

# Real time object tracking using deep learning and OpenCV

ZhiHao Jin<sup>1,3,4</sup>, Huanqing Yang<sup>2</sup>

<sup>1</sup>College of Physics, Sichuan University, Chengdu, China

<sup>2</sup>School of Electrical Engineering, Xi'an Jiaotong University, Xian, China

<sup>3</sup>2020141500198@stu.scu.edu.cn

<sup>4</sup>Corresponding author

**Abstract.** This article summarizes the background and application fields of Open-Source Computer Vision Library (OpenCV) and deep learning and conducts research based on their object detection and tracking applications. The model algorithm selected in this article is the Convolutional Neural Networks (CNN) algorithm. CNN algorithm can be used for object detection in real-time scenarios. Moreover, the CNN algorithm has good credibility. The Python program developed based on CNN algorithm can effectively achieve real-time object tracking. The model shows good detection and tracking performance for trained targets and can be further applied in more specific scenarios in the future. Because deep learning can process large-scale data and recognize complex patterns, it can automatically learn and extract advanced features. Combining the two can achieve faster and more accurate detection and tracking of target objects. After having a large enough training sample size, it can detect and track the specified object more accurately. However, it is precisely due to the enormous sample size required for deep learning that there are still some difficulties in applying it to real-time object tracking. This article first discusses the use of supervised learning methods for deep understanding. When the input sample capacity is large enough, the machine can better reflect the real-time detection and tracking of the target object. But the time required will significantly increase. This issue still needs to be addressed in the subsequent research process.

**Keywords:** deep learning, OpenCV, object detection, support vector machines, Convolutional Neural Networks

## 1. Introduction

Real-time object tracking is a critical task in computer vision, which has attracted wide attention in recent years. It involves continuous monitoring and positioning of objects of interest in the scene to achieve applications such as video surveillance, autonomous navigation, and human-computer interaction [1]. Traditional object tracking methods often rely on hand-designed features and explicit models, making it difficult to handle complex scenes, occlusion, and changes in the appearance of objects.

To overcome these limitations, advances in deep learning techniques in recent years have revolutionized the field of computer vision. Deep learning, as a subfield of machine learning, has shown remarkable ability in handling large-scale data and complex pattern recognition tasks. It utilizes multi-layer neural network models inspired by the structure and function of the human brain to learn and extract advanced features from raw data automatically.

Deep learning models, especially CNN, excel at object detection and classification tasks. By training these models on large-scale data sets, they learn to recognize and understand objects' inherent features and representations [2]. This ability to automatically learn discriminant features makes deep learning an ideal method for real-time object tracking.

At the same time, OpenCV became a widely adopted open-source framework for computer vision tasks. OpenCV offers a comprehensive set of image processing and computer vision algorithms that researchers and developers favor. Its modular architecture, cross-platform compatibility, and extensive documentation make it ideal for academic research and industrial applications.

OpenCV offers many capabilities, including image acquisition, preprocessing, feature extraction, and visual analysis. It has efficient real-time processing ability and can be used to develop a real-time object-tracking system. In addition, OpenCV seamlessly integrates with deep learning frameworks such as TensorFlow and PyTorch, enabling researchers to fully leverage the power of deep learning within the OpenCV ecosystem [3].

Combining deep learning and OpenCV has great potential to advance the field of real-time object tracking. Using deep learning models trained on large-scale data sets, object tracking systems can learn to generalize and adapt to various object appearances, changes, and complex environments. OpenCV is supporting in this regard, providing a robust real-time video processing framework and implementing feature extraction and tracking algorithms [4].

This article will delve into deep learning and OpenCV and the integration of the two in real-time object tracking. It will explore methods and techniques for training deep learning models for object tracking, integrating deep learning frameworks with OpenCV, and implementing efficient tracking algorithms. In addition, it will discuss the advantages, challenges, and potential applications of this combined approach, demonstrating the potential for more accurate, robust, and efficient real-time object-tracking systems.

## **2. The Deep Learning And OpenCV**

### *2.1. Background on Deep Learning*

Deep learning is a machine learning method that aims to achieve pattern recognition and predictive analysis by constructing and training neural networks. Compared to traditional machine learning methods, deep learning utilizes multi-layer neural network models that can automatically learn and extract high-level features from data. Neural networks, which simulate the connections between neurons in the human brain, are the fundamental building blocks of deep learning. They consist of input, hidden, and output layers, processing data through layers of nodes and connections. Gradient descent is an optimization method used in deep learning to minimize the loss function. In deep learning, the model gradually approaches the optimal solution by calculating the gradient of the loss function concerning the model parameters and updating the parameters in the opposite direction of the rise. Backpropagation is an optimization algorithm in deep learning used to adjust the weights and biases in neural networks, aiming to make the model's predictions closer to the actual values. It is based on the idea of gradient descent and propagates the gradients of prediction errors backward through each layer of the network, updating the network parameters layer by layer.

Additionally, pooling is often utilized to prevent overfitting in deep learning models. Pooling is a dimensionality reduction technique that reduces the spatial size of the input data, thus reducing the number of parameters and computational complexity in the network. In other words, pooling helps retain crucial information and minimize redundancy, leading to more efficient and robust models. In summary, deep learning leverages multi-layer neural network models, optimization methods such as gradient descent and backpropagation, and pooling techniques to automatically learn and extract high-level features from data, enabling more accurate and efficient pattern recognition and predictive analysis [5]. Deep learning has achieved significant breakthroughs in various fields, especially in computer vision. Its combination with OpenCV further enhances the capabilities and effectiveness of computer vision applications.

In recent years, deep learning architectures have succeeded in being implemented in numerous practical applications in many research areas, especially in computer vision, like face recognition and image classification, because deep learning architectures can learn more invariant and inherent features via the multi-layer learning described before. Deep understanding yields high-level image features that are more accurate and stable to represent the images' essence. Even in the case of complex videos, this approach provides better parts of the moving object to prevent drifting away from the target. For all these reasons, deep learning is used in our tracking system for learning an object representation that is robust against variations [6].

## *2.2. Background on OpenCV*

OpenCV is an open-source computer vision library that is free to use in commercial and academic projects. It offers various functionalities and modules for image processing, analysis, and computer vision tasks. OpenCV is known for its high performance, customizability, and scalability, making it suitable for various computer vision applications. It has been widely applied in robotics, autonomous driving, medical image processing, surveillance, and face recognition. OpenCV provides rich basic functionalities, including image reading, displaying, saving, and processing. It supports various image formats and offers image filtering, transformations, geometric operations, and more. OpenCV also provides camera interfaces for real-time acquisition and processing of camera data [7]. Its cross-platform nature and multi-language support make it an ideal tool for researching and applying computer vision algorithms. OpenCV plays a crucial role in academic research and industrial development, continuously contributing to the advancement of the computer vision field.

## *2.3. Advantages of Combining Deep Learning and OpenCV in Computer Vision*

Deep learning excels at handling large-scale data and complex pattern recognition tasks, enabling automatic learning and extracting high-level features. On the other hand, OpenCV provides a rich set of image processing and computer vision algorithms, offering efficient real-time processing capabilities. Combining deep learning with OpenCV leads to enhanced image processing and analysis capabilities. The combination of deep understanding and OpenCV leverages the strengths of both approaches, resulting in more accurate and faster image analysis and processing [8].

Combining deep learning and OpenCV holds significant importance and advantages in computer vision. They complement each other and, when used together, can achieve more powerful image processing and analysis capabilities. Deep learning excels in handling large-scale data and complex pattern recognition tasks, enabling automatic learning and extracting high-level features. On the other hand, OpenCV provides a rich set of image processing and computer vision algorithms, offering efficient real-time processing capabilities [9]. Combining deep learning and OpenCV can leverage the strengths of both approaches to achieve more accurate and faster image analysis and processing.

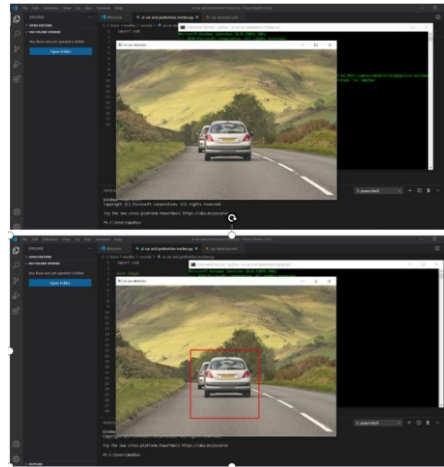
Furthermore, deep learning is widely applied within OpenCV for various computer vision tasks. For example, pre-trained deep learning models can be used for image classification by loading pre-trained neural network models and accurately classifying images. Deep learning can also be used for object detection by training deep learning models with object recognition capabilities to locate and identify objects in images. OpenCV provides strong support for deep learning. It integrates popular deep learning frameworks such as TensorFlow and PyTorch, allowing developers to conveniently use these frameworks for deep learning model training and inference.

Moreover, OpenCV provides deep learning modules, including functions for image processing and feature extraction, enabling the integration of deep learning models with other OpenCV functionalities for more comprehensive image processing and analysis [10]. By combining deep learning and OpenCV, researchers and developers can harness the powerful capabilities of deep understanding to solve complex computer vision tasks while benefiting from the rich functionalities and efficient performance of OpenCV for real-time image processing and analysis. This combination accelerates computer vision research progress and provides powerful tools and methods for image processing and analysis in practical applications.

### 3. Apply OpenCV and deep learning networks to object detection and tracking systems

#### 3.1. Application of OpenCV in Object Detection and Tracking

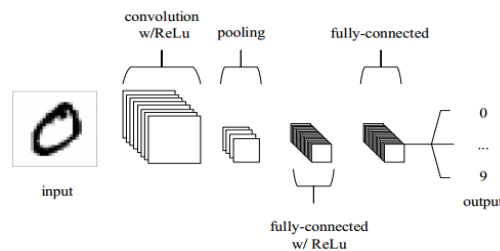
Let the machine understand the object to be detected through Supervised learning. it requires recording the actual coordinates of the moving object and pointing it out in the video.



**Figure 1.** Object Recognition [11]

Then, this article wants our model to detect the operating entity, as shown in Figure 1. The unit in motion (i.e., car) will be recognized and then encircled as a rectangle [11].

A convolutional neural network comprises an information layer and an output layer, just like other hidden layers. The hidden layer of CNN is usually composed of a series of convolution layers. These convolution layers have increasing or other point products. The Convolutional neural network contains several layers: 1. convolution layer, 2. nonlinear layer, 3. pooling layer, 4. complete connection layer. Fig. 2 shows the basic structure of CNN.



**Figure 2.** A simple CNN architecture [12]

By writing Python programs based on CNN architecture algorithms and applying them in OpenCV to achieve the goal of tracking objects. The efficiency of such a model can be observed by using the developed Python program for the real-time detection and tracking process of things and by combining it with deep learning; because deep understanding can process large-scale data and recognize complex patterns, it can automatically learn and extract advanced features, and the combination of the two can achieve faster and more accurate detection and tracking of target objects.

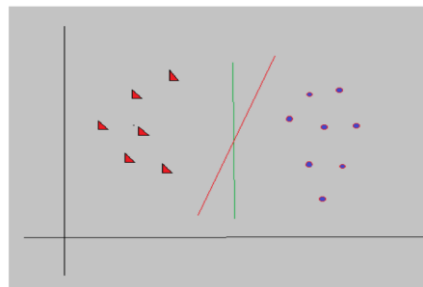
#### 3.2. Overview of the Application of Deep Learning Methods in Tracking

As a powerful feature learning method, deep learning has made significant achievements in many fields. However, there are still substantial limitations to deep learning in object tracking, which is much less than video recognition and object detection. This is mainly because 1) only the first data frame is annotated during video object tracking. In the subsequent online tracking process, the magnitude of

positive and negative samples is minimal. Still, deep learning requires a large amount of data for learning to achieve the purpose of subsequent analysis. 2) The network architecture of deep learning is relatively large, but target tracking requires higher real-time performance. The significant architecture of deep learning makes achieving the high real-time performance required in target tracking difficult. Therefore, after considering the network size and running speed, the application of deep learning is significantly discounted.

However, the vast deep learning network makes it difficult to apply in object tracking. However, the critical role of deep learning in appearance learning and feature extraction can still play a vital role. Therefore, in researching the application of deep knowledge in object tracking, the following methods can be followed: 1) Since deep learning neural networks can effectively extract features from objects, the components extracted by deep learning networks can be utilized. First, training is carried out in images and videos with a large amount of data to train the deep neural network. In the subsequent online object tracking, the introduced features of the target are extracted, and the deep neural network is adjusted using the obtained online data to achieve the goal of applying deep learning to target tracking. 2) Changes can also be made to deep neural networks, such as lowering the sample size that deep neural networks can learn, resulting in a slight improvement in their speed. To achieve the goal of balancing efficiency and sample size.

Support vector machines (SVM) can be divided into two basic structures: Support Vector Machines for Regression and Nonlinear Support Vector Machines.

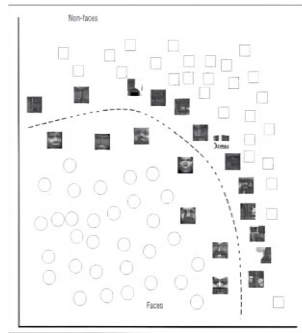


**Figure 3.** SVM classifier classifies into 2 classes [13]

How to use SVM to classify it to determine the target object and non-target object. The basic principle behind the working of Support vector machines is straightforward – Create a hyper-plane that separates the data set into classes, allowing us to start with a sample problem. Suppose that for a given data set, you've got to classify red triangles from blue circles. You aim to make a line that classifies the info into two classes, creating a distinction between red triangles and blue circles [13]. As shown in Fig. 3, this article uses SVM to detect objects. This system will scan the defined target object as much as possible on a scale for detection, dividing the most original.

### *3.3. Results of OpenCV and Deep Learning using in some area*

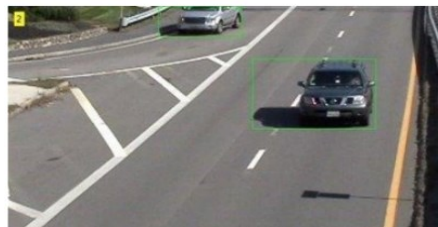
By using SVM to design a facial recognition system, such a system detects how the face is divided into overlapping sub-images by scanning the image on as many scales as possible. Support vector machines are used to classify these sub-images to determine the specific category of each image, whether it is a face or a nonhuman face.



**Figure 4.** How SVM is used in face detection [14]

The system handles multiple scales by examining windows taken from scaled versions of the original image. The primary use of SVMs is in the classification step, which is the most critical part of this work (see Figure 4) [14].

Fig. 5 shows the results of applying deep learning and OpenCV to car tracking.



**Figure 5.** Tracking of car [15]

When conducting car tracking, there are two ways to track the target: 1) Following the detection sequence, this method uses objects captured by closed circuit television at different times and in other states to follow them.) The second dynamic detection method calculates the car's position at a specific time interval based on its work at a particular moment.

#### 4. Conclusion

This article summarizes the background and application fields of OpenCV and deep learning and conducts research based on their object detection and tracking applications. The model algorithm selected in this article is the Convolutional Neural Networks algorithm. CNN algorithm can be used for object detection in real-time scenarios. Moreover, the CNN algorithm has good credibility. The Python program developed based on the CNN algorithm can achieve real-time object tracking effectively. The model shows good detection and tracking performance for trained targets and can be further applied in more specific scenarios in the future. Because deep learning can process large-scale data and recognize complex patterns, it can automatically learn and extract advanced features.

Combining the two can achieve faster and more accurate detection and tracking of target objects. After having a large enough training sample size, it can detect and track the specified object more accurately. However, it is precisely due to the enormous sample size required for deep learning that there are still some difficulties in applying it to real-time object tracking. This article first discusses the use of supervised learning methods for deep understanding. When the input sample capacity is large enough, the machine can better reflect the real-time detection and tracking of the target object. But the time required will significantly increase. This problem can be solved by using various learning methods to improve machine learning efficiency, including Supervised and unsupervised learning. However, due to the large sample size required for deep understanding, relying solely on using multiple learning methods cannot completely solve the real-time tracking problem of the target object. In the subsequent research process, in-depth research on small-sample learning models can be applied to object detection

and tracking based on deep learning, and existing problems can be improved by utilizing small-sample learning models. This is just a possible solution based on known knowledge; it still needs to be solved through continuous improvement of the system and algorithms in the subsequent research process.

## References

- [1] A. Mangawati, Mohana, M. Leesan and H. V. R. Aradhya, "Object Tracking Algorithms for Video Surveillance Applications," 2018 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2018, pp. 0667-0671, doi: 10.1109/ICCSP.2018.8524260.
- [2] A. Biswas, A. P. Jana, Mohana and S. Sai Tejas, "Classification of Objects in Video Records using Neural Network Framework," 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2018, pp. 564-569, doi: 10.1109/ICSSIT.2018.8748560.
- [3] F. K. Noble, "Comparison of OpenCV's feature detectors and feature matchers," 2016 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP), Nanjing, China, 2016, pp. 1-6, doi: 10.1109/M2VIP.2016.7827292.
- [4] B. M U, H. Raghuram and Mohana, "Real Time Object Distance and Dimension Measurement using Deep Learning and OpenCV," 2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 2023, pp. 929-932, doi: 10.1109/ICAIS56108.2023.10073888.
- [5] V. Rajesh, U. P. Naik and Mohana, "Quantum Convolutional Neural Networks (QCNN) Using Deep Learning for Computer Vision Applications," 2021 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, 2021, pp. 728-734, doi: 10.1109/RTEICT52294.2021.9574030.
- [6] Pang, S., del Coz, J.J., Yu, Z. et al, "Deep Learning and Preference Learning for Object Tracking: A Combined Approach," Neural Process Lett 47, pp.859-876, 2018, doi: 10.1007/s11063-017-9720-5.
- [7] X. Farhodov, O. -H. Kwon, K. W. Kang, S. -H. Lee and K. -R. Kwon, "Faster RCNN Detection Based OpenCV CSRT Tracker Using Drone Data," 2019 International Conference on Information Science and Communications Technologies (ICISCT), Tashkent, Uzbekistan, 2019, pp. 1-3, doi: 10.1109/ICISCT47635.2019.9012043.
- [8] V. Choudhary, P. Guha, K. Tripathi and S. Mishra, "Edge Detection of Variety of Cowpea Leaves Using OpenCV and Deep Learning," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 1312-1316, doi: 10.1109/ICAC3N56670.2022.10074348.
- [9] W. Wang, J. Wang, Z. Zhang and D. Shi, "OpenCV Implementation of Image Processing Optimization Architecture of Deep Learning Algorithm based on Big Data Processing Technology," 2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2022, pp. 65-68, doi: 10.1109/ICSCDS53736.2022.9760795.
- [10] Tushar, K. Kumar and S. Kumar, "Object Detection using OpenCV and Deep Learning," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 899-902, doi: 10.1109/ICAC3N56670.2022.10074012.
- [11] A. Sharma, J. Pathak, M. Prakash, and J. N. Singh, 'Object Detection using OpenCV and Python', in 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India: IEEE, Dec. 2021, pp. 501–505. doi: 10.1109/ICAC3N53548.2021.9725638.
- [12] K. O'Shea and R. Nash, 'An Introduction to Convolutional Neural Networks'. arXiv, Dec. 02, 2015. Accessed: Jun. 25, 2023. [Online]. Available: <http://arxiv.org/abs/1511.08458>

- [13] S. Satpute, H. Shende, V. Shukla, and B. Patil, 'Real Time Object Detection using Deep-Learning and OpenCV', vol. 07, no. 04, 2020.
- [14] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, 'Support vector machines', IEEE Intell. Syst. Their Appl., vol. 13, no. 4, pp. 18–28, Jul. 1998, doi: 10.1109/5254.708428.
- [15] G. Chandan, A. Jain, H. Jain, and Mohana, 'Real Time Object Detection and Tracking Using Deep Learning and OpenCV', in 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore: IEEE, Jul. 2018, pp. 1305–1308. doi: 10.1109/ICIRCA.2018.8597266.