AI-Based Keypoint Detection for Remote Robot Operation with Improved Safety

LUDI WANG† TAKESHI OHKAWA†

Abstract: This study introduces a keypoint detection AI system developed through transfer learning, with a pre-trained ResNet-50 neural network serving as its foundational framework. The system is designed to assist in remote robotic arm manipulation by capturing keypoints from network camera images, computing their coordinates, and using joint angles to control arm movements precisely. This research strives to contribute to the enhancement of accuracy and efficiency in remote robotic arm operations, while also highlighting the adaptability of transfer learning in customizing pre-trained ResNet-50 models for specific applications, offering novel technological solutions for robotics control

Keywords: Remote Robot Operation, Keypoint Detection

1. Introduction

As robots increasingly share workspaces and tasks with humans in our daily lives, achieving high precision in robot operations becomes crucial. However, this pursuit must be balanced with concerns about data security and stability to ensure the safety and reliability of robot operations. Sometimes, these issues stem from directly acquired sensor data. Each sensor type has its own characteristics resulting in problems such as occlusions (vision) and drift (inertial) when used in an isolated fashion [1]. The phenomenon of hand drifting typically occurs when hands approach the tracking boundary of the Leap Motion Controller [2]. These issues may result in significant inaccuracies in coordinate data, potentially compromising critical components of the robot, such as transmission systems and motors.

Using the ResNet-50 model, we have developed an AI system for human keypoint detection to achieve real-time estimation of human body poses while ensuring data security and stability for application in robotic arm control. In our research, we prioritize accuracy, and as a result, our system has been extensively trained on datasets to precisely recognize human skeletal coordinates. This significantly enhances overall accuracy and addresses data security and stability concerns.

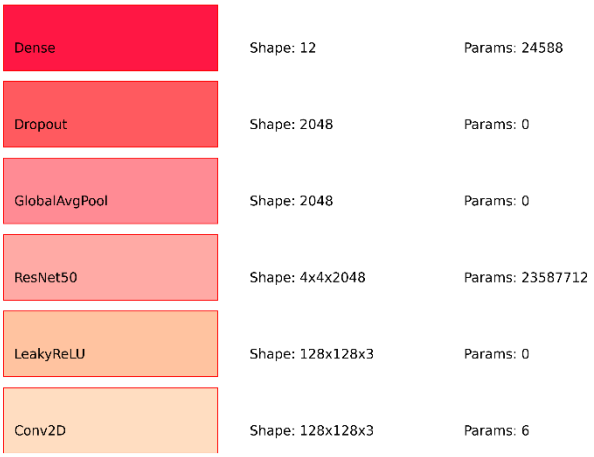

Figure 1 Deep Neural Network Model

2. Proposal

In this section, the design is segmented into two parts: data processing and model training.

2.1 Data processing

The data processing primarily involves 2 steps to create the training dataset:

Data Collection: As shown in Figure 2, we gathered 4000 images from the same individual, which were divided into training, validation, and test sets for model cross-validation. The dataset comprises various types of gestures and poses, along with images captured from multiple different perspectives.

Data Preparation: This involved meticulous examination of images and keypoint coordinates, removing incorrect or inconsistent data, and applying techniques like scaling, grayscale conversion, and normalization to enhance the images.

Figure 2 keypoint image

2.2 Model training

In addition to utilizing ResNet-50, there are many methods for keypoint detection using deep learning techniques, such as OpenPose. It involves parsing the positions of the head, arms, and legs for human pose detection. However, ResNet-50 was applied to enable real-time pose estimation at a faster speed compared to the existing OpenPose [3]. This is also one of the reasons why we chose ResNet-50.

Figure 1 depicts a deep neural network tailored for image classification. It encompasses convolutional layers, ResNet-50 feature extraction, global average pooling, dropout, and a fully connected layer. Input comprises 128x128 pixel-sized color

† Kumamoto University

human body images with 3 channels. The initial layer employs Leaky ReLU for feature extraction. A pretrained ResNet-50 model maps image features into a 2048-dimensional vector, reduced to 1-dimensional using global average pooling. Dropout prevents overfitting. The final fully connected layer maps feature to 12 output nodes, suitable for multi-class classification. The model has 23,612,306 parameters, 23,559,186 trainable, and 53,120 from ResNet-50. It's trainable for keypoint detection on the right dataset.

3. Evaluation

On a PC equipped with an AMD Ryzen 9 5900HS running at 4.6GHz and 16GB of memory, the loss and accuracy of model inference were measured. The average inference time for each inference was 43.27 milliseconds. As shown in Table 1, these are performance metrics related to collected training results.

Table 1 Training Metrics History

epoch	loss	accuracy	MAE*	LR**
1	0.10115	0.56891	0.1755	1.000e-03
2	0.00439	0.84863	0.0505	1.000e-03
3	0.00311	0.88604	0.0425	1.000e-03
.....	.....	.....	.....	.....
178	0.00021	0.97484	0.01118	3.219e-08
179	0.00021	0.97505	0.01131	3.219e-08
180	0.00021	0.97527	0.01125	3.219e-08

*MAE: Mean Absolute Error
**LR: Learning Rate

3.1 Loss

In our data, the loss values gradually decrease from the first training epoch to the 180th epoch, indicating continuous improvement in the model's performance on the training data. Figure 3 illustrates this reduction process. While the initial value is relatively high, it gradually converges to a lower level as training progresses.

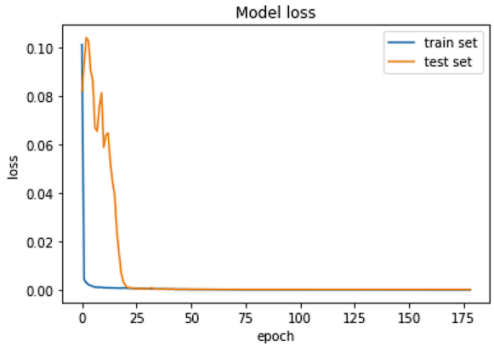

Figure 3 Model loss

3.2 Accuracy

Model accuracy steadily increases from approximately 0.57 in the first epoch to around 0.975 in the 180th epoch, as depicted in Figure 4. This demonstrates the model's continuous enhancement in performance for classification tasks.

MAE is the most natural measure of average error magnitude, all dimensioned evaluations and inter-comparisons of average model performance error should be based on MAE [4]. So, our main focus is to observe the changes in MAE.

MAE values progressively decrease from 0.1755 to 0.01125, signifying the model's gradual improvement in regression performance and its enhanced ability to fit the target data.

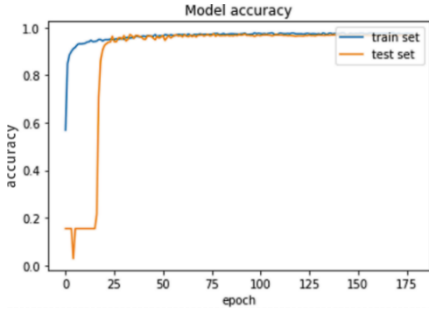

Figure 4 Model accuracy

4. Conclusion

By utilizing the ResNet-50 neural network as the foundational framework, we have effectively transformed it into a keypoint detection tool. This solution addresses the issue of inaccurate coordinate data caused by hand drifting, which is a common problem associated with sensors. It can be deployed on embedded devices, providing precise and stable control signals for the field of robotic arm control.

Looking ahead, we anticipate expanding this research to accommodate a broader range of robotic application scenarios while continually enhancing the system's performance and stability.

Reference

[1] António Amorim, Diana Guimarães, Tiago Mendona, Pedro Neto, Paulo Costa, and António Paulo Moreira, "Robust human position estimation in cooperative robotic cells," Robotics and Computer-Integrated Manufacturing, vol. 67, 2021, pp. 102035, ISSN 0736-5845, doi: 10.1016/j.rcim.2020.102035.

[2] Y. Wang, Y. Wu, S. Jung, S. Hoermann, S. Yao and R. W. Lindeman, "Enlarging the Usable Hand Tracking Area by Using Multiple Leap Motion Controllers in VR," in IEEE Sensors Journal, vol. 21, no. 16, pp. 17947-17961, 15 Aug.15, 2021, doi: 10.1109/JSEN.2021.3082988.

[3] J. Lee, T.-y. Kim, S. Beak, Y. Moon, and J. Jeong, "Real-Time Pose Estimation Based on ResNet-50 for Rapid Safety Prevention and Accident Detection for Field Workers," in Electronics, vol. 12, no. 16, article 3513, 19 Aug. 2023, Department of Smart Factory Convergence, Sungkyunkwan University, 2066 Seobu-ro, Jangan-gu, Suwon 16419, Republic of Korea, doi: https://doi.org/10.3390/electronics12163513.

[4] L. Vescovi, M. Rebetez, and F. Rong, “Assessing public health risk due to extremely high-temperature events: climate and social parameters,” Climate Research (CR), vol. 30, pp. 71-78, 2005. doi:10.3354/cr030071.

Acknowledgments This research was supported by Japan Science and Technology Agency (JST), CREST, JPMJCR19K1. We would like to express our gratitude to all of them.