A 3D hand pose estimation architecture based on depth camera

Zhaolong Deng, Yanliang Qiu, Xintao Xie, Zuanhui Lin

Zhaolong Deng, Yanliang Qiu, Xintao Xie, Zuanhui Lin, "A 3D hand pose estimation architecture based on depth camera," Proc. SPIE 12593, Second Guangdong-Hong Kong-Macao Greater Bay Area Artificial Intelligence and Big Data Forum (AIBDF 2022), 125930Z (16 March 2023); doi: 10.1117/12.2671350

SPIE.

Event: 2nd Guangdong-Hong Kong-Macao Greater Bay Area Artificial Intelligence and Big Data Forum (AIBDF 2022), 2022, Guangzhou, China

A 3D Hand Pose Estimation Architecture Based On Depth Camera

Zhaolong Deng[1], Yanliang Qiu[2,] Xintao Xie[2,] Zuanhui Lin[3*]
[1]College of Mathematics and Informatics, South China Agricultural University, Guangzhou 510642, P.R. China
[2]College of Electronic Engineering, South China Agricultural University, Guangzhou 510642, P.R. China
[3]College of Engineering, South China Agricultural University, Guangzhou 510642, P.R. China
[*] Corresponding author: lzh@scau.edu.cn

ABSTRACT

Considering the problem of the inability to obtain accurate depth information in 3D pose estimation, this research attempts to use a depth camera to obtain accurate depth information to solve this problem and achieve good results. In the process of research, it is found that the general object detection and evaluation method is not accurate enough under the framework proposed in this paper, so this research proposes an evaluation method suitable for this framework. A standardizer is also designed to optimize the detection effect while achieving efficient tracking objects. Ultimately, inference time is reduced by 35%. The implementation of this research architecture is open-sourced at https://github.com/DumbZarro/BuddHand.

Keywords-Hand; Estimation; Depth Camera

1. INTRODUCTION

The hand is one of the most complex parts of human body mechanics and anatomy. It realizes human-computer interaction (HCI) based on non-contact gestures, so it can be used as an intuitive and immersive interface for virtual reality (VR), augmented reality (AR), Interactive games and computer-aided design (CAD) will also benefit from recognition tasks, such as motion recognition and sign language recognition; other potential applications include imitation-based robot skill learning and robot grasping[1].

Recently, Facebook CEO Mark Zuckerberg announced that the company's future growth will go beyond building a "metaverse" of social networking applications and related hardware. According to the usual definition of the metaverse, users must use VR and AR equipment to "swim" the metaverse world. However, in the application of the virtual reality industry, there is still the problem of unnatural interaction. Most VR terminals on the market are based on VR glasses, equipped with a remote control handle as an input device. The biggest advantage of VR is its immersive experience, but at present, it only achieves immersive visual effects. Using a handle as an input device not only fails to achieve a tactile immersive experience but instead brings users a sense of fragmentation in the user experience because of the abrupt interaction logic. The use of computer vision technology to realize hand posture estimation, etc. as the input of the VR terminal can realize more free and flexible operations and more natural and smooth interaction logic.

Graphical interface interaction is the interaction mode with the highest acceptance, the highest utilization rate, and the longest use time. However, it still has some defects that cannot be ignored. First, excessive reliance on the human visual system and electronic screen of equipment lead to information overload due to excessive screen and data visualization. Second, it has poor applicability in immersive virtual environments such as AR or VR. Especially when users communicate with users in an immersive virtual environment, text input based on a virtual keyboard will reduce users' sense of experience. Although voice interaction has the advantages of natural interaction mode, low learning cost for users, and wide application range, it also has disadvantages that cannot be ignored. First, linear input mode makes it unable to continuously input and output much content. Second, information recognition is easily affected by the environment. When users are in a noisy environment, recognition is blocked and it is difficult to distinguish different users' voices, which is easy to lead to recognition errors. Gesture recognition enables users to interact naturally with the system without wearing any additional equipment. It and voice interaction are regarded as the best interaction combination in the driving situation. In the immersive virtual space scene, it has a huge User value[2]. However, the technology of brain-computer excuses is neither mature, and the acceptance of brain-computer interface interaction modes is still uncertain. In summary, gesture recognition is currently the best interaction mode in immersive virtual space scenes. Gesture recognition is an important part of human-computer interaction, which has a wide range of application scenarios and important research significance.

Second Guangdong-Hong Kong-Macao Greater Bay Area Artificial Intelligence and Big Data Forum (AIBDF 2022), edited by Kannimuthu Subramaniyam, Proc. of SPIE
Vol. 12593, 125930Z · © 2023 SPIE · 0277-786X · doi: 10.1117/12.2671350

Especially in recent years, with the development of RGB-D cameras (such as Kinect, Realsense), the research of gesture recognition based on RGB-D data is a hot research topic in current gesture recognition[3].

Pose estimation is widely used in gesture recognition, and it can be divided into 2D and 3D. However, 2D pose estimation is difficult to meet the needs of virtual reality, and 3D pose estimation, which can provide richer information, is more suitable for this scene. Compared with the 2D pose estimation, estimating the 3D pose from monocular 2D images is much more challenging. Besides all the challenges in the 2D part, monocular 3D pose estimation also suffers from the lack of sufficient in-the-wild 3D data with accurate 3D annotations, and losing the depth information may cause the inherent 2D-to-3D ambiguity problem[4]. If the network is to be applied to the actual scene, it is necessary to consider improving the inference speed to adapt to the embedded devices with insufficient computing power.

Based on the above content, this research will focus on 3D hand pose estimation based on depth camera (RGB-D).

2. RELATED WORK

2.1 3D POSE ESTIMATION BASED ON RGB

Current works generally can be divided into two classes: (1) direct estimation approaches, and (2) 2D-to-3D lifting approaches. Direct estimation methods infer a 3D human pose from 2D images or video frames without intermediately estimating the 2D pose representation. 2D-to-3D lifting approaches infer 3D human pose from an intermediately estimated 2D pose. Benefiting from the excellent performance of state-of-the-art 2D pose detectors, 2D-to-3D lifting approaches generally outperform direct estimation methods[5].

Because 3D pose estimation suffers from the lack of sufficient in-the-wild 3D data with accurate 3D annotations and losing the depth information may cause the inherent 2D-to-3D ambiguity problem, this study is more inclined to 2D-to-3D lifting approaches, which decompose 3D pose estimation into two steps: 2D pose estimation and lift 2D to 3D.

2D Pose Estimation is generally mature. It not only has a large number of data sets that are easy to customize and annotate to solve the problem of insufficient data but also has many excellent pre-training models that require a smaller amount of data when learning new scenarios. All the tricks and experience in this field can be fully and directly utilized in the field of 3D pose estimation. However, the pose estimation problem seems to be solved, but in fact, all the problems are transferred to the task of 2D to 3D lifting. In other words, the task of 2D to 3D lifting is suffered from the lack of sufficient in-the-wild 3D data with accurate 3D annotations, and losing the depth information may cause the inherent 2D-to-3D ambiguity problem.

The main approach of the task of 2D to 3D lifting is to use the neural network[6] to restore a 2D skeleton to a 3D skeleton. However, The problem mentioned above is exactly what this method is faced with. The result is also estimated based on prior knowledge. Using a depth camera to obtain depth and combine 3d reconstruction techniques to lift dimensions requires less computational power, and both problems mentioned above are solved. Therefore, this study believes that RGB-based algorithms are difficult to solve this problem and need to use RGBD images as input.

2.2 3D pose estimation based on RGB input

3D hand pose estimation is a relatively hot research direction recently, and MediaPipe Hands [7] is a popular study in this research direction based on RGB input. MediaPipe is an open-source application framework for multimedia machine learning models developed by Google Research. This research proposes a real-time 3D hand reasoning architecture that is easy to deploy on the mobile terminal. The architecture uses non-end-to-end learning, and divides the task into two steps: object detection and 3D keypoints detection, and achieves real-time detection by avoiding the use of object detection to save computational overhead. However, the depth predicted by the 3D keypoints network is only a probabilistic estimate or a fuzzy relative relationship. For example, the depth predicted in MediaPipe is only the depth of each keypoints relative to the wrist keypoint. At the same time, because the 3D keypoints detection network is larger and more complex than the 2D keypoints detection network, training a model requires more labeled data. However, because it is difficult to rely on manpower to label depth information, the production of 3D data sets is more difficult than 2D data sets, and the production and maintenance costs are higher. The above problems are due to the perspective phenomenon when the 3D projecting to the 2D, which leads to the lack of depth information in the RGB image. Collecting data in this area often requires professional equipment, and humans cannot obtain specific depth information from plan views alone. This shows that the model needs to have superhuman performance. The z-axis value of the final result is only relative to the z-axis value of keypoints of the wrist.

MediaPipe hand facilitating accelerated neural network inference on the device and synchronization of result visualization with the video capture stream. This is due to the rational design of its architecture. An important idea of its architecture is to predict the possible position of the target in the next frame by the result of the current frame, so as to replace the target detector and greatly reduce the infer time. This design is also used for reference in this study.

2.3 3D POSE ESTIMATION BASED ON DEPTH CAMERA

RGB-D information is that the standard RGB image information introduces depth information, and the depth information can provide the corresponding geometric relationship for the RGB image. In recent years, the development of depth cameras has been rapid, and the accuracy of depth measurement has been further improved. Related products include Intel RealSense series cameras, Kinetic series cameras, and so on. In the past, some egocentric hand pose estimation algorithms based on depth cameras [8]–[12] mostly used some more traditional algorithms or simple convolutional neural networks, which were slightly insufficient in performance.

The depth camera obtains the depth through sensor triangulation. It only needs to perform simple mapping to obtain the accurate depth information of any pixel. It not only effectively reduces the size of the neural network but also helps to apply the previous excellent 2D pose estimation works to the 3D pose estimation.

The depth camera also can directly obtain the point cloud, and the point cloud can provide more information for the work of estimating the pose, but the point cloud is more difficult to train. The model structure is larger than the model structure based on the 2D method. The result of 2D object detection and keypoints detection from RGB images are already very accurate. Using the point cloud as input may not necessarily improve the detection effect, but it must bring more complicated calculations.

2.4 2D POSE ESTIMATION BASED ON RGB INPUT

The field of posture estimation mainly studies the posture of the human body. This research focuses on the hand and studies the posture estimation of the hand. Although the objects are different, the methods are common. Some networks implement hand posture estimation while realizing human posture estimation. Algorithms for multi-person pose estimation can be divided into two types: top-down and bottom-up. For the top-down method, all the objects in the picture are usually found first, and then the keypoints are estimated for each object. For bottom-up, the idea is just the opposite. First, find all the keypoints in the picture, and then assemble these keypoints into an object. At present, there are many excellent related works for 2D hand pose estimation, such as CPM[13] and OpenPose[14]. The heatmap method proposed by the former is widely used in human pose estimation instead of landmark, and the subsequent 2D human pose estimation methods are almost all done around the form of the heatmap. The latter is the representative of bottom-up and the champion of COCO's 2016 human keypoints detection. In recent years, top-down algorithms are more popular, such as DeepPose[15], Stacked Hourglass[16], CPN[17], Simple Baseline[18], HRNet[19], etc., including the CPM mentioned above, The reasoning time of this kind of method is often longer, but its accuracy is higher.

3. ARCHITECTURE

3.1 FRAMEWORK

MediaPipe Hands proposes an architecture that enables it to predict hand posture in real-time. This research refers to its architecture and proposes a new architecture in combination with the depth camera. In the paper, MediaPipe Hands uses hand gestures to predict the position of the next bounding box, which can reduce the use of the object detection module, that is, reduce the computing power required to predict the bounding box, thereby speeding up the inference. This research has followed this idea, while some modifications have been made to the remaining modules. In this study, the end-to-end 3D keypoints detection is divided into two steps. The 2D keypoints are predicted first and then combined with depth to raise them to 3D. Therefore, this research replaces the RGB camera with an RGBD depth camera, and the input adds the dimension of depth. Then the 3D keypoints detector was replaced with a 2D keypoints detector, and a module was added behind it to fuse keypoints and depth information and map them to 3D space. The framework flowcharts is shown in figure 1.

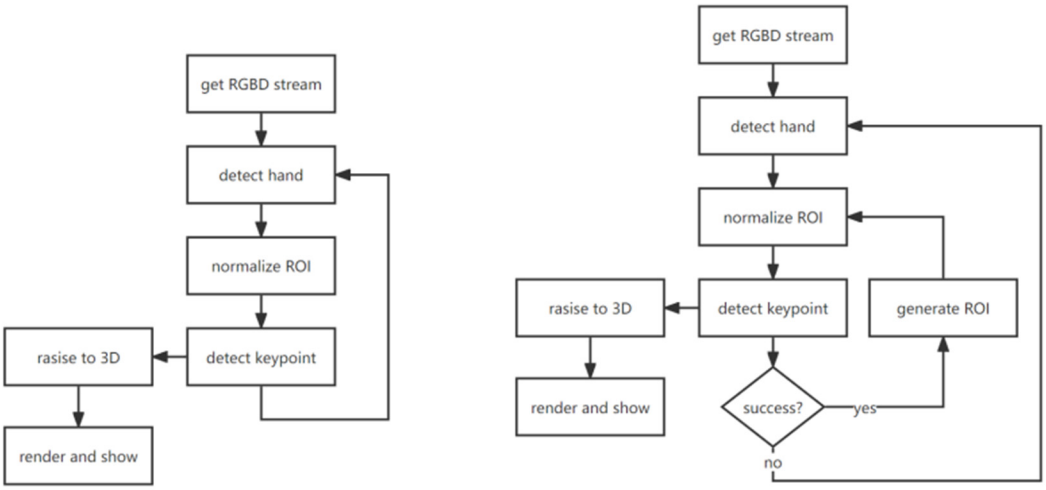

Figure 1. Framework flowcharts

The detection of hands and the detection of keypoints is based on the neural network method, which requires more computing power and relatively slow infer speed, and its infer time takes up the majority of the whole frame's infer time. As shown above, after the design, the frame will only carry out hand detection based on the neural network at the initial stage of the first frame and when the key points cannot be detected. In the rest of the frame, the new ROI is generated directly through keypoint, which to some extent avoids the use of the neural network to predict hand position, saves computing power, and reduces model infer time.

3.2 Data Stream

Hand Detector and Keypoint Detector are the two most important modules in the whole framework, and the inference speed and effect of the whole framework depend on these two modules. However, they are not the focus of this study. These two modules can theoretically be replaced by either frame detector, as long as the inputs and outputs are guaranteed to conform to the framework requirements. With the continuous innovation and development of technology and theory, there may be more excellent detectors working in the future, and the framework has strong compatibility for different detectors. As long as the new work can satisfy the given input to produce output in accordance with the format, the old module can be replaced and the performance and effect of the whole framework can be improved. The input and output formats of different detectors are generally similar and easy to convert to each other. The input and output data flows within the framework are described in more detail below. The framework dataflow diagram is shown in figure 2.

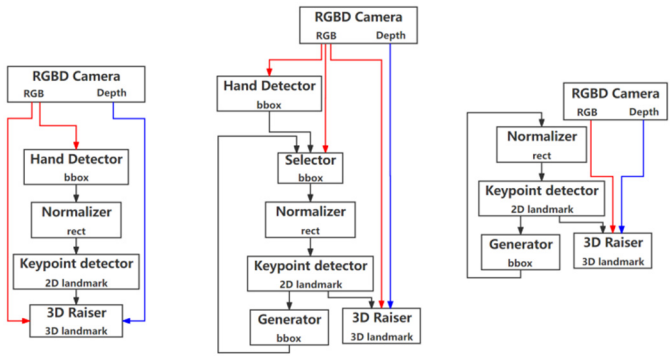

Figure 2. Framework dataflow diagram

We need an RGB-D camera to get RGB images and depth images. The RGB graph is pushed to the Hand Detector for prediction, and one or more BBoxes(bounding box) are obtained. BBox is an array containing X, Y, W, and H parameters, which is used to represent the position of the Hand object in the original picture, and it is passed into the selector. Where x and y are the upper-left coordinates of BBoxes, w and h are the width and height of the BBox respectively. The selector

is a binary selector that determines which module to get the BBox from according to different situations. Without any data changes to the BBox, the selected series of BBoxes are passed into Normalizer and Normalizer standardizes the BBox to a certain extent to form a new array of rectangular boxes. If the input box representation method of keyPoint Detector is different, it can also be converted here. The obtained rectangular boxes are transmitted to the keypoint detector for 2D pose estimation to obtain 2D keypoint of hand. In this study, it is a 21X3 matrix to represent x, y, and z coordinates of 21 key points as shown in figure 3.

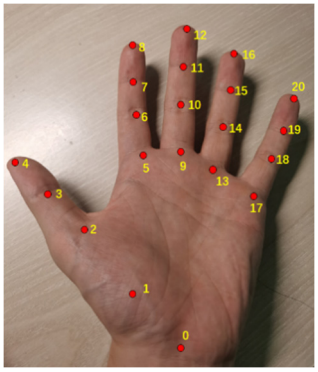

Figure 3. Key points

The 2D keypoint of the hand is passed into the Generator and the 3D Raiser. The Generator is responsible for predicting that a new BBox will be passed into the Selector for selection by 2D keypoint of hand.3D Raiser calculates 3D keypoints by combining depth maps, RGB maps, and 2D keypoints.

3.3 Generator and Normalizer

The higher infer speed of the framework is due to two important modules in the framework, both of which are related to ROI (Region of interest). The ROI generator processes the detected 2D keypoints into a new bounding box (minimum bounding rectangle), which is equivalent to the bounding box detected by the object. These bounding boxes will be used as input and entered into the ROI normalizer for normalization.

This study found that using the bounding box close to the hand will easily lose the information on the boundary (fingertips) and thus affect the prediction result. Therefore, in the ROI standardization module, the boundary of the bounding box is expanded to the surroundings to avoid losing the boundary information and improve the detection effect. At the same time, video-based object detection uses a frame-by-frame detection method, and when the number of frames of the video is high, the hand position is often not far away in the two adjacent frames. By expending the minimum bounding rectangle of the previous frame's hand posture, the next frame's hand remains inside the bounding box. Although the position will be off-center at this time, as long as it does not exceed the bounding box, this hardly affects the prediction of the keypoints detector, thus achieving the replacement goal of the effect of the detector. Since the detection frame of each frame is derived from the circumscribed rectangle of the hand of the previous frame, the position of the generated frame will also move accordingly, so as to achieve the purpose of hand tracking. The tracing process diagram is shown in figure 4.

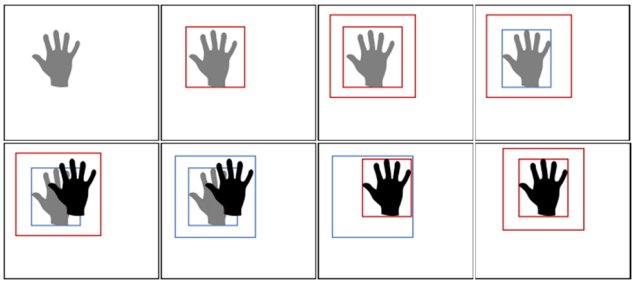

Figure 4. Tracing process diagram

The first picture indicates that the first frame image is obtained. The second figure shows that the object detector is called to predict the position of the image (the bounding box may not fit well at this time). The third figure shows the normalization of bounding boxes. The fourth figure shows the new bounding box generated by predicting 2d key points through the keypoint detector. Five pictures indicate that the next frame of image is obtained, at which time the hand target

has shifted. The sixth picture shows the standardized bounding box. It can be seen that the new hand is still in the standardized bounding box. The seventh figure shows the new bounding box generated by predicting 2d key points through the keypoint detector. The eighth figure shows standardizing bounding boxes.

3.4 3D RAISER

Pixel coordinates are the horizontal and vertical coordinates of pixels in a frame, usually with the origin in the upper left corner. The camera coordinates are the coordinates in the coordinate system with the camera as the origin.3D Raiser's job is to convert known pixel coordinates and depths into camera coordinates in the real world. In order to realize the conversion between pixel coordinates and camera coordinates, we can use some knowledge related to 3D reconstruction. The relationship between pixel coordinates and camera coordinates is as follows:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{f_x X_c}{Z_c} + c_x \\ \frac{f_y Y_c}{Z_c} + c_y \end{bmatrix} \tag{1}$$

After transformation, it is not difficult to get the following formula:

$$X_c = \frac{(v - c_x) * Z_c}{f_x} \tag{2}$$

$$Y_c = \frac{(v - c_y) * Z_c}{f_y} \tag{3}$$

$$Z_c = depth \tag{4}$$

(u, v) are the coordinates of the key points of the hand in the pixel coordinate system. (Xc, Yc, Zc) are the coordinates of the key points of the hand in the camera coordinate system. (fx, fy) are focal lengths in pixels. (cx, cy) represents the principal point (the principal point is also abbreviated ppx,ppy in some places), which is the intersection of the perpendicular line between the center of photography and the image plane. The principal point and focal length are determined by the camera, and the pixel coordinates are derived by prediction, and the depth coordinates can be obtained by the depth camera. Given the above values, the coordinates of the key points of the hand can be calculated in the camera coordinate system. Due to the need to reconstruct 3d space, camera coordinates are usually further converted into world coordinates. However, in virtual reality interactive scenes, camera coordinates are often enough to meet the requirements.

3.5 DATASET

In the process of implementation, this research used the following data sets for training and evaluation:

- EgoHands[20]: This data set contains 48 videos about two people performing complex activities taken from a first-person perspective. The research intercepted some frames in the video and annotated them. The data set has 4800 pictures containing 15,000 annotation information. Most of the marked hands are covered or closed, which is relatively difficult.
- OneHand10k[21]: This data set contains 10K RGB images. The characteristic of this data set is that each image contains only one hand and the hand is usually close to the camera, occupying a relatively large picture.

EgoHands is used to train the YOLOX model with a small amount of data. With some data enhancement methods used, overfitting begins in about 30 epochs. However, due to the following characteristics of the data set, it fits well with the application scenarios of AR/VR devices. The first is the first-person perspective, which is the same as that captured by the integrated device. The second is that the collected data is the interception of the video. The difference between the intercepted frame and the captured picture is that the intercepted frame has dynamic blur as shown in figure 5, which can make the model perform better in video recognition.

Figure 5. Dynamic blur frame of EgoHand dataset

Finally, because the content of the video shooting is a scene of complex activities, and the hands are mostly covered or closed. These are difficult problems in the field of hand pose estimation. There are a large number of such objects which is shown in figure 6 that help improve the recognition effect of the network. To facilitate the evaluation of the performance of the model, this study converts its annotations into COCO format for training and uses OKS (Object Keypoint Similarity based mAP) to evaluate the model.

Figure 6. The sample of EgoHand dataset

And OneHands10k is used to train the keypoints detection model. Because it only contains one hand and occupies a larger screen, the data set has a certain similarity with the object detection results. This study believes that this will make the model better suited to the detection task in this state. The final keypoint detection model used is is a pre-training model trained in onehand10k by mobilenetv2 provided by mmpose. When evaluating its performance, COCO's attitude estimation evaluation standard is also used.

4. HARDWARE AND SOFTWARE ENVIRONMENT

4.1 Development Environment

The computer hardware configuration parameters used in this test are as follows: The operating system was Window10, the processor was AMD Ryzen 5 3600x @ 3.8GHz ×12 with 16 GB of RAM. The Graphics Processing Unit (GPU) is NVIDIA® GeForce® RTX 2060 SUPER™, and the video memory is 8 GB. Install the corresponding driver of the graphics card on the computer hardware device, install the CUDA10.2 general parallel computing architecture and Cudnn acceleration tool library required by the GPU running environment, install the Anaconda environment release, and build the Anaconda virtual environment. Pytorch1.5, Python3.6, pyrealsense2, numpy, yolox, mmpose, opencv and so on are configured in the virtual environment, and the program is run under the integrated development software pycharm.

4.2 Depth Camera

The depth camera used in this study is Intel® RealSense™ depth camera D435. The Intel® RealSense™ depth camera D435 is a stereo solution, offering quality depth for a variety of applications. Its wide field of view is perfect for applications such as robotics or augmented and virtual reality, where seeing as much of the scene as possible is vitally important. With a range of up to 10m and support the output of 1280x720 resolution of the depth of the picture, and ordinary video transmission can reach 90fps and the global shutter sensors provide great low‑light sensitivity. The D435 camera has four round holes on the front, as shown below. From left to right, the first and third are IR Stereo Cameral, the second is IR Projector, and the fourth is a color camera. The appearance of the camera is shown in Figure 7.

Figure 7. RealSense D435

In addition to the ability to capture depth, the camera can also obtain RGB images. The RGB camera uses roller shutter sensor technology. The image resolution can reach 1920 × 1080 frames, and the frame rate is 30 frames per second. The camera is 9 cm long and 2.5 cm wide, which is convenient for integrated to various devices.

4.3 IMPLEMENT DETAIL

When implementing this architecture, this research uses the recently released object detector YOLOX [22] to replace the original object detector SSD[23], which compromises speed and accuracy and further improves the trade-off performance. This study uses the hand data set EgoHands to train the YOLOX-s model, and the training results are shown in figure 8.

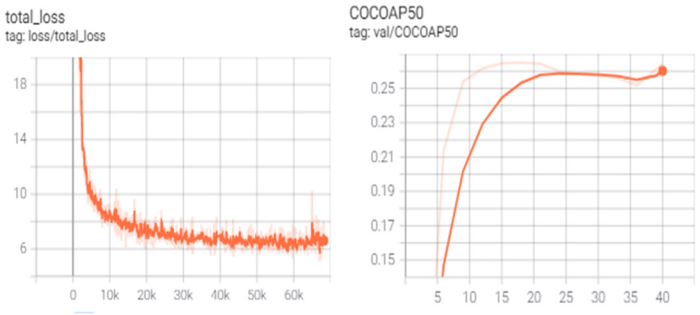

Figure 8. Loss and AP

It can be seen from the figure that the model begins to overfit at about 30 epochs. At this time, the detected AP is not high, and there is a certain gap between the official reference AP, and it takes about 300 epochs to reach the official standard. On the one hand, it is because the number of training iterations is not enough. On the other hand, this research believes that this is due to the difficulty of the data set itself. The details will be expanded below. However, when this research is tested in an actual environment, a good detection effect can be achieved by setting a lower confidence level. Since this research focuses on the improvement of the architecture, it does not further expand and optimize the data set. However, the separation of test results and actual results makes this study further explore and study the rationality of the evaluation method.

TABLE 1. TRAINING RESULTS

	IoU	***area***	***maxDets***	***Value***
AP	0.5:0.95	all	100	0.267
	0.50	all	100	0.352
	0.75	all	100	0.318
	0.50:0.95	small	100	0.058
	0.50:0.95	medium	100	0.164
	0.50:0.95	large	100	0.280
AR	0.50:0.95	all	1	0.272
	0.50:0.95	all	10	0.798
	0.50:0.95	all	100	0.798
	0.50:0.95	small	100	0.129
	0.50:0.95	medium	100	0.560

It can be seen from the table 1 that the reason for its good performance is that the hand is close to the camera when testing the effect of the model in the actual environment, and the recall rate of the model for the large object is relatively high so that all objects are detected. At the same time, this study also found that the false positive objects (ears, arms) detected by individual objects were discarded without detecting keypoints, so the overall result was fewer false-positive results. Therefore, this study believes that misdetection will not have a great impact on the overall task. Under the architecture proposed in this study, the object detector only works when it is initialized and when the object is lost. The rest of the time is the ROI generator to generate the object as the input of the keypoints detector. At the same time, the False detection part will not be output when the keypoints detection module cannot identify the keypoint. It can be seen that a small amount of false detection by the detector has little effect on the overall task. On the contrary, the recall rate is particularly important under this structure. Therefore, simply using AP to measure the slightly unreasonable part of the model will result in a certain deviation between the detection result and the actual effect. When calculating AP, the recall rate should be given more weight. Therefore, this research proposes the following formula to replace the traditional AP to evaluate the object detection model under this architecture.

$$f(m,n) = \sum_{k=1}^{N} \max_{\tilde{k} \geq k} P^m(\tilde{k}) \Delta r^n(k)$$

In the above formula, P is the precision rate and Δr is the increment of recall. The formula replaces the original recall with the quadratic power based on AP calculation by interpolation method, which helps to improve the influence of recall rate on AP. The second power could have been any other natural number, but it is important to note that the higher the degree, the more difficult it is to reach the same threshold. The equation graph of different powers is shown in figure 9. Even so, the evaluation formula cannot fully reflect the merits and demerits of an object detector. The inference speed should also be considered, and individual errors can be ignored at a high enough speed.

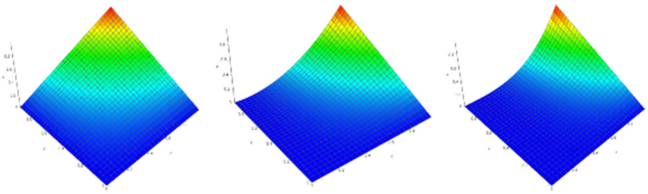

Figure 9. Equation Grapher of different powers

The keypoints detection module of this research uses a heatmap-based method. Since the reasoning speed of the entire model depends on this module, this research chooses to use MobileNetV2[24] as the skeleton network of the keypoints detection module to extract features, and to improve the reasoning speed as much as possible.

This study uses the Intel RealSense D435 depth camera to obtain RGB images and depth images and increases the input of the depth image dimension. From the original 2D input, it has increased to 3D, making 2D keypoints that can be combined with depth maps to get 3D keypoints.

4.4 Model Training and Testing

In this study, the idea of Divide and conquer was adopted in the training of the model, that is, the object detector and keypoint detector were trained separately. Generally speaking, this may lead to poor performance due to the different distribution of training data of the two models. However, this problem was taken into account when network models and data sets were selected in the early stage so that the training data of each model was as similar as possible to the display scene, so the final results were also in line with expectations. However, it is a pity that the data set capacity used is relatively small. In order to face the problem of small data set capacity, we used transfer learning and data enhancement in the process of training to solve this problem.

Transfer Learning Definition is the ability of a system to recognize and apply knowledge and skills learned in previous domains/tasks to novel domains/tasks. The goal is to make the most of the previously annotated data while ensuring the accuracy of the model for the new task. The low-level knowledge of two similar tasks can be shared. For example, the grain knowledge in the target recognition task can be applied to different targets. Such knowledge is usually stored in the parameters of the first few layers of the neural network, so freezing technology appears. Freezing the

parameters of the first few layers of the model trained by others does not change to achieve knowledge sharing, and retraining the parameters of the last several layers to achieve the effect of completing new tasks. Such techniques can reduce the amount of data required for model fitting.

Contrary to the idea of transfer learning, Data Enhancement technology focuses on obtaining more Data to make up for the lack of training image Dataset. In order to obtain more data, we only need to perform slight image processing on the existing data set. Subtle changes, such as flips, translations, rotations, etc., will be recognized as a different image by the network and double the amount of data at a low cost. Although the additional data will cause uneven data distribution, it can enhance the robustness of different scenarios of the model. They can also be divided into offline augmentation, online augmentation, and augmentation on the fly, depending on their timing.

Transfer learning was used in the training of YOLOX, and this study used the official pre-trained model for further training. This study also used flips, Translations, rotations, Shear, HSV gamut transformations, Mixup, Mosaic, and other data enhancement methods. The batch size was set to 8 and FP16 technology was also used, but the model was still over-fitted at 30 epochs. Finally, the trained model performs well under the parameter Settings of confidence 0.3 and non-maximum threshold 0.45. The keypoint detection model is a pre-training model trained in onehand10k by mobilenetv2 provided by mmpose.

5. RESULT

Some of the Inference results are shown in Figure 10:

Figure 10. Inference result

The above is the rendering output results of some 2D keypoints of this architecture, which confirms some of the phenomena mentioned above. For example, due to the existence of the ROI standard device, even the keypoints outside the bounding box can be detected (top left corner), the occluded keypoints are not detected (top right corner), and the following two pictures shows the detection of multiple objects. It should be noted that since this study did not use the data set simulated by 3D modeling, the keypoints that are occluded will not be detected, and the depth of these keypoints cannot be measured. However, this study thinks that the keypoints of the occlusion are not very important, and the shape of the occluded part cannot be judged by light relying on vision. Even if it is predicted, the author thinks that it does not make much sense, because it is unknowable whether the result is correct or not. Of course, if you are obsessed with obtaining such keypoint information, you can replace the 2D keypoints module in the architecture with a keypoints detection model that can estimate the relative depth. The predicted relative depth combined with the depth camera to obtain the absolute depth of the keypoints of the wrist can calculate the depth of each keypoints, but the premise is that the keypoint of the wrist must be detected.

This architecture significantly improves the inference speed. When this architecture is not used, the model detection speed takes an average of 0.05s to infer a frame of pictures, but after switching to this architecture, the inference time drops to 0.03s. The comparison of inference speed is shown in figure 11. In addition, the variance of the reasoning time of this framework is smaller, and the reasoning performance is more stable and smooth.

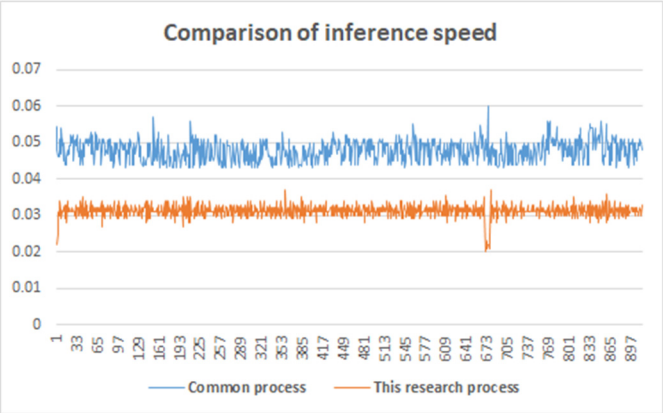

Figure 11. Comparison of inference speed

6. CONCLUSIONS

In this paper, a depth camera is used as input, and an efficient 3D hand pose estimation architecture is designed. This architecture retains the efficient architecture of MediaPipe hands, and at the same time modifies some modules to enable it to detect the absolute depth of the 3D hand gesture. At the same time, it also proposes a more reasonable evaluation method for the object detector under this framework and an efficient tracking method. However, the model's use of depth only stays at obtaining the depth information of keypoints, and fails to further integrate the two modal information to solve the influence of light and dark and color on the object detector and the keypoints detector.

ACKNOWLEDGEMENT

Special Funding Project for Cultivating Science and Technology Innovation for College Students in Guangdong, grant number: pdjh2022a0072.

REFERENCE

[1] L. Huang, B. Zhang, Z. Guo, Y. Xiao, Z. Cao, and J. Yuan, “Survey on depth and RGB image-based 3D hand shape and pose estimation,” Virtual Real. Intell. Hardw., vol. 3, no. 3, pp. 207–234, Jun. 2021, doi: 10.1016/j.vrih.2021.05.002.

[2] “The development status, insufficient and outlook of human-machine interaction.” http://www.lunwenstudy.com/jixiegc/166655.html (accessed Nov. 03, 2021).

[3] “Research on personalized gesture interactive technology based on RGB-D data - CNKI.” https://sso.dglib.cn/interlibSSO/goto/29/+jmr9bmjh9mds/kcms/detail/detail.aspx?dbcode=SNAD&dbname=SNAD&filename=SNAD000001845796&uniplatform=NZKPT&v=TjRX_0fSS4tU9eLrO20Y8KE0S6fApWxkahjhZyrsoC9uPDKGEtTxe0lx6quvVhWKenzTcWuaHzs%3d (accessed Nov. 03, 2021).

[4] W. Liu, Q. Bao, Y. Sun, and T. Mei, “Recent Advances in Monocular 2D and 3D Human Pose Estimation: A Deep Learning Perspective,” ArXiv210411536 Cs, Apr. 2021, Accessed: Jul. 31, 2021. [Online]. Available: http://arxiv.org/abs/2104.11536

[5] C. Zheng, S. Zhu, M. Mendieta, T. Yang, C. Chen, and Z. Ding, “3D Human Pose Estimation with Spatial and Temporal Transformers,” ArXiv210310455 Cs, Mar. 2021, Accessed: Jul. 31, 2021. [Online]. Available: http://arxiv.org/abs/2103.10455

[6] J. Martinez, R. Hossain, J. Romero, and J. J. Little, “A Simple Yet Effective Baseline for 3d Human Pose Estimation,” in 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Oct. 2017, pp. 2659–2668. doi: 10.1109/ICCV.2017.288.

[7] F. Zhang et al., “MediaPipe Hands: On-device Real-time Hand Tracking,” ArXiv200610214 Cs, Jun. 2020, Accessed: Aug. 13, 2021. [Online]. Available: http://arxiv.org/abs/2006.10214

[8] G. Rogez, M. Khademi, J. S. Supančič III, J. M. M. Montiel, and D. Ramanan, "3D Hand Pose Detection in Egocentric RGB-D Images," in Computer Vision - ECCV 2014 Workshops, Cham, 2015, pp. 356–371. doi: 10.1007/978-3-319-16178-5_25.
[9] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt, "Real-Time Hand Tracking Under Occlusion From an Egocentric RGB-D Sensor," 2017, pp. 1154–1163. Accessed: Aug. 12, 2021. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2017/html/Mueller_Real-Time_Hand_Tracking_ICCV_2017_paper.html
[10] S. Sridhar, F. Mueller, M. Zollhoefer, D. Casas, A. Oulasvirta, and C. Theobalt, "Real-time Joint Tracking of a Hand Manipulating an Object from RGB-D Input," presented at the European Conference on Computer Vision (ECCV), 2016, 2016. Accessed: Aug. 26, 2021. [Online]. Available: https://handtracker.mpi-inf.mpg.de/projects/RealtimeHO/index.htm
[11] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Tracking the articulated motion of two strongly interacting hands," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, Jun. 2012, pp. 1862–1869. doi: 10.1109/CVPR.2012.6247885.
[12] D. Tang, T.-H. Yu, and T.-K. Kim, "Real-Time Articulated Hand Pose Estimation Using Semi-supervised Transductive Regression Forests," in 2013 IEEE International Conference on Computer Vision, Sydney, Australia, Dec. 2013, pp. 3224–3231. doi: 10.1109/ICCV.2013.400.
[13] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional Pose Machines," ArXiv160200134 Cs, Apr. 2016, Accessed: Jul. 28, 2021. [Online]. Available: http://arxiv.org/abs/1602.00134
[14] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," ArXiv181208008 Cs, May 2019, Accessed: Jul. 28, 2021. [Online]. Available: http://arxiv.org/abs/1812.08008
[15] A. Toshev and C. Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, Jun. 2014, pp. 1653–1660. doi: 10.1109/CVPR.2014.214.
[16] A. Newell, K. Yang, and J. Deng, "Stacked Hourglass Networks for Human Pose Estimation," ArXiv160306937 Cs, Jul. 2016, Accessed: Jul. 28, 2021. [Online]. Available: http://arxiv.org/abs/1603.06937
[17] Y. Chen, Z. Wang, Y. Peng, Z. Zhang, G. Yu, and J. Sun, "Cascaded Pyramid Network for Multi-person Pose Estimation," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, Jun. 2018, pp. 7103–7112. doi: 10.1109/CVPR.2018.00742.
[18] B. Xiao, H. Wu, and Y. Wei, "Simple Baselines for Human Pose Estimation and Tracking," in Computer Vision – ECCV 2018, vol. 11210, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 472–487. doi: 10.1007/978-3-030-01231-1_29.
[19] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep High-Resolution Representation Learning for Human Pose Estimation," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, Jun. 2019, pp. 5686–5696. doi: 10.1109/CVPR.2019.00584.
[20] S. Bambach, S. Lee, D. J. Crandall, and C. Yu, "Lending A Hand: Detecting Hands and Recognizing Activities in Complex Egocentric Interactions," in 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, Dec. 2015, pp. 1949–1957. doi: 10.1109/ICCV.2015.226.
[21] Y. Wang, C. Peng, and Y. Liu, "Mask-Pose Cascaded CNN for 2D Hand Pose Estimation From Single Color Image," IEEE Trans. Circuits Syst. Video Technol., vol. 29, no. 11, pp. 3258–3268, Nov. 2019, doi: 10.1109/TCSVT.2018.2879980.
[22] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOX: Exceeding YOLO Series in 2021," ArXiv210708430 Cs, Aug. 2021, Accessed: Aug. 28, 2021. [Online]. Available: http://arxiv.org/abs/2107.08430
[23] W. Liu et al., "SSD: Single Shot MultiBox Detector," in Computer Vision – ECCV 2016, vol. 9905, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 21–37. doi: 10.1007/978-3-319-46448-0_2.
[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," ArXiv180104381 Cs, Mar. 2019, Accessed: Sep. 06, 2021. [Online]. Available: http://arxiv.org/abs/1801.04381