

Chapter: 05

COMPUTER VISION WITH OPEN CV

Mohd Hyder Gouri*

Faculty, Glocal School of Science and Technology,
Glocal University, Saharanpur, U.P.

*Correspondence to: hyder@theglocaluniversity.in

Mr. Mohit Kumar

Faculty, Glocal School of Science and Technology,
Glocal University, Saharanpur, U.P.

DOI: <https://doi.org/10.52458/9789388996747.nsp2023.eb.ch-05>

Ch.Id:-GU/NSP/EB/EFMLDSP/2023/Ch-05

ABSTRACT

Computers can now analyze and comprehend visual data from the outside world because to the development of computer vision, a branch of artificial intelligence. It entails the creation of algorithms and methods to extract significant data from pictures and movies. The sophisticated open-source computer vision and machine learning software library known as OpenCV makes it simpler to work with visual data. OpenCV is an acronym for Open-Source Computer Vision Library. The goal of computer vision is to mimic the capacity of the human visual system to comprehend and interpret the visual environment. It involves activities like object detection, facial recognition, image segmentation, and more. It also incorporates image and video analysis. Numerous industries use computer vision, including autonomous vehicles, robotics, surveillance, and augmented reality. Frequently popular open-source library OpenCV offers tools, routines, pre-built algorithms, and tools for executing different computer vision tasks. Although it was initially created by Intel, the project is now community-driven. Python and other languages provide bindings for OpenCV, which is built in C++. Computers can now analyze and comprehend visual data from the outside world because to the development of computer vision, a branch of artificial intelligence. It entails the creation of algorithms and methods to extract

significant data from pictures and movies. The sophisticated open-source computer vision and machine learning software library known as OpenCV makes it simpler to work with visual data. OpenCV is an acronym for Open-Source Computer Vision Library. The goal of computer vision is to mimic the capacity of the human visual system to comprehend and interpret the visual environment. tasks such as image and video analysis.

Keywords: OpenCV, Open-Source Computer Vision Library, Image recognition, Object detection, Image segmentation

5.1 INTRODUCTION TO OPENCV

OpenCV, or Open-Source Computer Vision Library, is a machine learning and computer vision software library created for a variety of computer vision applications. Artificial Intelligence's field of computer vision aims to make it possible for machines to comprehend and interpret visual data from the outside world, such as pictures and movies. By offering a variety of tools, functions, and pre-built algorithms that make it simpler to work with visual data, OpenCV plays a vital role in this area.

- a. **Rich Set of Functionality:** OpenCV is a comprehensive library that offers a rich set of computer vision and image processing functions. It includes tools for image manipulation, feature detection, object tracking, facial recognition, camera calibration, and much more.
- b. **Programming Language Support:** Although Python is the most widely used programming language, OpenCV also offers user-friendly interfaces for many other languages. OpenCV is largely designed in C++. Developers may now use OpenCV's capability in their choice programming environment thanks to this.
- c. **Efficient Algorithms:** For real-time applications like robotics, augmented reality, and autonomous cars, OpenCV is renowned for its efficient and streamlined algorithms.
- d. **Community and Documentation:** For assistance with your computer vision projects, you can find a variety of online resources, forums, and tutorials thanks to the sizeable and vibrant community of OpenCV users and developers. Additionally, OpenCV offers a wealth of documentation to help developers comprehend and make use of its functionalities.

5.2 LITERATURE REVIEW

A summary of the research, uses, and developments in the area of computer vision and artificial intelligence is included in a study on the integration of OpenCV

with machine learning (ML). I'll outline some important themes and subjects you might research in this kind of literature review below: Cross-platform and open source: Because OpenCV is open-source, it is accessible to everyone and promotes community participation and cooperation. It is made to function across a number of operating systems, including Windows, Linux, macOS, Android, and iOS, making it available to a variety of developers.

5.3 INTRODUCTION TO COMPUTER VISION AND MACHINE LEARNING

Introduce the terms "computer vision" and "machine learning" first. Explain fundamental concepts and their uses.

a. Open CV and Its Capabilities

Review OpenCV's capabilities and features, paying special attention to how it affects image and video analysis.

b. Machine Learning Algorithms in Computer Vision

Investigate the application of various ML algorithms, including SVMs, decision trees, neural networks, and deep learning models, in computer vision.

- **Applications:** Numerous industries use OpenCV, including robots, imaging in medicine, the automobile industry, augmented reality, security and surveillance, and more. It is appropriate for a variety of applications due to its performance and adaptability.
- **Integration with Machine Learning:** You can use machine learning methods in your computer vision applications by integrating OpenCV with well-known machine learning frameworks like TensorFlow and PyTorch.
- **Open CV-Python:** The "opencv-python" package, which offers OpenCV Python bindings, can be used if you're working with Python. It makes it easier for Python scripts to use OpenCV functionalities.
- **Image preprocessing:** OpenCV allows you to manipulate and process images, including operations like filtering, thresholding, and resizing.
- **Feature extraction:** OpenCV enables the extraction of key points and descriptors for use in applications like image matching.
- **Object detection** It provides tools for detecting and tracking objects within images and videos.

c. Real world applications in image analysis

There are several practical uses for image analysis in many industries, including healthcare, automotive, agricultural, security, and more. Here are a few examples of image analysis in action:

1. Medical Imaging

- **Diagnostic Radiology:** For the purpose of identifying and diagnosing illnesses and injuries, X-rays, MRIs, and CT scans are examined.
- **Pathology:** To find cancer and to diagnose diseases, microscopic images of tissue samples are examined.
- **Ultrasound Imaging:** For prenatal care and condition diagnosis, ultrasound images are used.

2. Defect Detection: Product faults can be found via image analysis, ensuring quality control.

3. Object Tracking: On assembly lines, automated systems monitor the movement of the items.

4. Geospatial Analysis: Aerial and satellite photos are used in GIS (Geographic Information Systems) to map the topography, assess land usage, and design infrastructure.

5. Remote Sensing: Planning for the environment and cities is aided by images from distant sensors.

d. Unsupervised learning with K-means and DBSCAN

Planning for the environment and cities is aided by images from distant sensors. When dealing with data that have asymmetric geometries and changing cluster densities, DBSCAN is more reliable than K-means. It is less sensitive to the original positioning of data points and has the ability to automatically calculate the number of clusters. For the epsilon and minPts values, however, careful parameter adjustment may be necessary. Consider the characteristics of your data and the objectives of your clustering task when deciding between K-means and DBSCAN: When you anticipate that clusters will be broadly spherical, similar in size, and you have a good idea of the number of clusters (K), use K-means.

When you anticipate that clusters will have uneven shapes or different densities and you want the algorithm to determine how many clusters there are, use DBSCAN automatically.

e. The concept of unsupervised learning

Unsupervised learning is a machine learning technique where the model seeks out patterns or structure on its own rather than being given labeled data. For many sorts of data clustering jobs, two well-liked unsupervised clustering algorithms are K-means and DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

f. K-means

A dataset is divided into K non-overlapping clusters (subgroups) using the K-means partitioning method, and each data point is assigned to the cluster with the closest mean. This is how K-means operates.:

1. **Initialization:** K initial cluster centers should be chosen at random or using a heuristic.
2. **Assignment:** Each data point should be put into the cluster with the closest center.
3. **Update:** The cluster centers should now be calculated as the mean of all the data points included in that cluster.
4. **Repeat:** Continue assigning and updating until convergence, or when there is no longer a noticeable change in the cluster centers.

g. DBSCAN: Density -based clustering

The density-based clustering algorithm DBSCAN, on the other hand, assembles data points that are close to one another in a high-density area. As to how it operates:

1. **Initialization:** Take a random, unexplored data point as your starting point.
2. **Expansion:** A dense zone is discovered and all of the points in it are added to the same cluster if there are at least "minPts" data points within a distance of "epsilon" of the present point.
3. **Connectivity:** The algorithm continues to explore the density-connected points in the dataset.
4. **Noise:** Un-clustered data points and data points in low-density areas are classified as noise.

h. Hands on clustering examples

Let's go over some practical clustering examples utilizing Python and well-known tools like scikit-learn. For simplicity, we'll only use generated datasets. Ensure that Matplotlib and Scikit-Learn are installed. If you haven't done so already, you can install them using pip:

- **Example 1:** K-Means Clustering
- **Example 2:** DBSCAN Clustering

5.4 CONCLUSION

In this chapter, OpenCV is a powerful and versatile library for computer vision and image processing tasks. It simplifies the process of working with visual data, making it accessible to developers and researchers in various fields, from robotics to healthcare, and beyond. Whether you're a beginner or an expert in computer vision, OpenCV is an invaluable tool in your toolkit.

REFERENCES

1. *"Learning OpenCV 4 Computer Vision with Python 3"* by Joseph Howse, Prateek Joshi, and Michael Beyeler: This book provides a comprehensive guide to computer vision using OpenCV with Python and covers various applications and techniques.
2. *"OpenCV 3 Computer Vision Application Programming Cookbook"* by Robert Laganieri: This book offers practical examples and recipes for implementing computer vision applications using OpenCV.
3. *"OpenCV By Example"* by Prateek Joshi: This book provides a hands-on approach to learning OpenCV through practical examples and projects, making it suitable for those looking to apply computer vision in real-world scenarios.
4. Reddy, B. A., Srija, B. S., Sharma, R., Rohini, C., Sandhya, D., Sravani, M. P., & Vishnupur, N. *Plant Health Monitoring and Chemical Spraying Robot*.
5. Ketoja, J. A., Paunonen, S., Jetsu, P., & Pääkkönen, E. (2019). *Compression strength mechanisms of low-density fibrous materials*. *Materials*, 12(3), 384.
6. Tarbă, N., Schmidt, D., Popovici, A. E., Stăniloiu, E., Avataovului, C., & Prodan, M. (2020). *on performing skew detection and correction using multiple experts' decision*. *welcome to the journal of information systems & operations management (issn 1843-4711; idb indexation: proquest, repec, qbe, ebsco, copernicus)*. This journal is an open access journal published two times a year by the Romanian-American University., 188.
7. Kumar, A., Naughton, J., & Patel, J. M. (2015, May). *Learning generalized linear models over normalized data*. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (pp. 1969-1984).