Research and application of real-time human posture recognition technology based on TensorFlow.js and CNN model

Xiya Yu, Yao Tan*, Shuxian Gao, Yuhan Zhang, Haiying Fang
Tongda College of Nanjing University of Posts and Telecommunications,
Yangzhou, Jiangsu, China

ABSTRACT

With the continuous progress of society and the rapid development of technology, emerging technologies such as artificial intelligence are becoming increasingly prevalent in daily life. In particular, in the field of human pose recognition, traditional solutions have mostly relied on high-performance servers or specialized hardware, which have limitations such as high cost, poor real-time performance, and low convenience. Therefore, developing a low-cost, high-performance human pose recognition system is of great significance for promoting technological progress and meeting people's needs for intelligent living. TensorFlow.js is an open-source machine learning library that makes it possible for AI models to run in Web browser environments. On the basis of TensorFlow.js technology, combined with the front-end Vue.js framework, an online real-time human pose recognition system was developed on the Web browser end. By calling the PoseNet model and utilizing the CNN model to optimize the overall learning performance, the COCO key points are defined and recognized, and the pose recognition model is deployed on the Web browser end, lowering the system's usage threshold and improving user access efficiency. At the same time, it ensures a relatively high recognition accuracy, reducing dependence on server resources, and achieving lightweight, low-cost real-time pose recognition detection and analysis functions.

Keywords: TensorFlow.js, COCO keypoints, CNN model, vue.js framework, posenet model,web browser side.

1. INTRODUCTION

With the continuous advancement of technology and the rapid development of artificial intelligence, we are currently in an era of high-speed development of artificial intelligence. Human pose recognition, an emerging computer technology that is becoming increasingly popular and important, has been widely applied in many fields such as fitness guidance, sports safety, and virtual reality. This technology helps computer systems understand complex human actions and behaviors, and provides more personalized and precise services in a continuously refined direction. However, traditional human pose recognition systems are mostly dependent on high-performance servers and professional hardware equipment for complex program calculation and processing. This not only increases the deployment cost of the system, but also limits its application scope on mobile devices and other equipment. With the emergence of the TensorFlow.js framework, it is possible to run complex machine learning models in Web browsers, providing a new solution for real-time human pose recognition. Compared with traditional human pose recognition technology, this technology does not require complex hardware support, does not require powerful computing devices or complex model training, and users only need to access Web browsers through the network. Based on this, the technology combines the Vue.js framework in front-end technology to achieve component-based modular development, making the technology successfully landed and applied, greatly expanding the application scenarios and user groups of human pose recognition technology. In addition, the implementation of the technology enhances the advantages of privacy and real-time data processing. By directly processing video data on the front end, all related data are stored locally on the computer, avoiding remote data transmission and better protecting users' personal privacy.

2. CORE TECHNOLOGY

2.1 TensorFlow.js

TensorFlow.js is an open-source JavaScript library designed for developing and executing machine learning models, supporting work in Web browsers and Node.js environments. As part of the TensorFlow ecosystem, TensorFlow.js

* yuxy@njupt.edu.cn

International Conference on Optics, Electronics, and Communication Engineering (OECE 2024), edited by Yang Yue, Proc. of SPIE Vol. 13395, 1339527 · © 2024 SPIE · 0277-786X · Published under a Creative Commons Attribution CC-BY 3.0 License · doi: 10.1117/12.3049898

inherits the powerful features and wide range of use cases of the TensorFlow framework, while also boasting flexibility and ease of use, making artificial intelligence-related technologies more accessible to front-end developers. One of the notable advantages of TensorFlow.js is that it can run machine learning models directly on the client side without sending data to the server for processing[1], reducing latency and improving response speed while greatly protecting users' privacy. Furthermore, by eliminating the negative impact of the limited running environment, developers can easily integrate complex machine learning models into existing Web applications, providing users with rich and excellent interactive experiences. In the front-end field, the superiority of TensorFlow.js lies in its cross-platform compatibility and high performance. It not only supports various mainstream browsers, but also utilizes WebGL to accelerate hardware, ensuring good performance in computationally intensive tasks. With TensorFlow.js, it is easy to implement advanced features such as image recognition, language processing, and human pose recognition, without a deep understanding of the underlying machine learning principles. Additionally, TensorFlow.js provides a rich API and pre-trained models, which allows users or developers to quickly get started even without a machine learning background. All they need is a pre-trained dataset, and they can train and deploy the model[2]. At the same time, TensorFlow.js has an active community and abundant learning resources, providing excellent community support for developers.

2.2 Vue.js Framework

Vue.js is a progressive JavaScript framework that has gained popularity in the front-end development community since its launch. It has become a popular choice due to its lightweight, easy-to-learn, and highly efficient features. The core advantage of Vue.js lies in its data-driven view component system, which enables developers to easily build dynamic interfaces and rich user interaction experiences with a simple API[3]. Additionally, Vue.js supports virtual DOM, ensuring high performance for applications, and its modular and component-based design philosophy greatly improves code reuse and maintainability.

In particular, during the research process for this project, Vue.js was used for component-based development based on the MVVM(Model-View-ViewModel) architectural pattern. Its operating mode diagram can be seen in Figure 1. This pattern promotes the separation of the view(UI) and business logic(Model). In MVVM, the ViewModel is the core of Vue.js, which is an instance of Vue. ViewModel binds View and Model through bidirectional data binding[4]. This means that when the data model changes, the view will automatically update; conversely, when the user operates on the view, the data model will change accordingly. This automatic synchronization simplifies the complexity of manually manipulating the DOM and data state synchronization in traditional front-end development.

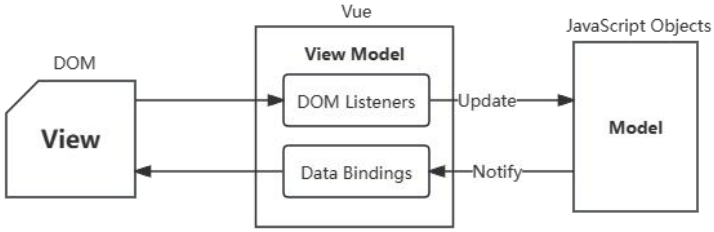

Figure 1. MVVM Diagram.

Meanwhile, the component-based development of Vue.js is another manifestation of its advanced features. Component-based development allows developers to break down complex application interfaces into numerous small, independent, and reusable components. Each component contains its own template, logic, and styles, not only making the code clearer and more organized but also improving development efficiency and application maintainability. Vue.js's component system supports parent-child component communication, component nesting, and advanced features such as slots and dynamic components, which provide strong support for building dynamic and interactive SPA.

2.3 PoseNet Model

PoseNet is a deep learning-based visual recognition model specifically designed to estimate the pose of people in images or videos and works on the COCO human keypoint dataset. As part of Google's TensorFlow.js, PoseNet can run in real-time in the browser and recognize the key points of the human body, including the eyes, ears, nose, shoulders, elbows, wrists, knees, and ankles. The capabilities of PoseNet make it have a wide range of application prospects in fields such as motion analysis, health monitoring, and interactive design. One of the advantages of using the PoseNet model is its low requirement for computing resources, unlike traditional pose recognition models that require powerful GPU support,

PoseNet can run directly on the client browser, greatly reducing data transmission time and ensuring the privacy of user data[5]. This lightweight feature means that PoseNet is very suitable for real-time pose recognition on mobile devices and Web platforms, providing developers with unprecedented convenience and flexibility.

Combining TensorFlow.js, the implementation of PoseNet becomes simpler and more efficient. Developers can easily integrate and deploy the PoseNet model using the APIs provided by TensorFlow.js, without needing to delve into the underlying machine learning knowledge or handle complex environment configurations. For single-person human pose recognition, it is implemented by inputting the image, ratio factor, horizontally flipped image, and output frequency. For multi-person models, additional parameters such as the maximum number of pose recognition, pose confidence score, and NMS radius need to be added[6]. This model's ability to run in a Web browser greatly accelerates the development cycle and significantly improves the accessibility and user experience of the application. Additionally, it allows developers to implement complex pose recognition functions with minimal effort, providing users with real-time, secure, and privacy-protected pose recognition services.

3. RESEARCH METHODS

With the help of user-owned camera hardware devices such as smartphones and laptops, the user's overall body state will be automatically identified from real-time and recorded videos. The operation flow of the system is shown in Figure 2.

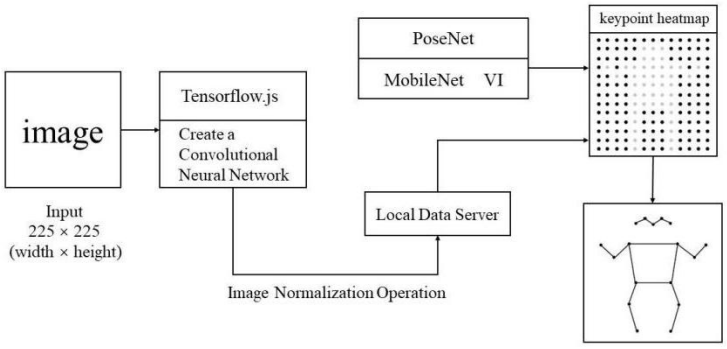

Figure 2. System Flowchart.

3.1 Data input

During the operation of the system, the data input stage is a critical part of the entire process. This stage mainly involves the capture process of real-time video streams and image data through camera devices. The quality and real-time nature of the data input directly affect the effect and performance of subsequent processing. To optimize data input, it is necessary to consider adjusting video resolution and controlling frame rate to improve system efficiency, and pre-processing data preparation, including normalization, size adjustment, and color space conversion. These measures help ensure that the data meets the input requirements of the model and provide strong support for subsequent data processing and pose model estimation. The careful design and preparation of the data input stage lay a solid foundation for the smooth operation of the whole human pose recognition system.

3.2 Data preprocessing

The effect of data preprocessing directly affects the performance and accuracy of the recognition system. After the video enters the code program as input, the code will perform a series of image processing operations on the video frames to ensure that they can be effectively interpreted by the model. The recognition process includes several key steps, such as image scaling, normalization, and color space conversion, with the goal of adjusting the input video frames to the expected format and size, while minimizing the impact of external factors such as lighting and shadows.

After the image is fully processed, the code will call the PoseNet model for the next step of data processing. PoseNet is a well-known deep learning model specifically designed for real-time human pose recognition. It can identify multiple key points in an image. The core advantage of calling this model is its lightweight level design and high degree of stability, which allows it to achieve fast predictions without sacrificing accuracy, making it very suitable for various real-time processing scenarios.

After the preprocessed image data is transmitted to the model through the forward propagation algorithm, the forward propagation is a basic process in deep learning. It sends the processed video frames into the local network and propagates them through the network's layers until the final output is obtained. In the context of human pose recognition, the output of the forward propagation is a prediction of the positions of key points in the image. The position information of these key points will be decoded and used for further applications, such as action analysis, fitness guidance, or interactive entertainment.

The direct goal of the forward propagation is to transmit the results of the recognition and analysis process to the local data server. This step aims to save the processing results for future use and provide support for data analysis and feedback. The server performs additional processing while storing the data, such as data aggregation and long-term trend analysis. At the same time, to ensure the security and privacy of the data, it will only be stored on the local server.

3.3 Data analysis

Data analysis is the core of human pose recognition process, which determines whether the system can accurately and efficiently recognize human poses. The main task of this stage is to use preprocessed data, through convolutional neural networks(CNN), and the Adam learning method[7], to optimize the overall learning performance and complete the location and recognition of human key points and poses. This process is carried out on the local data server to ensure processing speed and data security.

The system retrieves preprocessed data sets from the local data server, which have undergone preprocessing steps such as cleaning and standardization to effectively remove noise and irrelevant information and retain key features for pose recognition. The system builds a convolutional neural network by constructing convolutional layers to extract image features, which can be described by the following formula, where I represents the input image and K represents the convolution kernel. This method enhances the spatial understanding of the image by sliding the convolution kernel over the image to capture local features, greatly increasing the image's spatial understanding ability.

$$f_{ij} = \sum_m \sum_n I_i + m, j + n \cdot K_{mn} \tag{1}$$

The network adds a SoftMax function to classify keypoints and achieve accurate localization. COCO(Common Objects in Context) standard is used for joint point localization. This is a widely used standard for human keypoint recognition, designed for object detection, human keypoint detection, and semantic segmentation[8], which includes the main joint points of the human body, such as the head, shoulders, elbows, wrists, hips, knees, and ankles. The system builds a skeleton model of the human body by identifying and locating these keypoints, laying the foundation for subsequent pose recognition.

Finally, the system will perform the human pose recognition part. During this process, the system needs to identify the key points of the human body and understand the relationships and pose types between these points. To do this, the system uses a long short-term memory network(LSTM) to process sequential data and design pose discriminators, which models can capture long-distance dependencies in time series and help understand the changes in pose during continuous actions.The overall operating principle of the above model can be seen in Figure 3.

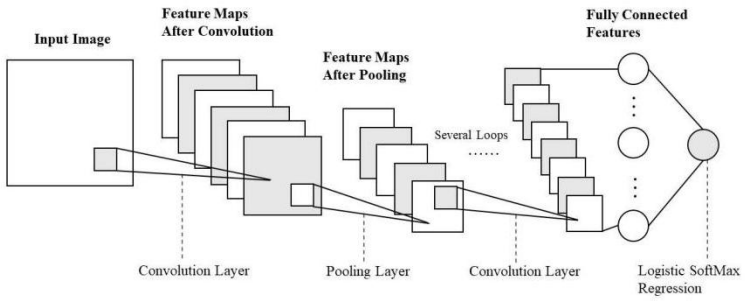

Figure 3. CNN Network Diagram.

3.4 Data output

The processing stage from the output of data analysis to the front-end page is a key part of the system's user interaction experience. This stage is mainly implemented by calling the Canvas element in HTML5. The real-time drawing of the recognition results is achieved through this method. Canvas provides a method of drawing graphics using JavaScript and HTML canvas. In this process, the system first converts the data processed by TensorFlow.js into graphical information, and then dynamically draws it on the front-end page using the Canvas API and refreshes the canvas in real-time. The specific output style diagram of recognizing human postures is shown in Figure 4.

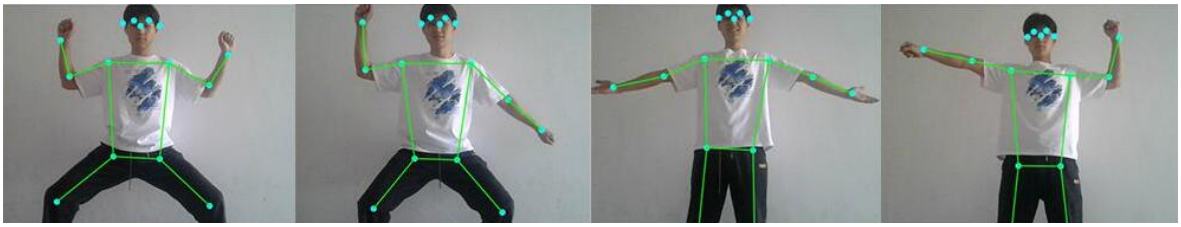

Figure 4. An Online-Output Style Diagram for Recognizable Human Postures.

Furthermore, a generalized Canvas protocol was introduced, which proposed a formal model to enhance the security and usability of WSN protocols[9]. The process diagram can be seen in Figure 5. This not only enables accurate display of human posture, but also enhances the user's interactive experience and understanding through graphical means, while ensuring data security.

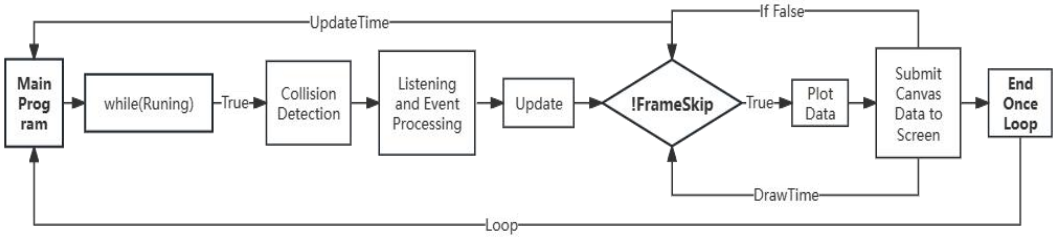

Figure 5. The Running Process of Canvas in Threads.

The entire front-end of the system, including the TensorFlow.js module, was built using Vue.js, further improving the development efficiency and user experience. By using the component-based development of Vue.js approach, the page was divided into independent, reusable components, with each component responsible for a portion of the functionality. This development model is not only more structured and easier to maintain than traditional front-end development, but also significantly improves development efficiency and project scalability. In particular, in scenarios where real-time drawing of human pose recognition results requires high performance, the efficient data of Vue.js binding and virtual DOM technology can effectively reduce the burden of page rendering, ensuring a smooth user experience.

4. SYSTEM OPTIMIZATION

4.1 Introduce WebGL as the Backend for the TensorFlow.js Module

In modern web applications, such systems based on TensorFlow.js can analyze users' video streams in real-time, identify the positions of various parts of the human body, and provide strong support for remote fitness guidance, virtual try-on, etc. However, human pose recognition is a computationally intensive task, and executing complex deep learning models directly in the JavaScript environment may lead to performance bottlenecks. To solve this problem and considering that PoseNet is a medium-sized edge model, the project chose to use WebGL as the backend.

WebGL is an API that allows for hardware-accelerated graphics in web browsers without the need for plugins[10]. When TensorFlow.js utilizes WebGL as the backend, it can perform parallel computing through the GPU, greatly improving data processing speed. This is because GPUs are specifically designed for processing complex graphics and images, and can perform tens of thousands of computing tasks simultaneously, while traditional CPUs are more focused on sequential calculations. By assigning computing tasks to the GPU, this system can achieve smoother human pose recognition. At

the same time, it can maintain high accuracy and response speed even in complex scenarios or when processing high-definition video streams.

Additionally, while there are newer WSAM options available as a new backend choice, after considering a series of factors including browser compatibility, WebGL has been found to outperform WSAM in terms of computational speed for medium-sized models such as PoseNet after repeated testing. The inference times of WebGL and WSAM for different models are shown in Figure 6. With the development of Web technology, almost all modern browsers natively support WebGL, which means that online human pose recognition systems can run seamlessly on various devices without the need for additional software or plugins. Users can access the service through a web browser, greatly improving the system's usability and adoption rate.

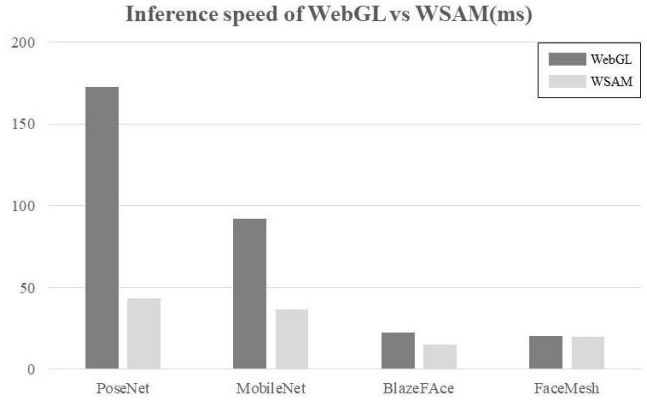

Figure 6. Inference Time Comparison of WebGL and WSAM across Different Models.

4.2 Optimize image classifiers for fixed scenes

In the development process of optimizing image classifiers, a crucial step is to adopt a binary cross-entropy loss function specifically tailored for a particular scenario, continuously optimizing the binary states of "human presence" in the discriminative scene. Combining this calculation method with the optimization method, the model can learn and distinguish between these two situations more accurately, providing more professional and reliable data analysis and transformation for accurate human detection and scene analysis.

Specifically, in the model training stage, it is necessary to cover as much diversity in the scene as possible, such as different lighting conditions, changes in human posture, and diversity in scene backgrounds. These factors may all affect the recognition effect, therefore, a high-quality and diverse dataset is a key factor in optimizing the model's performance[11]. For each image, the "human presence" or "absence" label is marked on the image based on the image, and the binary cross-entropy loss function is used to guide the model training. This function precisely adjusts the model's weights by comparing the model's predicted probability with the actual label, ensuring that the model can accurately identify the scene state in actual applications. In practice, the model is trained iteratively multiple times, with each iteration trying to minimize the following loss function:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (2)$$

Where N is the total number of training samples, y_i is the actual label of the i th sample, the scene with a human is marked as 1, the scene with no human is marked as 0, and p_i is the probability of the model predicting the i th sample as a human. In the online system we implemented, through the calculation and analysis of this function, when the model's prediction p_i is close to the actual label y_i, the loss value is small. When the predicted value differs greatly from the actual label, the loss value is higher. Therefore, by continuously optimizing the specific value of the loss function, the model can gradually learn more accurate predictions during the training process. The training diagram and reference standards can be seen in Figure 7 below.

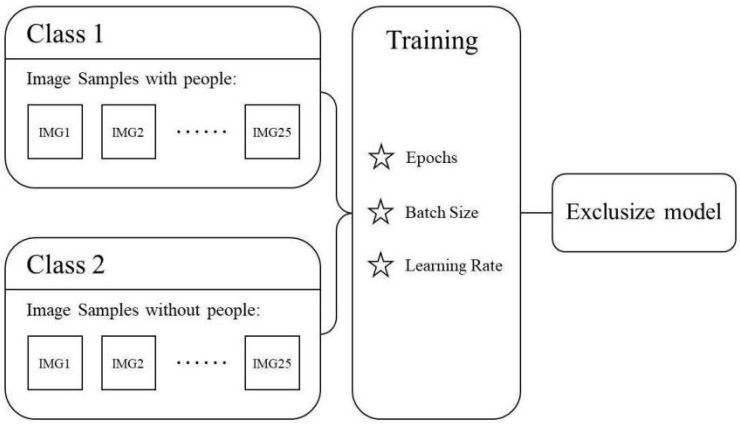

Figure 7. Optimizing Image Classifiers.

This process not only helps the model learn how to extract useful features from images, but also ensures that the model can maintain a high recognition accuracy rate when facing new and unseen scenes. Through this meticulous training method, the optimized system has demonstrated higher accuracy and reliability in identifying whether there is a person in a scene, significantly improving the system's practicality.

4.3 Docker container deployment

Compared to traditional virtual machine deployment, Docker is lighter weight, whereas virtual machines seem too bulky in comparison. In terms of project deployment capabilities in the current stage, Docker is also more convenient. The comparison between virtual machines and Docker containerization features is shown in Table 1:

Table 1. The Differences Between Virtual Machines and Docker Containers.

Characteristic	Container	Virtual machine
Isolation Level	Process	Operating system
Isolation Strategy	CGroups	Hypervisor
Resource Occupation (MAX)	5%	15%
Startup Duration	Seconds	Minutes
Image Storage	KB-MB	GB-TB

Docker containerization technology allows you to package an application and all its dependencies into a self-contained, portable container. For the human pose recognition system, this means that the TensorFlow.js model, related libraries, and other dependencies can be packaged into a single container without worrying about environment configuration issues. This means that the environment will be consistent regardless of where the system is deployed, whether it's a development environment, a testing environment, or a production environment, and it avoids the problems caused by inconsistent environments. Additionally, Docker containerization deployment has the feature of fast deployment and scaling. With the help of Docker images, the human pose recognition system can be quickly deployed on different servers without the need for manual and tedious environment configuration. Furthermore, since containers are lightweight, they can be started and stopped in a short time, thus improving the system's flexibility and scalability.

In terms of security, each Docker container is isolated from each other, allowing multiple containers to run on the same server without interfering with each other. This isolation not only helps to avoid potential conflicts but also improves the system's security by preventing malicious users from attacking one container to affect the running of other containers[12]. This provides convenience for the smooth operation and maintenance of the system.

5. CONCLUSION

The research and implementation of a real-time online human pose recognition system based on TensorFlow.js and browser technology is an example of the combination of artificial intelligence technology and front-end technology. With the accelerated popularization of the Internet and mobile devices, the performance of browsers and mobile terminals has been continuously optimized, and the performance and compatibility of online systems will also be continuously optimized, which will further provide users with more convenient and efficient human pose recognition services and bring huge application potential to the fields of fitness, medicine, and security. Currently, online real-time systems based on browsers have the advantages of cross-platform and convenience, and can provide users with more personalized and intelligent services. In the future, as users' needs continue to increase and technology continues to improve, the system will become an indispensable smart assistant in people's daily lives and contribute to the development and progress of society. At the same time, as machine learning technology continues to develop and optimize, the system will further improve the accuracy and real-time performance of recognition, and by continuously optimizing algorithms and improving model performance, the system is expected to achieve popularization and application in more fields, bringing more convenience and innovation to people's lives.

ACKNOWLEDGEMENT

Fund Project: Jiangsu Provincial Undergraduate Innovation and Entrepreneurship Training Program, Project Name: Design and Implementation of a Human Pose Recognition System Based on TensorFlow.js,Project No.: 202313989040Y Establishment Time: 2023.5

REFERENCES

[1] Smilkov, D., Thorat, N., Assogba, Y., “Tensorflow.js: Machine learning for the web and beyond,” Proceedings of Machine Learning and Systems, 1, 309-321(2019).
[2] Arnesia, P, D., Pratama, N, A., Sjafrina, F., “Aplikasi artificial intelligence untuk mendeteksi objek berbasis web menggunakan library tensorflow js, react js dan coco dataset,” JSiI (Jurnal Sistem Informasi), 9(1), 62-69 (2022).
[3] Zhu, E, H., “Research on web front-end application based on vue.js,” Science and Technology Innovation, 20, 119-121(2017).
[4] Chen, L., Lu, D., Lu, D., “Data process visualization design based on SVG and vue,” Computer Systems Applications, 31(4), 130-136 (2022).
[5] Jo, B, J., Kim, S, K., “Comparative analysis of openpose, posenet, and movenet models for pose estimation in mobile devices,” Traitement du Signal: signal image parole, 1, 39 (2022).
[6] Chen, Y., Shen, C., Wei, X.S, and Yang, j.“Adversarial posenet: a structure-aware convolutional network for human pose estimation,” IEEE Computer Society, 10, 1212-1221 (2017).
[7] Zhou, Y, K., Wang, Y., Zhao, Y, F. and Yuan, Yan., “Human pose recognition based on CNN,” Computer and Modernization, 02(009), 1006-2475 (2019).
[8] Wang, Wei., Wang, C, L., Pei, Z., “Method of human pose estimation guided by heatmap connection,” Journal of Xi'an Polytechnic University, 35(5), 9 (2021).
[9] Marián, N., “Design and analysis of a generalized canvas protocol,” Die Hebamme, 27(01), 44-47(2010).
[10] Danchilla, B., [Beginning WebGL for HTML5], Apress, Berlin, 329 (2012).
[11] Li, Y., Zhang, Y.and Wang, C., “Mini-gemini: mining the potential of multi-modality vision language models,” arXiv preprint arXiv, 2403, 18814(2024).
[12] Liu, S, y., Li, Q. and Li, Bin., “Research on container isolation based on docker technology,” Software, 4, 110-113 (2015).