

Implementation of Computer Vision Applications using OpenCV in C++

Adwait Bhawe, Vaishnavi Jadhav, Punit Bhandari, Vaishnavi Patil

*Department of Electronics and Telecommunication Engineering, Pune Vidyarthi Griha's College of Engineering Technology Project Guide: Prof. Anand Najan
Sponsored by NSM Solutions Pvt. Ltd. Industry guide: Mr. Nishad Mande*

Abstract — Computer Vision (CV) has emerged as a critical field in various domains, ranging from autonomous vehicles to medical imaging. OpenCV, an open-source computer vision library, provides a rich set of functions and algorithms for image processing, feature detection, object recognition, and more. This paper presents a comprehensive study of integrating OpenCV with C++, aiming to explore its potential for developing robust and efficient CV applications. We present practical examples that demonstrate the seamless integration of OpenCV with C++. We showcase real-world applications such as facial recognition, motion detection, and augmented reality, highlighting the versatility and extensibility of the OpenCV library when utilized alongside the C++ programming language

Keywords - Computer Vision, OpenCV, C++, Text Recognition, colour detection, barcode decoding, Text on Image

I. INTRODUCTION

OpenCV (Open-Source Computer Vision Library) is a widely-used open-source computer vision and machine learning software library. It provides developers with a rich set of tools and algorithms for image and video processing, object detection and recognition, and other computer vision tasks. OpenCV is written in C++, which makes it a powerful and efficient library for real-time computer vision applications.

C++ provides a powerful and flexible environment for implementing CV algorithms, leveraging features like operator overloading, templates, and memory management. We explore how C++ enhances the performance of OpenCV-based applications through its low-level control and efficient resource utilization.

We will be using various OpenCV and image processing functions and algorithms like Grey-scaling, edge-detection, morphology, contours, thresholding etc.

Computer vision tasks such as object detection and recognition benefit from image processing techniques.

Computer vision systems can recognise and categorise objects in pictures or video streams by extracting features, matching patterns, and using machine learning methods. Image processing enables applications like object tracking, surveillance, and autonomous navigation by helping to distinguish things based on colour, shape, texture, or other visual properties. Tools for picture analysis and interpretation are provided by image processing in order to get useful data from images. This involves activities like measuring, motion analysis, scene comprehension, and Object and Text Recognition, for the following given usecases

1. Colour Detection
2. Barcode Decoding
3. Text Recognition
4. Text on Image

II. LITERATURE REVIEW

The field of computer vision in C++ is active and quickly developing, with many new scientific discoveries and useful applications. In our Project we used four different use cases of computer-vision which we implemented using the programming language C++.

There are large number of studies and research in the field of computer-vision but it is all done in the programming language of python. Implementing it in C++ was a challenge in itself as not many researches were there regarding our use cases. Here is a summary of the important studies.

1. "Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library" by Adrian Kaehler and Gary Bradski - This book is a comprehensive guide to the OpenCV library and is ideal for beginners who want to learn computer vision using C++ and OpenCV.

2. "Recognizing one-dimensional barcode using OpenCV" by Junhao Liang and Tianqi Wang give a depth view in how barcode decoding works.
3. "Process of Detecting Barcodes Using Image Processing" by Kopila Pariyar explained the process of barcode detection
4. "OCR Service Platform Based on OpenCV" by Jianjun Cai, Erxin Sun, Zongjuan Chen. The study found that the OCR technique achieved high accuracy in text recognition and detection.
5. "Mastering OpenCV 4 with Python: A Practical Guide Covering Topics from Image Processing, Augmented Reality to Deep Learning with OpenCV 4 and Python 3.7" by Alberto Fernandez and Prateek Joshi - This book covers the latest version of OpenCV and provides hands-on examples of how to use OpenCV with Python to solve real-world problems.

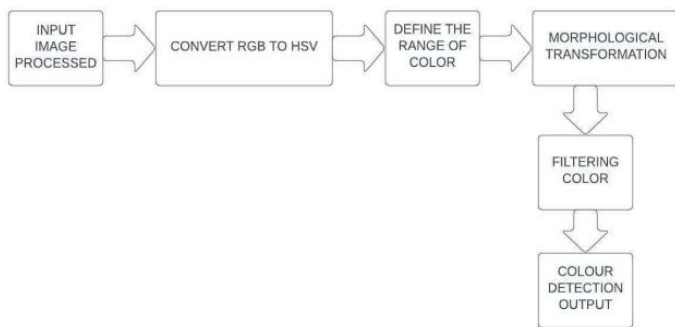
III. METHODOLOGY/EXPERIMENTAL

While working on this project we had to take a different approach while coming up with the code for each use case as there was not enough study materials related to the topic.

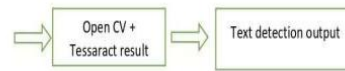
This led us to build the code and decide the method from scratch. With different permutation and combinations used as well as taking useful functions from programming language python and converting and tweaking the function/algorithm as per our need helped us to gain more insight about the expected outcome.

A. Block Diagram

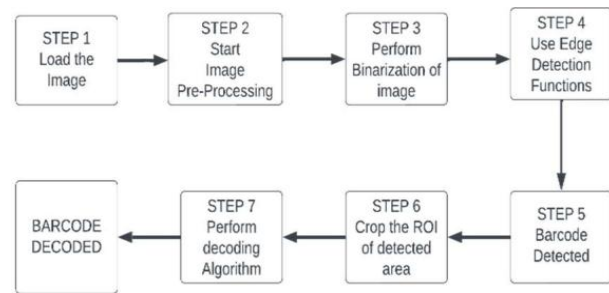
a) Colour Detection



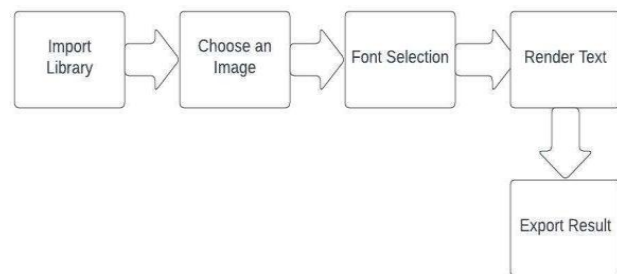
b) Text Recognition



c) Barcode Decoding



d) Text on Image



B. Synthesis/Algorithm/Design/Method

1) Colour Detection RGB Colour Model -

Red, green, and blue colours are combined in varying ratios to create a variety of colours in the RGB colour model, which is an additive colour scheme. A colour model is a method for multiplying a few core colours to produce many hues. Model light is used to display colour in additive colour. Colour mixing is additive in the RGB colour model [7]. It is the technique of combining the red, green, and blue primary colours in various ratios to create other colours. HSV colour gamut The Hue (H), the colour type (such as red, green), and the saturation (S) of the colour are the three elements that make up the HSV (Hue, Saturation, Value) model, sometimes known as HSB (Hue, Saturation, Brightness). It has a red angle of 0 to 360 degrees.

2) Text Recognition –Tesseract Library:

Text extraction from photographs or scanned documents is made possible via text recognition in OpenCV using the Tesseract package, which is a reliable and popular method. Tesseract, a Google-developed open-source OCR engine, for precise and effective text recognition.

Tesseract analyses and recognises the text within the specified regions using cutting-edge OCR algorithms. Text recognition in OpenCV utilising the Tesseract library enables the extraction of textual information from photos or scanned documents with high accuracy and efficiency by combining OpenCV's image processing skills with Tesseract's potent OCR techniques.

3) Barcode Decoding –

In order to extract information from digital images, computer vision must first modify and analyse the images. This process is known as image processing and detection. Several important methods and processes that are frequently used in image processing and detection.

- Grey scaling and the Sobel Function: Greyscale photos can be used with the Sobel function [3], a well-known edge detection technique, to locate and highlight edges based on the gradient of pixel intensity.
- Blurring and Thresholding: Blurring, commonly referred to as smoothing, is a technique for removing small features from an image and reducing noise and thresholding entails separating a picture into binary sections according to a predetermined threshold value.
- Morphology Operations: A set of operations used to examine and modify the shapes and structures found in an image are referred to as morphology operations.
- Contours: The borders or outlines of items in a picture are known as contours. They are essential in the detection, identification, and study of objects' shapes.
- ZBar Library: Zbar is an open-source software library that offers the ability to scan barcodes and QR codes.

- Region of Interest: In image processing, ROI stands for region of interest. It describes a particular area or region inside an image that is picked out for additional examination, processing, or information extraction.
- Decoding Algorithm: it is the algorithm based on Otsu Model.

4) Text on Image –

- Put Text: This function is used to write text on an image. It takes the following parameters:
- Image: The input image on which the text will be written.
- Text: The actual text string to be written.
- Position: The coordinates (x, y) of the starting point of the text.
- Font type and size: The font type and size of the text.
- Color: The color of the text.
- Thickness: The thickness of the text.
- Line type: The type of line used for drawing the text.
- Text Size: This function calculates the size of the text string.
- Scalar: This class represents a color in OpenCV. You can specify the color using the BGR (Blue, Green, Red) format.
- These are pre-defined font types available in OpenCV for writing text. FONT_HERSHEY_SIMPLEX, FONT_HERSHEY_PLAIN, FONT_HERSHEY_DUPLEX, etc.

C. Expected Outcome

1) Colour Detection:

We were able to successfully isolate the objects of which colour code we had given in the HSV tracker. Thus Object detection was successful.



4) Text on Image:

We got the desired outcome for our text on image use case



We were able to successfully Decode UPC-A barcodes using basic OpenCV Functions and algorithms in C++ without the Pre-trained library Zbar.

IV. CONCLUSION AND FUTURESCOPE

The project "Implementation of Computer Vision applications using OpenCV in C++" is an innovative project that can be seen as a demonstration project for understanding the basic concepts of computer vision, machine learning, and image processing by using different algorithms.

It also provides information on Object Detection models, Artificial Intelligence, and Computer Vision Algorithms for researchers who are really interested in the topic. This project can be used by researchers as an overview or an example project to understand how image processing functions, its significance in the field of computer vision, and the fundamental steps that must be taken into consideration. It acts as a significant demo project. Except for the phases

where the Pre-trained libraries such as Tesseract and ZBar
[] were withdrawn, the fundamentals of image processing

and computer vision are easy to understand. This led us to employing different method for those use-cases.

We were able to complete our project's primary objective—the implementation of all computer vision apps in C++, despite a few small difficulties. Testing a variety of other C++ and Python programming language may help this project advance. Applications for computer vision in C++ may be used to set up automation plants or automate conveyer belt system operations for a particular project or product line.

More OpenCV advancements and applications are expected in the following areas:

- 1) **Industrial Automation:** Computer vision is increasingly being used in industrial automation to increase quality control, inspection, and process optimization
- 2) **Security and Surveillance:** Computer vision is extremely important in security and surveillance systems. The object detection, tracking, and behavior analysis capabilities of OpenCV help to construct intelligent video surveillance systems.
- 3) **Real-time and Embedded Systems:** As smart devices and the Internet of Things (IoT) proliferate, there is an increasing demand for real-time computer vision applications that can run efficiently on resource-constrained devices.
- 4) **Integration with other Technologies:** Deep learning techniques, specifically convolutional neural networks (CNNs), have transformed computer vision. TensorFlow and PyTorch are two popular libraries that have been integrated. In augmented reality (AR) and virtual reality (VR) applications, computer vision is critical.

Finally, with its broad features, cross-platform compatibility, active community support, and integration capabilities, the future of OpenCV implemented in C++ appears excellent. While it has a steep learning curve and only limited support for advanced approaches, its excellent performance and versatility make it a powerful tool for a wide range of computer vision applications.

V. REFERENCES

- [1] <https://pyimagesearch.com/2014/12/15/real-time-barcode-detection-video-python-opencv/>
- [2] <https://www.ijser.org/researchpaper/Naive-Bayes-Approached-in-Color-Detection-using-Pandas-OpenCV.pdf>
- [3] <https://laconicml.com/barcode-detection-opencv/>
- [4] <https://pyimagesearch.com/2018/05/21/an-opencv-barcode-and-qr-code-scanner-with-zbar/>
- [5] https://link.springer.com/chapter/10.1007/978-3-030-20904-9_9
- [6] papers.ssrn.com/sol3/Delivery.cfm/SSRN_ID3879981_code3665699.pdf?abstractid=3879981