performed the feature extraction part and the fusion classification algorithm. The next step is to improve and expand the behavioral action recognition data set and apply it in the chemical analysis experiment business scenario.

The action detection part of this paper is mainly based on an improved two-stream convolutional network. This part inputs the preprocessed image information into the network, uses the EfficientNetv2 [20] algorithm to calculate the RGB image and optical flow image features, and then uses the extracted feature information to use linear classification. The SVM [21–24] is used to classify the behavior and obtain the identification information of the action.

As shown in Figure 5, the two-stream convolutional network divides the input video into two channels for processing, one of which is to extract the task arm and scene-related information in the RGB image by the convolutional neural network, and the other is to process the optical flow image information, which is finally normalized and fused by the Softmax function.

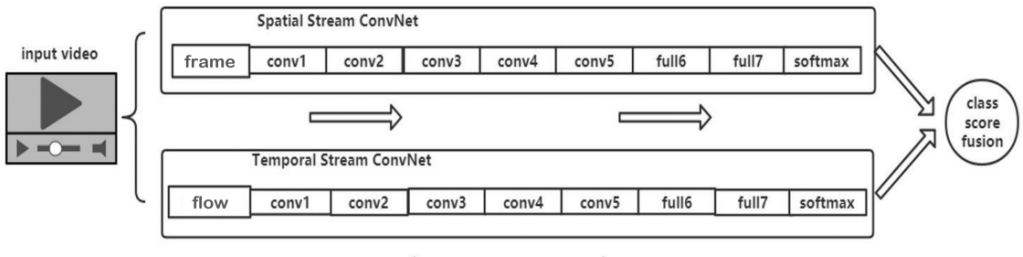


Figure 5. Two-stream convolutional network structure.

The extraction of optical flow images in the network (action video preprocessing) is obtained by gradient-based operations. The key principles of the algorithm (1) are as follows: first, set the image sequence $\boldsymbol{I(x, y, t)}$; the vector $\boldsymbol{X = [x, y]}$. The sequence is extracted from the previous and subsequent frames in a demo video, that is, when the local optical flow image of the video is basically constant. For any $\boldsymbol{Y \in N(x)}$, there are:

$$\frac{d}{dt}\nabla I(X,t) = \frac{\partial \nabla I}{\partial X}\frac{\partial X}{\partial t} + \frac{\partial \nabla I}{\partial t} = H(I)\cdot d + (\nabla I)_t = 0 \tag{1}$$

where $\boldsymbol{X}$ is the $\boldsymbol{x}$ vector, $\boldsymbol{H(I)}$ is the Hesse matrix of the image sequence $\boldsymbol{I}$, and the relationship between $\boldsymbol{X}$ and the offset $\boldsymbol{d}$ is introduced in (2):

$$E(X,d) = \| \ (H(I)\cdot d + (\nabla I)t) \ \|^2 \tag{2}$$

Setting the derivative equal to 0 yields (3):

$$d = -(H^T(I)H(I))^{-1}(H^T(I)(\nabla I)_t) \tag{3}$$

The above process can be summarized as analyzing the changes of the pixels in the video image on the timeline and the correlation between adjacent frame images, finding the corresponding relationship between the previous frame and the current frame, and calculating the motion information (the offset is a kind of motion information), followed by drawing the optical flow image.

In the calculation feature part, as shown in Figure 6, compared to the previous Efficient-Net [25] algorithm, EfficientNetv2 uses Fused-MBConv to replace the MBConv structure,

that is, the conventional 3 × 3 convolution is used to replace the 3 × 3 depth convolution in MBConv and 1 × 1 convolution to improve the calculation speed of the network.

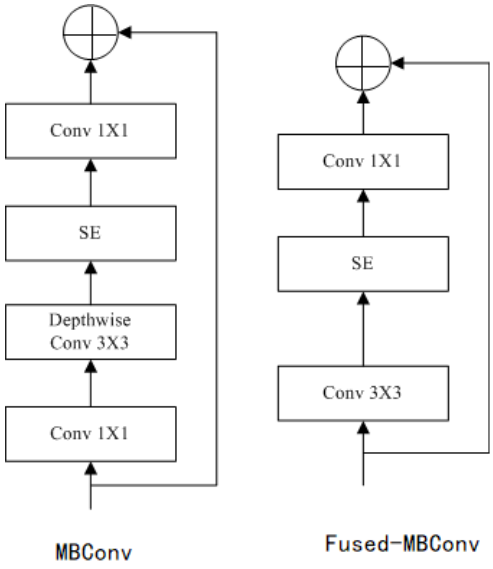


Figure 6. Schematic diagram of the improvement of EfficientNetv2 structure.

To obtain the feature information of the RGB image and optical flow image, it needs to be classified and verified. Support vector machine (SVM) is a binary classification model used to solve the separating hyperplane that can correctly divide the training data set. It also has the largest geometric interval. As shown in Figure 7, $\boldsymbol{w} \times \boldsymbol{x} + \boldsymbol{b} = 0$ is the separation hyperplane. There are generally many such hyperplanes, but the separation hyperplane with the largest interval is indeed the only one. For the optimal value among them, Formula (4) can be used to select, which is as follows:

$$\max_{w,b}(\min_{x1}\frac{y_i(w^T x_i + b_i)}{|w|}) \tag{4}$$

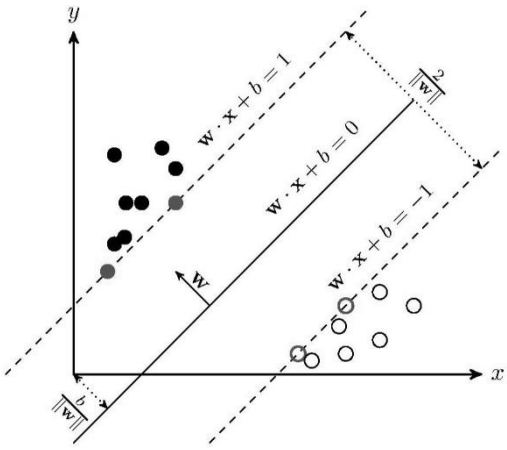


Figure 7. SVM principle.

The overall structure is shown in Figure 8. The RGB and optical flow feature extraction parts in the dual-stream convolutional network use lightweight Efficientnetv2 to perform convolution and pooling, respectively, and then combine the actions given by the SVM classifier for the two branches. The information is classified and, finally, identified action information is given. The data in the experiment are a self-made data set for chemical analysis. In the laboratory environment, a fixed camera is used to record the behaviors

of stirring, picking up the experiment, putting down the experiment, taking liquid, and mixing liquid. This behavior was recorded 50 times. The video duration of the dataset is controlled within 10 s, and the recorded video changes in lighting, background, and occlusion, forming 10,000 chemical analysis action videos, with an average of 10 clip videos per video. The video format is 320 × 240, 25 fps and the audio is saved as a wav format file. The data set has a huge number of video frames, and there are data redundancy, interference, etc., which have a great impact on training and learning. Using the improved two-stream convolutional neural network, the recognition accuracy can reach 92%, which is improved in this experiment. Compared with other methods, the accuracy of the latter method is improved, but the training process still takes a significant amount of time.

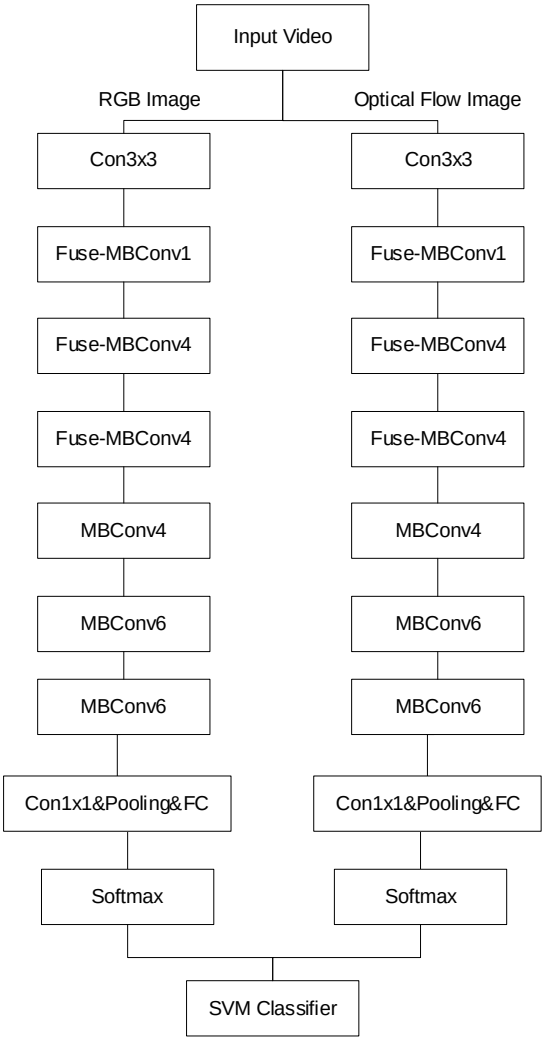


Figure 8. Action detection network structure.

After obtaining a complete set of identification information on the action, it is necessary to determine the execution sequence of the manipulator action of the module and obtain the executable action of the manipulator. First, design and name the motion primitives of the manipulator and store all the designed motion primitives in the library. The action information detected above is stored in a sequence, and the action primitives in the matching directory are used to determine the action primitive sequence in the current time. In each teaching process, the program matches the sequence combination composed of multiple action primitives, and then search for the associated actions in the manipulator action group library. The recognized action numbers are connected in series to obtain a sequence, and this sequence is used to find a complete match or the closest action group;

these are identified as the recognized experimental actions and given a coverage. After the program finds the action group with the greatest match, it continues to match with the action group in the library and stores the relevant information that the action group coverage is higher than 50% in the matching process.

4.2. Speech Recognition and Keyword Extraction

Teaching is a process of speaking while doing. The operator dictates the current action during arm movement, which requires the addition of voice technology. The key to speech technology is to process natural language, recognize speech and generate text, so that machines can listen, speak, understand and think [26]. This paper uses Baidu's real-time speech recognition, which is based on Deep Peak2 end-to-end modeling, transforms the received audio stream into text characters in real time, and then uses regular expressions to extract action keywords. The flow chart is shown in Figure 9.

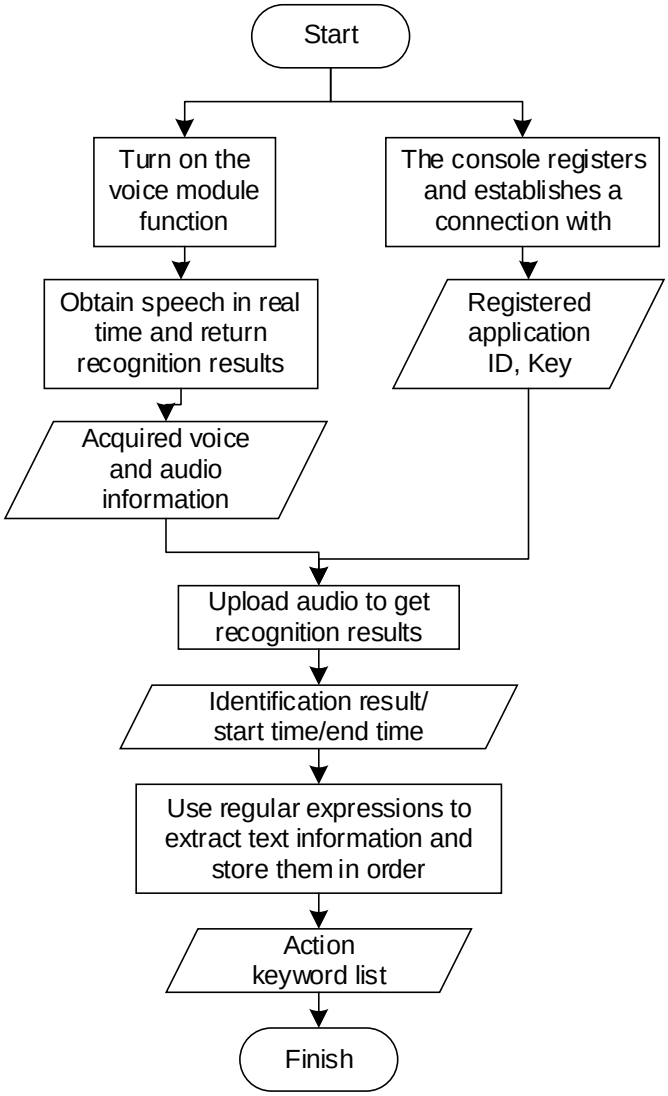


Figure 9. Flow chart of speech recognition and keyword extraction.

In this process, since it is speech recognition in chemical experiments, it is necessary to supplement the training set text corpus for many terms in chemical experiments and the speech data required by the current experimental process. The corpus data include the format of the speech files, name and text information. The corpus data set summarizes about 55 min of relevant identification content, and its audio files are converted into a direct binary sequence PCM file format after analog-to-digital conversion, which realizes

the digitization of the sound and deletes the file header and end marks that differ from other file formats, in order to facilitate the concatenation of files. After supplementing the corpus data set, the accuracy of speech recognition in experimental business scenarios can be effectively improved by 7–15%.

In the above speech recognition process, the program needs to read the action file keywords manually set earlier in the text in advance and write them into memory variables. In each teaching process, the keywords recognized by the speech are matched to the text. If the match is correct, the identifier number IdentNumber is recorded immediately and stored.

Summarize all the matched speech keyword numbers in the whole teaching process in order, similar to the action detection and matching part; search for the action group of the robot arm; and search for a complete match or the closest action group. This is the speech recognition to action group, which retains the coverage information and records other related action group information with a coverage higher than 50%.

4.3. Audio and Video Information Fusion

Since the manipulator teaching program in this paper is mainly run-in embedded devices, in some scenarios, mobile terminals and embedded devices are much inferior in configuration, computing power, and device performance compared to servers or PCs [27]. Under such restrictive conditions, it is difficult to achieve high-speed and accurate recognition algorithms and teaching methods. Moreover, teaching is a dynamic process, which requires the continuous recognition of speech and actions. In this process, recognition failure will inevitably occur and affect the accuracy. Therefore, this paper draws on the information fusion of sensors [15,16,28] and writes a highly targeted algorithm to combine the key information obtained by the two separate modules of action detection and speech recognition, so as to improve the accuracy of skill acquisition and save performance.

The video recognition Information and speech recognition information obtained during the teaching process of the manipulator are an action group and its coverage. In the case of the most ideal running effect, the length of the action group given by the video and the voice is equal, and the data are shown in Table 2 as an example. The two modules in Table 2 list the highest coverage information ***ACT_G0*** generated by the current teaching process and other action coverage information greater than 50% coverage.

Table 2. Matching degree distribution table.

	ACT_G0	**ACT_G1**	**ACT_G2**	**ACT_G3**	**ACT_G4**
Video module	0.8	0.75	0.7	0.6	0.5
Audio module	0.8	0.7	0.7	0.6	0.6

If, in this teaching process, the two modules have the same maximum coverage action group—that is, the video and voice parts have selected the same action group with the highest matching degree—then the action can be regarded as a teaching action and does not need to follow the algorithmic process of fusion. The action can also be considered as being outside of this low-probability case.

After analyzing and testing the common fusion of confidence-based independent classifiers, this paper moves the fusion point to the action group category. The action group sequence given by the acquisition module is fused.

Algorithm idea: Define the ***ACT_G0*** of the video module part as ***VG0***, and then define it as ***VGn*** in turn. Similarly, the audio module part is defined as ***AG0***, ***AG1***...***AGn***. Compare ***AG0*** with 1–***n*** in the ***VG*** part and find a ***VGx*** that has a similarity of 100% with ***AG0***, according to the positive sequence—that is, if more than 50% of the ***VG*** matches ***AG0***, then record and store the coverage product of ***AG0*** and ***VGx*** (***AG0*** × ***VGx***, $0 < \boldsymbol{x} \leq \boldsymbol{n}$). The same is true for the ***AG*** part. The above two parts of the results are compared to the action group that outputs the optimal matching value (weights are equally divided), as shown in Formula (5):

$$FG = Mag[(AG0 \times VGx), (VG0 \times AGy)]x, y \in [0, n] \quad (5)$$

Among them, Mag is the function of the action group to find the maximum value, and ***FG*** is the action group of the robot arm that obtains the maximum value after the coverage rate is multiplied.

When the action group whose similarity with ***ACT_G0*** is 100% in the above process is empty, enter the following search algorithm: there is a similarity (<100%) between any two sets of actions, and the product of the original coverage is used to equalize the two. The similarity between the two is obtained, the correlation value of the two is obtained, and all the correlation values are aggregated to output the maximum action, that is, the maximum value is obtained by the fusion of two actions with different coverage, and then the two actions. In the group, select the action group that is closest to the original set of action primitive sequences (with the largest coverage), which is the final teaching action. The above process is shown in Formulas (6) and (7).

$$FQ(x,y) = AGx \times VGy \times \text{Fit}(AGx, VGy) \quad x,y \in [0,n] \tag{6}$$

$$FFG = MAX\_G\{MAX[FQ(x,y)]\} \quad x,y \in [0,n] \tag{7}$$

Among them, in Formula (6), fit is the similarity between two sets of action groups, ***FQ*** is the product of a pair of coverage and multiplied by the action-related value of the similarity between the two. In Formula (7), all the two modules are action information fusion; output a pair of actions with the optimal correlation value, and then select the action group with the highest coverage with the original action primitive sequence, that is, ***FFG***. In this way, the verification and fusion of the video module and the relevant action information of the audio module are completed, and the manipulator determines the final motor skills.

5. Manipulator Motion Control

The D-H method is usually used to build the model and analyze the motion of the mechanical arm. The D-H method is a common kinematic solution method in the field of robotics, which is beneficial to analyze and establish the kinematic model of the robotic arm and calculate the forward and inverse solutions. Through D-H modeling, the transformation matrix between each joint can be obtained, so as to obtain the transformation matrix from the base coordinate system to the claw coordinate system and the position and attitude of the end of the manipulator.

As shown in Figure 10, the specific method of establishing the link structure coordinate system for the manipulator is as follows: where $j-1$ and j represent two links, $j-1$, j and $j+1$ represent three axis joints, and the axis joint coordinates. The x-axis, y-axis, and z-axis of the system follow the right-hand rule. Among them, $\boldsymbol{a}$ is used to indicate the length of the connecting rod, $\boldsymbol{\alpha}$ is used to indicate the rotation angle of the connecting rod, $\boldsymbol{d}$ is used to indicate the offset distance of the connecting rod, and $\boldsymbol{\theta}$ is used to indicate the axis angle of the joint.

Table 3 is the attached connecting rod D-H parameters, in which the parameters of each connecting rod are d = 103 mm; a1 = 140 mm; b2 = 160 mm; a3 = 70 mm.

As shown in Figure 11, the joint coordinate system of the three-degree-of-freedom experimental manipulator is established on the basis of the D-H parameter coordinate system, including the three rotating joints of the manipulator and the position of the end effector.

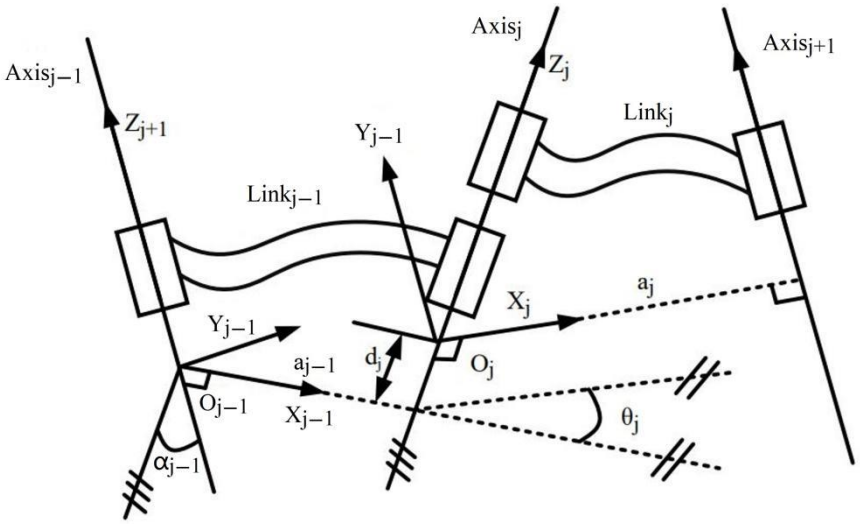


Figure 10. Robot arm link coordinate system.

Table 3. Parameters of the experimental manipulator connecting rod.

Link	Connecting Rod Angle	Connecting Rod Torsion Angle	Corner Range	Connecting Rod Distance	Connecting Rod Distance
i	$\theta_n/(\circ)$	$\alpha_n/(\circ)$	$\theta_i/(\circ)$	d_i/mm	d_i/mm
1	θ_1	0	−135~135	d	d
2	θ_2	90	−15~80	0	0
3	θ_3	0	−20~95	0	0
4	θ_4	0	−90~90	0	0

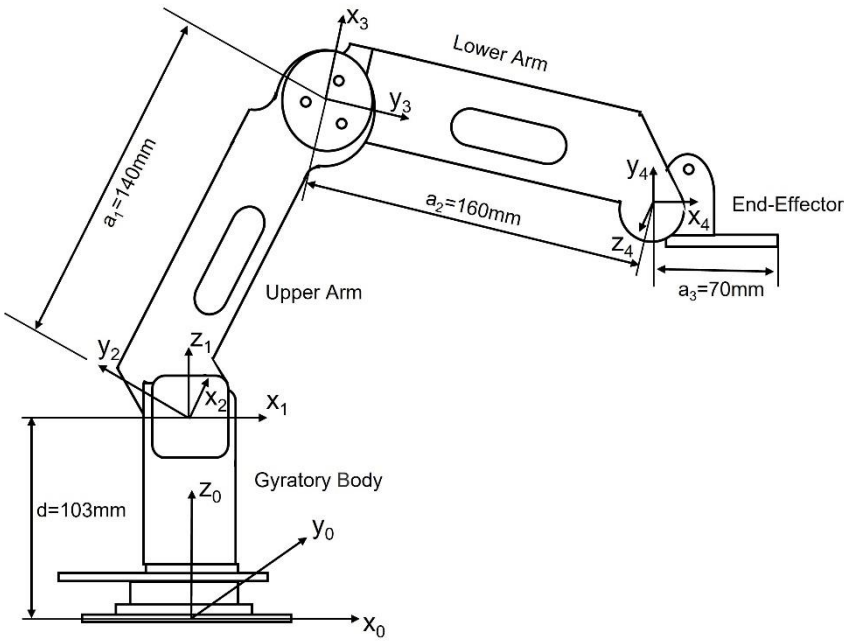


Figure 11. Robot arm model: set the coordinate system according to the D-H method.

The analysis concluded that, for the demonstration function to be implemented in this paper and the designed three-degree-of-freedom robotic arm, it is more suitable to use the geometric solution method in the inverse kinematics solution because our network can get the position where the end actuator or the clamping jaws of the robotic arm are located very simply and precisely, and the robotic arm has only three motors responsible for the

operation of *X*, *Y* and *Z*. Figure 12 shows the simplified spatial coordinate diagram of the drawn experimental manipulator.

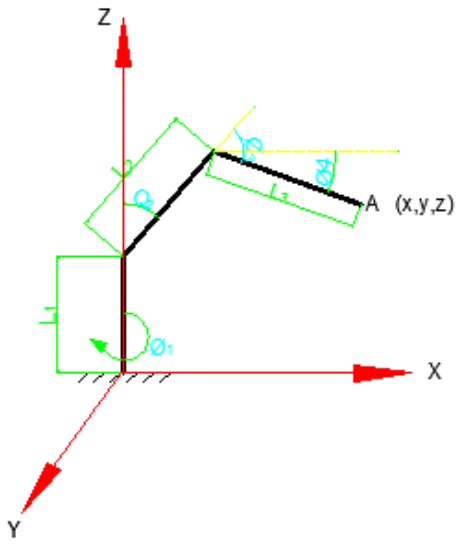
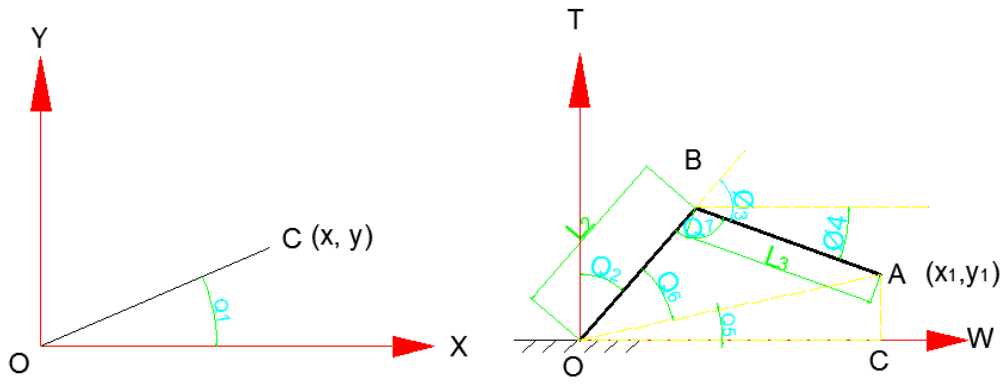


Figure 12. Space coordinate system of the three-degree-of-freedom manipulator.

There are mainly three constants in this manipulator: link length *L*1, link length *L*2 and link length *L*3; three rotation dependent variables: rotation angles *Q*1, *Q*2 and *Q*3; one variable: the general size of the gripper at the end of the manipulator; and a coordinate position (*X*, *Y*, *Z*).

As shown in Figure 13a, looking at it as a plane coordinate system with $\boldsymbol{x}$ as the horizontal axis and $\boldsymbol{y}$ as the vertical axis, the tangent of the connecting rod rotation angle $\boldsymbol{\theta_{cox}}$ is:

$$\tan\theta_{COX} = y/x \tag{8}$$



(**a**) (**b**)

Figure 13. Coordinate system plane. (**a**) XOY plane (**b**) XOZ plane.

Reverse find:

$$\theta_1 = \theta_{COX} = \arctan y/x \tag{9}$$

From the perspective of the XOZ plane of the space coordinate system of the manipulator, as shown in Figure 13b, from the coordinates of point *A* (***x1***, ***z1***), we can know that *OC* = x_1, *AC* =z_1 and obtain:

$$\theta_{AOC} = \arctan z_1/x_1 \tag{10}$$

Similarly, the cosine formula can be used to obtain:

$$\theta_{BOA} = \arctan\frac{{L_2}^2 + {x_1}^2 + {z_1}^2 - {L_3}^2}{2L_2\sqrt{{x_1}^2 + {z_1}^2}} \tag{11}$$

$$\theta_{OBA} = \arccos\frac{{L_2}^2 + {L_3}^2 - ({x_1}^2 + {z_1}^2)}{2L_2L_3} \tag{12}$$

Introduce this into formulas:

$$\theta_2 = \pi/2 - \theta_{AOC} - \theta_{BOA} \tag{13}$$

$$\theta_3 = \pi - \theta_{OBA} \tag{14}$$

$$\theta_4 = \pi/2 - \theta_2 - \theta_3 \tag{15}$$

The corresponding rotation angle of the link *L*2, the corresponding rotation angle of the link *L*3, and the horizontal angle of the link *L*3 can be obtained.

According to the above process, the inverse kinematics solution for the geometric solution of the three-degree-of-freedom experimental manipulator in this paper can be summarized as the flow chart shown in Figure 14:

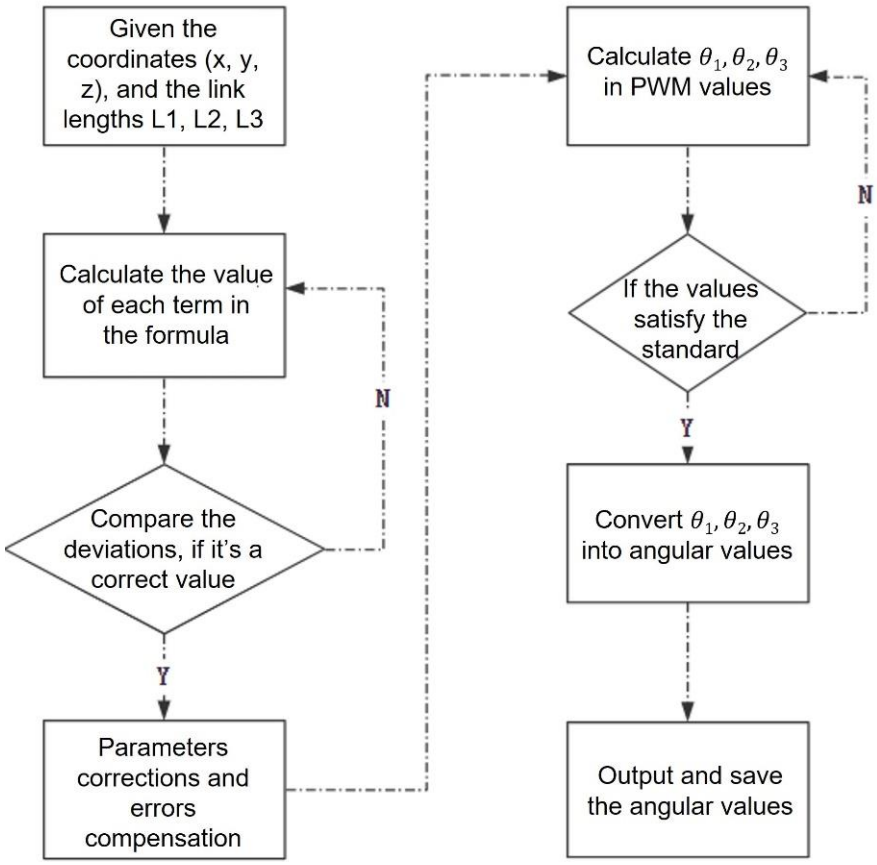


Figure 14. Inverse kinematics solution flow of mechanical arm geometry method.

After the teaching is completed, the program loads the finalized XML action parameter file. The program reads the corresponding direction and step parameters of the *X*, *Y* and *Z* motors in sequence, executes them in sequence, and controls the high- and low-level outputs of the related pins of the Raspberry Pi to control the operation of the motor. Among them, the rotation angle of the robot arm corresponding to the step parameter 1024 is 18°, and the maximum is 10,240.

When the manipulator is in motion, the camera installed near the gripper at the front end of the manipulator can receive real-time image data and transmit them to the Raspberry Pi through the USB port. The system combines it with the grasping pose recognition algorithm [29] for grasping. Five variables are used: ***(x, y, θ, h, w)*** to describe the gripping position and direction of the gripper when the manipulator grips the object. As shown in the rectangular box in Figure 15, ***(x, y)*** is used to represent the center position of the rectangular box; ***θ*** is used to represent the angle between the horizontal axis in the image and the current tilt position of the rectangular box; ***h*** denotes height; and ***w*** is used to represent width.

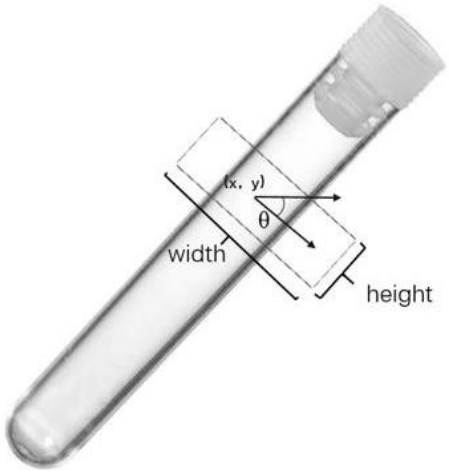


Figure 15. Grasping pose.

When the manipulator reaches the vicinity of the object, the position of the front end of the manipulator needs to be adjusted slightly, according to the position of the rectangular frame. That is, six basic adjustment actions are set: left and right, up and down, and clockwise/counterclockwise rotation. Use the above actions to adjust the mechanical arm to reach the preset position and achieve object grasping.

The grabbing pose algorithm is based on the Cornell Grasping Dataset, and on the basis of the data set, it continues to supplement and train the grabbing positions related to chemical equipment, which compensatively improves the accuracy of grabbing pose recognition.

6. System Performance Testing and Analysis

6.1. Test Experiments and Results

In the information fusion part, there is a requirement on whether the two modules generate the same number of action primitives, so an experiment is set up to record the number of action primitives in each module.

Test experiment: The experiment process was on a fixed test bench. The experimenter simulated the whole set of experimental actions and dictated the actions at the same time. Since this process does not require the movement of the robotic arm, but only observes its teaching process, the experimental program ran in the Windows 10 operating system to record the number of action primitives in the action group it generates. A total of 12 groups of different teaching tests were carried out in the process, and each group of actions was a set of coherent actions composed of 10 action primitives, as shown in Table 4, according to a certain logic and sequence. The 12 groups of action combinations are shown in Table 5. The number of action primitives of a complete set of actions detected by the speech part and the action part obtained in each teaching process is shown in Table 6.

Table 4. Action primitives.

Action Primitive Number	Experimental Action	Action Description
①	Take the test tube	Remove the test tubes from the test tube rack.
②	Shake the test tube	Hold the test tube in your hand and shake it from side to side (it is common to shake the arm repeatedly in the same direction).
③	Stir	Perform circle stirring movements with your arms (usually, the arms are kept in one position to draw a circle).
④	Liquid titration	Keeping your arms balanced and wrist vertical, perform the titration action.
⑤	Rinse the instrument	Hold the test tube in your hand for rinsing.
⑥	Take the glass rod	Retrieve the glass rod from the glass rod holder.
⑦	Gripping solids	Hold the jig to perform the action of pinching the object.
⑧	Place solids	Put the clamped solid into the test tube.
⑨	Place the test tube	Put the test tube in your hand back into the test tube rack.
⑩	Take high test tube	Hold the test tube and raise the arm to the head position.

Table 5. Composition of action primitives in action groups.

Action Group	Action Primitive
Task 1	① ④ ② ⑨
Task 2	① ⑦ ⑧ ④ ⑨
Task 3	① ④ ③ ⑨ ⑩
Task 4	① ④ ③ ⑤ ⑩ ⑨
Task 5	① ④ ⑥ ③ ⑩ ② ⑤ ⑨
Task 6	① ⑩ ⑦ ⑧ ④ ⑥ ③ ⑨
Task 7	① ④ ② ③ ⑤ ⑦ ⑧ ⑨
Task 8	① ⑦ ⑧ ⑨ ① ④ ② ⑨
Task 9	① ⑤ ⑩ ④ ②
Task 10	① ② ④ ③ ⑤ ⑦ ⑧ ⑨
Task 11	① ⑦ ⑧ ④ ③ ⑩ ② ⑤ ⑨
Task 12	① ⑤ ⑨

Table 6. Number of action primitives.

(Number/*n*)	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Reality	4	5	5	6	8	8
Video module	4	5	5	6	6	8
Audio module	4	5	5	6	8	8
	Task 7	**Task 8**	**Task 9**	**Task 10**	**Task 11**	**Task 12**
Reality	8	8	5	8	9	3
Video module	6	7	5	8	8	3
Audio module	7	7	5	7	9	3

The data in Table 6 show that in the teaching process of some simple action groups with a small number of action primitives, the number of action primitives generated by the two modules is basically the same. According to the results, we assume that when the complexity of an action group continues to increase, with the increase in the number of action primitives that comprise the action group, the number of missed and lost ones will theoretically increase with a small confidence range. The proportional probability of the number will gradually decrease from 100% to 81.2%, and in practice, when teaching some conventional action groups, it can basically reach more than 95%—that is, the number of actions that are missed to be recognized is less than 5%, and the overall effect good.

On the basis of the above consistent situation, the coverage of the final action group is shown in Table 7 below:

Table 7. Action group coverage.

Task 1	**Task 2**	**Task 3**	**Task 4**
100%	80%	80%	83.3%
Task 6	**Task 8**	**Task 9**	**Task 12**
75%	71.4%	80%	100%

The results shown in the record table are supplemented with 36 teaching experiments. The teaching program is run in the Raspberry Pi Raspbian desktop operating system, four groups of basic experimental procedures are selected, and seven effective teaching experiments in each group of repeated actions are selected. Record the action group identification coverage in its process. Combining the recognition results of the speech part and the single module of the video part, the correct action primitives identified in the 28 effective teaching experiments (the experiments that are not completely executed due to the personal operation errors of the personnel) are listed in the video, speech and fusion. The comparison of numbers and accuracy rates is shown in Table 8:

Table 8. Comparison of fusion and single-module experiments.

ACT1	**Reality**	**Video**	**Audio**	**Fusion**	**ACT2**	**Reality**	**Video**	**Audio**	**Fusion**
Task 1	6	4	6	6	Task 1	5	5	5	5
Task 2	6	5	4	5	Task2	5	4	4	4
Task 3	6	5	4	6	Task 3	5	4	3	4
Task 4	6	6	6	6	Task 4	5	3	3	3
Task 5	6	3	4	4	Task 5	5	4	3	4
Task 6	6	4	5	5	Task 6	5	5	5	5
Task 7	6	5	5	6	Task 7	5	4	5	5
SUM	42	32	34	37	SUM	35	29	28	30
Accuracy	1	76%	82%	89%	Accuracy	1	82.8%	80%	85.7%
ACT3	**Reality**	**Video**	**Audio**	**Fusion**	**ACT4**	**Reality**	**Video**	**Audio**	**Fusion**
Task 1	4	4	4	4	Task 1	8	6	7	7
Task 2	4	4	4	4	Task 2	8	7	6	7
Task 3	4	2	3	3	Task 3	8	5	6	6
Task 4	4	4	4	4	Task 4	8	7	8	7
Task 5	4	4	4	4	Task 5	8	7	7	7
Task 6	4	4	4	4	Task 6	8	5	5	6
Task 7	4	3	3	4	Task 7	8	6	6	7
SUM	28	25	26	27	SUM	56	43	45	47
Accuracy	1	89.3%	92.8%	96.4%	Accuracy	1	76.8%	80.3%	83.9%

It can be seen from Table 4 that, compared with single-module recognition, the teaching accuracy rate after fusion is improved by about 5−7%, and the result verifies the effectiveness of audio-visual fusion. According to the number of action primitives in the test task and the coverage rate of their action groups, the overall teaching coverage rate after adding information fusion is calculated to be about 87.7%. According to the number of action primitives in the test task and its action group coverage, the overall teaching accuracy is finally calculated to be about 81.4%. This result achieves good results in the field of the non-contact skill acquisition of chemical robotic arms without using support equipment.

6.2. Test Effect and Problem Analysis

At present, the system does not have a clear solution to the inconsistency, and the follow-up research needs to improve and modify the relevant algorithms in the inconsistency.

In terms of accuracy, the stability and accuracy of the manipulator teaching system depends on the accuracy of its action behavior recognition and speech recognition parts. Since it runs in an embedded device, part of its action recognition and matching algorithm needs to be lightweight, so part of the accuracy is sacrificed in the case of improving the recognition speed. The near-field Chinese Mandarin recognition accuracy rate of the speech recognition part is 95%, and the accuracy rate of further keyword matching is higher

than that of simple recognition. The two modules are continuously recognized during the teaching process, and it is inevitable that there will be lost recognition, which will reduce the accuracy. However, based on the information fusion, combining the advantages of the two, the final action coverage can be obtained. Basically, as it is stable at more than 81%, this result is satisfactory at present. After the teaching is completed, the execution effect of the robot arm on the action group basically depends on the accuracy of the teaching process. When performing the robot arm execution experiment, as shown in Figure 16, it is found that each action group can be adjusted during the teaching process. If every action primitive is identified, then the execution effect of the robotic arm can achieve the expected goal. If the action primitives of the action group are different, the results of visual action recognition can often be recognized. However, if there are two actions that are highly similar in the teaching process, misrecognition is easy to occur during visual action recognition. At this time, the accuracy can be further improved by correcting the speech recognition results, which can be seen from the above experimental data table. The different actions, the light occlusion of the test bench and the noise in the environment during the experiment have a certain impact on the teaching process. In the later experiments, a better experimental environment will be built to reduce the external influence on the experiment. Different experimenters have different proficiency in movement and different execution postures and speeds, which also makes visual action recognition more difficult. Therefore, during the teaching process, the movements should be as smooth and distinguishable as possible. The recognition accuracy should also be improved.

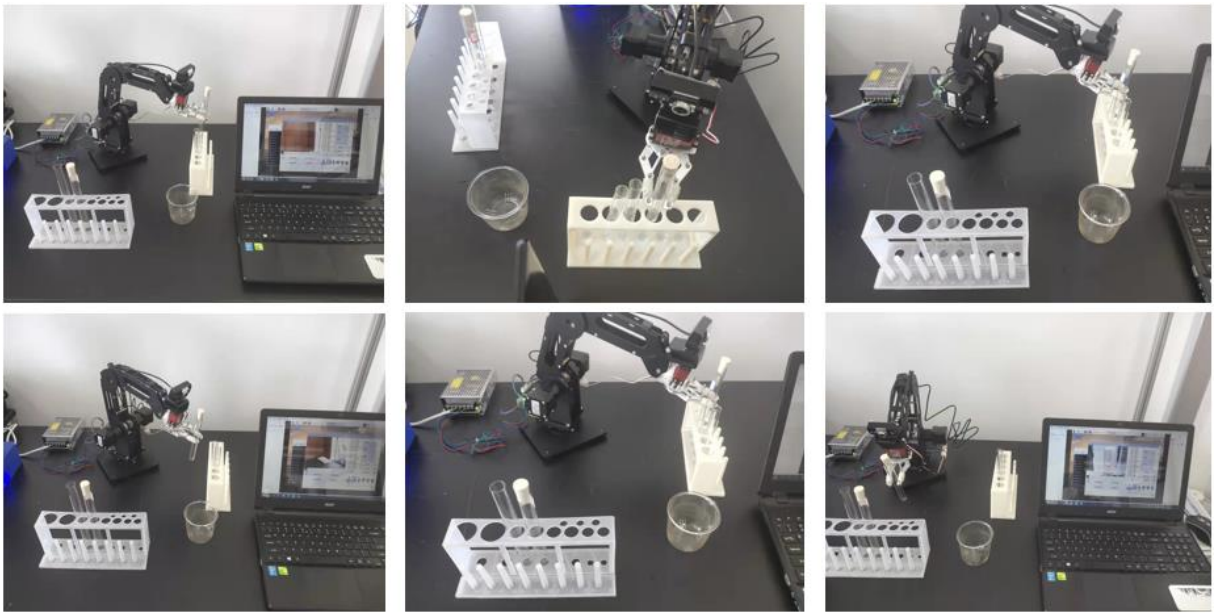


Figure 16. Experimental setup.

In terms of performance and speed, when the program is run on the Raspberry Pi system with 2 GB of RAM, the hit time of a single action primitive (the time it takes to identify the two modules) is within 1 s (the jump time of the action detection program segment is about 0.59 s), while the speech recognition is about 0.82 s. In addition, when the voice part needs to perform multiple loop verifications later, it can be changed to use the language compiled by machine code to run the loop part specifically, create a dynamic link library for this part, and use the external function library types to call, which can significantly improve the loop speed.

When performing the pose-grasping experiment, as shown in Figure 17, because the test tube and other equipment are made of transparent materials, the recognition effect is greatly affected. Try to use label recognition clamping instead (as shown in Figure 17) or use label recognition as a compensatory measure to improve the recognition efficiency.

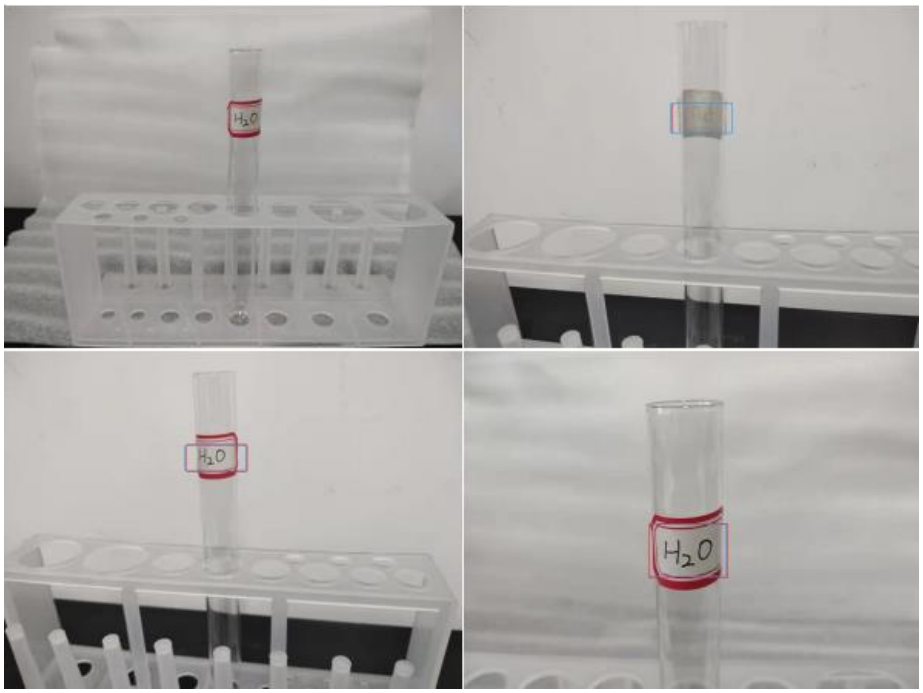


Figure 17. Label location scraping experiment.

7. Concluding Remarks

Operators need to perform various chemical experiments in chemical laboratories; these are cumbersome, and some of them are harmful to the human body. Therefore, it is necessary to use a manipulator instead of an operator to conduct experiments. However, the experimental manipulator that is currently used needs to be programmed by professional manipulator controllers, which is difficult for chemical operators. Therefore, in order to solve the above problems, this paper proposes a simple and efficient manipulator teaching system based on motion detection and speech recognition.

The operator dictates his movements during the experiment. The system uses motion detection to detect the movement of the operator's arm, matches with voice recognition, and uses algorithms related to information fusion to teach the manipulator the motor skills that should be performed. The manipulator grasps objects in combination with pose recognition during the execution process, and completes a set of experimental tasks. The accuracy rate of the system in the acquisition of motor skills can reach more than 81%.

Based on the design and experimental results of this paper, the experimental manipulator is taught and programmed to acquire and execute experimental skills, and there is a certain applicability and feasibility to use the manipulator in chemical analysis experiments. However, some problems in the system were found during the experiment: the transparent material of the test tube and light affected the recognition accuracy, the response speed of the device had a delay, similar behaviors were easily misidentified, and the recognition accuracy needed to be further improved in the actual application process, etc. Since the manipulator is a self-developed manipulator in the laboratory, no model in the corresponding simulation environment has been established, and the subsequent research and development workload is heavy. In the future, we will focus on solving the above problems, create a suitable data production environment in the laboratory, minimize the interference such as light and noise, form models in the simulation environment which are open source for everyone to use, improve the efficiency of later research, and further improve the recognition accuracy and speed. The manipulator will be replaced if necessary to achieve the purpose, but it will still be researched in the direction of low cost and low power consumption.

Author Contributions: Conceptualization, C.L. and X.M.; methodology, C.L. and X.M.; validation, X.C., H.S. and B.W.; formal analysis, X.C., H.S. and B.W.; resources, C.L. and X.M.; writing—original draft preparation, X.C. and H.S.; writing—review and editing, C.L. and X.C.; supervision, X.M.; project administration, C.L.; funding acquisition, C.L.; All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study did not report any data.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lu, Y. Artificial intelligence: A survey on evolution, models, applications and future trends. *J. Manag. Anal.* **2019**, *6*, 1–29. [CrossRef]
2. Zheng, L.; Liu, S.; Wang, S. Current situation and future of Chinese industrial robot development. *Int. J. Mech. Eng. Robot. Res.* **2016**, *5*, 295–300. [CrossRef]
3. Karabegović, I. The role of industrial robots in the development of automotive industry in China. *Int. J. Eng. Work.* **2016**, *3*, 92–97.
4. Lochmüller, C.H.; Lung, K.R.; Cousins, K.R. Applications of optimization strategies in the design of intelligent laboratory robotic procedures. *Anal. Lett.* **1985**, *18*, 439–448. [CrossRef]
5. Liu, H.; Wang, L. Gesture recognition for human-robot collaboration: A review. *Int. J. Ind. Ergon.* **2018**, *68*, 355–367. [CrossRef]
6. Wang, T.M.; Tao, Y.; Liu, H. Current researches and future development trend of intelligent robot: A review. *Int. J. Autom. Comput.* **2018**, *15*, 525–546. [CrossRef]
7. Khan, A.T.; Cao, X.; Li, Z.; Li, S. Evolutionary Computation Based Real-time Robot Arm Path-planning Using Beetle Antennae Search. *EAI Endorsed Trans. AI Robot.* **2022**, *1*, 1–10. [CrossRef]
8. Li, C.; Zhu, S.; Sun, Z.; Rogers, J. BAS Optimized ELM for KUKA iiwa Robot Learning. *IEEE Trans. Circuits Syst. II Express Briefs* **2020**, *68*, 1987–1991. [CrossRef]
9. Allibert, G.; Courtial, E.; Chaumette, F. Predictive control for constrained image-based visual servoing. *IEEE Trans. Robot.* **2010**, *26*, 933–939. [CrossRef]
10. Qian, K.; Niu, J.; Yang, H. Developing a gesture based remote human-robot interaction system using kinect. *Int. J. Smart Home* **2013**, *7*, 203–208.
11. Ajit, A.; Acharya, K.; Samanta, A. A review of convolutional neural networks. In Proceedings of the 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), Vellore, India, 24–25 February 2020; IEEE. pp. 1–5.
12. Haage, M.; Piperagkas, G.; Papadopoulos, C.; Mariolis, I.; Malec, J.; Bekiroglu, Y.; Hedelind, M.; Tzovaras, D. Teaching assembly by demonstration using advanced human robot interaction and a knowledge integration framework. *Procedia Manuf.* **2017**, *11*, 164–173. [CrossRef]
13. Li, C.; Fahmy, A.; Sienz, J. Development of a neural network-based control system for the DLR-HIT II robot hand using leap motion. *IEEE Access* **2019**, *7*, 136914–136923. [CrossRef]
14. Li, C.; Yang, C.; Annamalai, A.; Xu, Q.; Li, S. Development of writing task recombination technology based on DMP segmentation via verbal command for Baxter robot. *Syst. Sci. Control. Eng.* **2018**, *6*, 350–359. [CrossRef]
15. Esteban, J.; Starr, A.; Willetts, R.; Hannah, P.; Bryanston-Cross, P. A review of data fusion models and architectures: Towards engineering guidelines. *Neural Comput. Appl.* **2005**, *14*, 273–281. [CrossRef]
16. Liu, K.; Liu, B.; Blasch, E.; Shen, D.; Wang, Z.; Ling, H.; Chen, G. A cloud infrastructure for target detection and tracking using audio and video fusion. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015; pp. 74–81.
17. Suay, H.B.; Toris, R.; Chernova, S. A practical comparison of three robot learning from demonstration algorithm. *Int. J. Soc. Robot.* **2012**, *4*, 319–330. [CrossRef]
18. Zhizeng, L.; Jingbing, Z. Speech recognition and its application in voice-based robot control system. In Proceedings of the 2004 International Conference on Intelligent Mechatronics and Automation, Chengdu, China, 26–31 August 2004; IEEE. pp. 960–963. [CrossRef]
19. Habil, H.J.; Al-Jarwany, Q.A.; Hawas, M.N.; Mnati, M.J. Raspberry Pi 4 and Python Based on Speed and Direction of DC Motor. In Proceedings of the 2022 4th Global Power, Energy and Communication Conference (GPECOM), Nevsehir, Turkey, 14–17 June 2022; IEEE. pp. 541–545. [CrossRef]
20. Tan, M.; Le, Q. Efficientnetv2: Smaller models and faster training. In Proceedings of the International Conference on Machine Learning (PMLR), Virtual, 18–24 July 2021; pp. 10096–10106.
21. Schuldt, C.; Laptev, I.; Caputo, B. Recognizing human actions: A local SVM approach. In Proceedings of the 17th International Conference on Pattern Recognition (ICPR), Cambridge, UK, 26 August 2004; IEEE. Volume 3, pp. 32–36. [CrossRef]

22. Duan, Y.; Zou, B.; Xu, J.; Chen, F.; Wei, J.; Tang, Y.Y. OAA-SVM-MS: A fast and efficient multi-class classification algorithm. *Neurocomputing* **2021**, *454*, 448–460. [CrossRef]
23. Kaushik, A.; Kaur, G. Review On: Gait Recognition Technique using SVM and K-means with Gait PAL and PAL Entropy. *(IJCSIT) Int. J. Comput. Sci. Inf. Technol.* **2014**, *5*, 3699–3702.
24. Abdullah, D.M.; Abdulazeez, A.M. Machine Learning Applications based on SVM Classification A Review. *Qubahan Acad. J.* **2021**, *1*, 81–90. [CrossRef]
25. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning (PMLR), California, CA, USA, 9–15 June 2019; pp. 6105–6114. [CrossRef]
26. Ermolina, A.; Tiberius, V. Voice-controlled intelligent personal assistants in health care: International Delphi Study. *J. Med. Internet Res.* **2021**, *23*, e25312. [CrossRef]
27. Yunhuan, L.I.; Jiwei, W.E.N.; Li, P.E.N.G. High frame rate Light-Weight Siamese Network target tracking. *J. Front. Comput. Sci. Technol.* **2021**, 1–13. [CrossRef]
28. Xu, R. Path planning of mobile robot based on multi-sensor information fusion. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 1–8. [CrossRef]
29. Redmon, J.; Angelova, A. Real-time grasp detection using convolutional neural networks. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 25–30 May 2015; IEEE. pp. 1316–1322. [CrossRef]