Introduction to Programming through C++ (UE18EE400SA)

Applications of Computer Vision with OpenCV in C++

Team Members

Name	SRN	Dept
Deeksha H Kulal	PES1201800501	ECE
Archana K	PES1201801697	ECE
Padmapriya V	PES1201800759	EEE

Abstract

As we know, OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

We intend to use OpenCV in C++ for execution of the computer vision applications as C++ is proven to be an excellent language for the implementation and creating real-world products.

We would like to work starting from the basics of image processing from as simple as capturing an image through the webcam and some basic functions. Then we would like to move forward towards the following:

1. Resize and Cropping of an image

2. Drawing Shapes

3. Warp perspective

4. Color and Contour Detection, etc.

After understanding the fundamentals through the above topics, we would like to proceed towards the real-time applications wherein the above-mentioned fundamentals could be used for our benefits.

This could include various applications like the following:

1.QR Code Scanner

2. Virtual Painter

3. Document Scanner

4. License Plate Detector, etc.

Hence as a whole we would like to get an overall insight of how exactly OpenCV could be used in C++.

Problem Statement

To deal with the applications of computer vision with OpenCV in C++.

Proposed solution

We hope to help them visualise the steps involved in image processing techniques using OpenCV in C++ by providing a step by step breakdown of the process.

In our project we aim to deal with the applications of Computer Vision using OpenCV in C++.In this process, we would like to demonstrate a virtual painter, document scanner, QR Code scanner and License plate detector.

Virtual Painter is a CPP program that uses OpenCV to perform video processing through a webcam. The video shows color points according to color on the video.

Document Scanner is a very useful tool that can be used to scan a document and crop it by excluding it from all other backgrounds. Similarly QR Code scans a particular QR Code and returns the information encoded within it.

License plate detector detects the license plates in a given image or in a video, captured using a webcam in this case. This idea plays a very important role mainly for security purposes.

Software Used

OpenCV

Visual Studio is a software development tool that has been available for more than two decades. Visual Studio 97 was the initial version. Since then, many new versions have been released, the most recent of which being Microsoft Visual Studio 2022.

The Visual Studio IDE (integrated development environment) is a piece of software that allows programmers to write and edit code. Its user interface is used to edit, debug, and build code in software development. Visual Studio comes with a code editor that supports IntelliSense (code completion) and code refactoring. The integrated debugger can be used as both a source-level and a machine-level debugger. A code profiler, a designer for creating GUI applications, a web designer, a class designer, and a database schema designer are among the other built-in tools.

Visual Studio advantages are as follows

- For C++ it's one of the best IDEs and it supports many languages
- It has many extensions
- It has a free version
- It is used by a lot of developers

C++

C++ is a free-form programming language that supports procedural, object-oriented, and generic programming. It is statically typed, compiled, and case-sensitive.

Because it combines both high-level and low-level language capabilities, C++ is classified as a middle-level language.

Bjarne Stroustrup began developing C++ as an upgrade to the C language in 1979 at Bell Labs in Murray Hill, New Jersey, and it was originally known as C with Classes until being renamed C++ in 1983.

C++ is a superset of C, which means that almost any lawful C programme is also legal in C++.

Document Scanner

Document Scanner is a type of input device. On paper, the data or information is written. The scanner is fed with the paper. The written information on paper is converted to an electronic version, which is then stored in a computer. Text, handwritten content, and image extras can all be found in the input papers.

Because the document was saved on a computer, it could be kept safe for a longer period of time. The document will be kept on file indefinitely. We can make changes to the paper as needed. When necessary, the document can be printed.

The steps that we need to follow to build this project are:

1. Convert the image to grayscale
2. Find the edges in the image
3. Use the edges to find all the contours
4. Select only the contours of the document
5. Apply warp perspective to get the top-down view of the document.

Initially the image is read from the user's directory.Then it is resized to the required dimensions. A small note on how the image is read and resized is given below.

Read, Display and Write an Image using OpenCV

Image processing and computer vision require the ability to read, display, and write images. Even if we are processing photographs by cropping, resizing, rotating, or adding different filters, we will need to read them first. Hence it is important we understand these fundamental processes.

There are three built-in functions in OpenCV, the world's largest computer vision library, and theses are:

1. imread() is a function that allows us to read an image.
2. imshow() is a function that shows an image in a window.
3. imwrite() creates a file in the specified directory.

These take the parameter as filename of the image which is to be read from a file, displayed or written into a file.

Resize and Cropping of an image

To resize an image, either we scale it along each axis (height and width) using the scale factors supplied, or simply select the desired height and width.When resizing an image, we must keep the following things in mind:

- It's crucial to remember the image's original aspect ratio (width by height) if we wish to retain it the same in the scaled version.
- If we want to shrink the size of an image, we will have to resample the pixels.
- Increasing the size of an image necessitates its reconstruction. This necessitates the creation of new pixels using interpolation.
- To complete these procedures, various interpolation techniques are used. In OpenCV, there are several approaches to choose from, and the best one depends on the application.

After the image is read, it is necessary to perform some preprocessing operations. Firstly the image is converted from grayscale.Then the image is blurred using Gaussian Blur and Canny edge detection techniques are used to detect its edges.A rectangular structuring element is defined to be the kernel using which the image is dilated.

Conversion of Image into Grayscale

We use the function cvtcolor() to convert the image to it's Grayscale equivalent. This function does require changes to the Red, Green, Blue variants of color. cvtColor() is a method that is used to convert an image from one color space to another.

Image Blurring

The image is blurred by using a low-pass filter kernel to convolve it. It's good for getting rid of noises. It really removes high frequency material from the image (such as noise and edges). As a result, the margins are slightly blurred in this procedure. There are primarily four types of blurring techniques available in OpenCV. We intend to use the Gaussian Blur.

Gaussian Blurring

In this, instead of a box filter, a gaussian kernel is used. We use the function:

cv.GaussianBlur (src, dst, ksize, sigmaX, sigmaY = 0, borderType = cv.BORDER_DEFAULT)

Parameters

src	Input image and the image can have any number of channels,
dst	output image of the same size and type as src.
ksize	blurring kernel size.
sigmaX	Gaussian kernel standard deviation in X direction.
sigmaY	Gaussian kernel standard deviation in Y direction
borderType	pixel extrapolation method

Edge Detection Using OpenCV

Edge detection is a type of image processing that identifies the boundaries (edges) of objects or regions within an image. Edges are one of the most significant characteristics of photographs. As a result, edge detection is widely used in computer vision processing pipelines.

Canny Edge Detection

Because of its robustness and flexibility, Canny Edge Detection is one of the most widely used edge detection technologies today. For extracting edges from an image, the algorithm uses a three-stage method. Then there is image blurring, which is an essential preprocessing step for reducing noise. This results in a four-stage procedure, which contains the following steps:

1. Noise Reduction
2. Calculating Intensity Gradient of the Image
3. Suppression of False Edges
4. Hysteresis Thresholding

Image Dilation

This technique entails convolving an image A with a kernel (B), which can be any form or size but is most commonly a square or circular. Here a square kernel is used.
The kernel B has a defined anchor point, which is usually the kernel's centre.
We compute the maximal pixel value overlapped by B as the kernel B scans over the image and replaces the image pixel in the anchor point position with that maximal value. As you might expect, this maximising procedure causes bright areas in an image to expand.

In the next step, all the contours in the image are found out and the biggest among them are returned as a vector of points.

Image Contouring

Using contour detection, we can detect the borders of objects, and localize them easily in an image.
A contour is created by joining all of the points on an object's boundaries. A specific contour is usually defined as a set of boundary pixels with the same hue and intensity. OpenCV makes finding and drawing outlines in images easy. It has two straightforward functions:

1. findContours()
2. drawContours()

Also, it has two different algorithms for contour detection:

1. CHAIN_APPROX_SIMPLE
2. CHAIN_APPROX_NONE

Here we use the findContours() function.

image	The binary input image
mode	contour-retrieval mode **RETR_EXTERNAL** If we use this flag, it returns only extreme outer flags. All child contours are left behind.
method	contour-approximation method **CHAIN_APPROX_SIMPLE** It compresses horizontal, vertical, and diagonal segments and leaves only their end points

Then we loop through all the contours and find the area of all contours using the contourArea function. We can consider further operations only for contours above a specific area.Then to find the bounding box around these objects we have to first determine its perimeter which is done using the arcLength function which accepts the contours and also a boolean flag of whether the shape is open or closed.The approxPolyDP function helps us to find the number of corner points the particular object has. So all the contours are iterated through and the biggest contour with size as four and maximum area is returned.

ApproxPolyDP

The technique of approximating a contour shape to another shape with a smaller number of vertices in such a way that the gap between the contours of the shapes is equal to or less than the given precision is known as approximation of a contour shape. And the approxPolyDP() function in OpenCV is used to approximate the shape of polygonal curves to the specified precision, and approxPolyDP() returns the approximated contour whose shape is the same as the input curve. Approximation of a shape of the contour is widely used in the field of robotics to classify patterns and analyse scenes.

cv.approxPolyDP (curve, approxCurve, epsilon, closed)

Parameters

curve	Input vector of 2D points stored in **cv.Mat**.
Approx Curve	Result of the approximation. The type should match the type of the input curve.
epsilon	Parameter specifying the approximation accuracy. This is the maximum distance between the original curve and its approximation.
closed	If true, the approximated curve is closed (its first and last vertices are connected). Otherwise, it is not closed.

A function draw points is created which draws circles at the detected edge points of conPoly and it is also numbered. The conPoly does not provide us with the exact order of how the document corners are to be considered. They can be varying each time we execute the program. Hence, reordering has to be done considering the maximum and minimum values obtained when the x and y coordinates are either added or subtracted. So two vectors of integers called the SumPoints and SubPoints are created where the sum and difference of the coordinate points are to be stored.

Edge to be numbered point 0 would have the minimum value in SumPoints and point 1 would have the minimum value in SumPoints. Similarly, edge to be numbered point 2 would have the minimum value in SubPoints and point 3 would have the maximum value in SubPoints.

Image Warping

It is necessary to get our document end points and adjust it to the size of an A4 sheet.
The source points are defined to be the points obtained after reordering and the destination points are those coordinates corresponding to the width and height of the A4 sheet defined.
getPerspectiveTransform()
Parameters

src	Coordinates of quadrangle vertices in the source image.
dst	Coordinates of the corresponding quadrangle

	vertices in the destination image.
solveMethod	method passed to cv::solve (DecompTypes)

warpPerspective()

Parameters

src	Input image
dst	Output image that has the size and the same type as src .
M	3×3 transformation matrix.
dsize	size of the output image.
flags	combination of interpolation methods
borderMode	pixel extrapolation method
borderValue	value used in case of a constant border; by default, it equals 0.

Result

Original Image

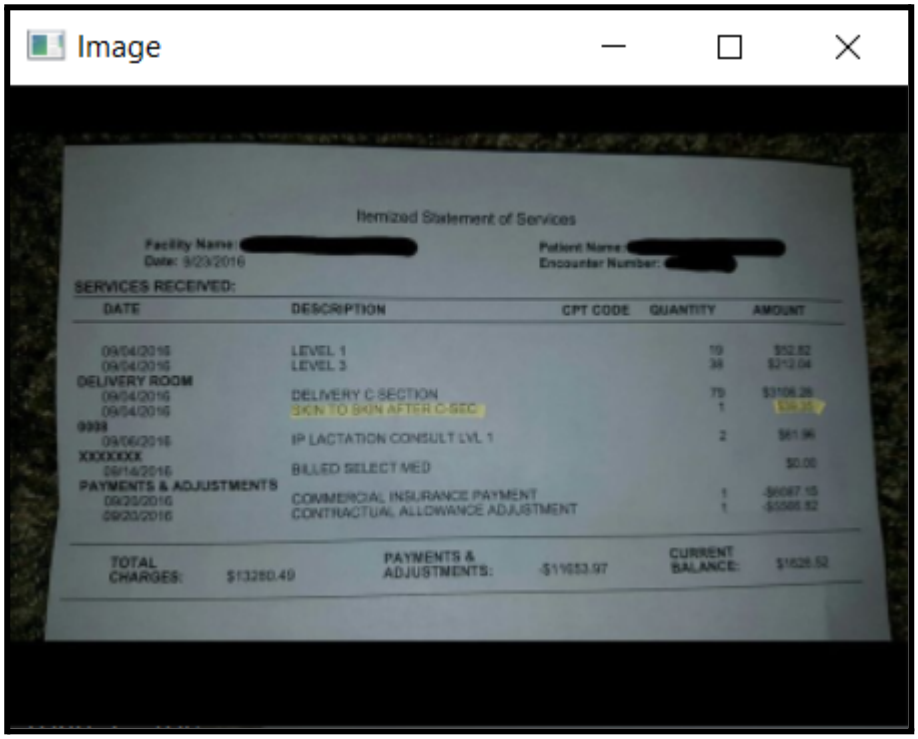
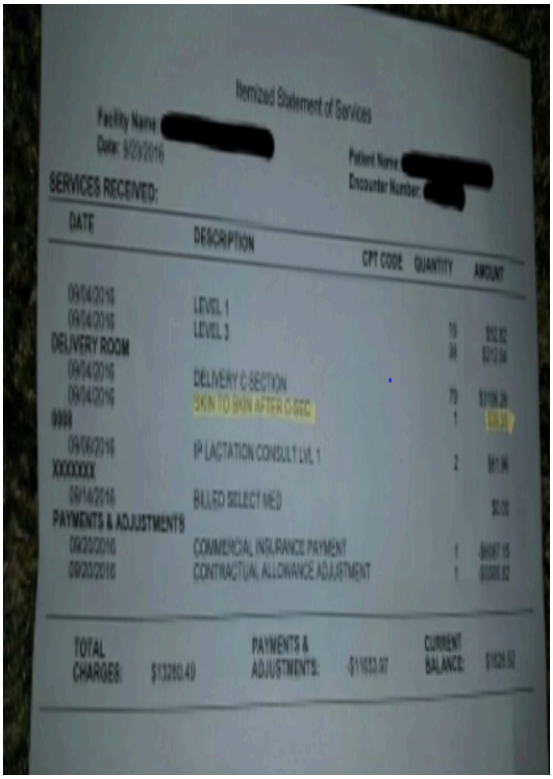


Image after Scanning



QRCode Scanner

A quick response (QR) code is a sort of barcode that encodes information as a series of pixels in a square-shaped grid and can be read simply by a digital device. QR codes are often used in marketing and advertising efforts to monitor information about products in a supply chain.

Firstly, the image is read and if the corresponding frame is found to be empty, then a message appears that the frame load failed.The frame is then preprocessed using the cvtColor function to convert the image to grayscale.
The QRCodeDetector initializes an object detector that is used to process the qr code and the information obtained from it is stored in a string and displayed.

Result

```
Decoded Data : http://www.google.com
```

Virtual Painter

Virtual Painter algorithm is simply printing the colored points on video according to the color of the video, these colored points are stored in a vector according to their position. The video is stored as a sequence of images.
The algorithm code converts the incoming image into HSVspace,which is a very suitable color space and perfect for color tracking.
The tracking bars have organized the HSV values into the required color range of the colored object .

Contour detection of the Mask of Colour Object.

After detecting the mask in the air canvas,it is now time to locate its center position for drawing the line. The system will perform some morphological operations on the Mask to make it free of impurities and detect contour easily.
Drawing the line using the position of the contour
The position of the outline in each subsequent frame is memorized and we will use these accumulated points to create a line using OpenCV's drawing capabilities.

The steps in this algorithm are depicted below:
Step 1: Find the color range of the target object and save it.
Step 2: Apply the correct morphological operations to reduce noise in the video
Step 3: Detect and track the colored object with contour detection.
Step 4: Find the object's x,y location coordinates to draw on the screen.

Any number of color pens can be used . Inorder to detect more color pens:

1. Use ColorPicker program to pick HSV of color to add more color as needed.
2. Add the HSV values to the colors list in VirtualPaint.cpp and the RGB values of the color to myColorValues at respective index.

The program works as follows:

- The system using the web camera records the video of the user who holds an object (usually a marker)of a particular color as a tool used for drawing or writing.
- This video is displayed and the user holds this object in front of the camera so that the system can detect the color of the object on a real time basis.
- The algorithm then starts tracing the movement of the object in the user's hands and paints the area (pixels) according to it.
- The painting occurs with the color detected by the system in the initial phase.

Important Terminologies

Scalar - It is a template class for a 4-element vector derived from Vec. The type Scalar is widely used in OpenCV to pass pixel values.

RETR_EXTERNAL- retrieves only the extreme outer contours.

CHAIN_APPROX_SIMPLE- compresses horizontal, vertical and diagonal segments and leaves only their end points.

Mat- Mat class of OpenCV library is used to store the values of an image. It represents an n-dimensional array and is used to store image data of grayscale or color images, voxel volumes, vector fields, point clouds, tensors, histograms, etc.

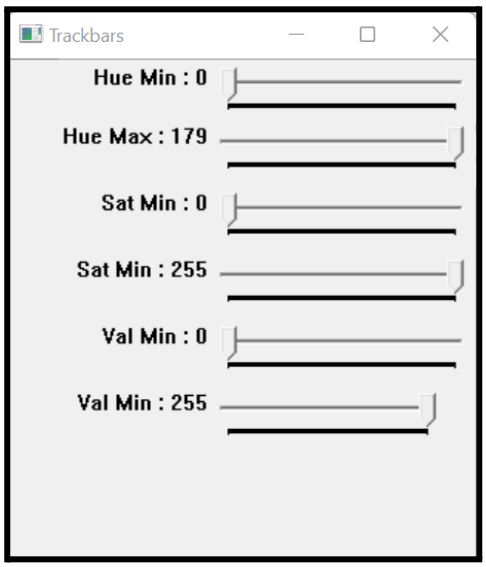
Inbuilt Functions

findContours( )- The function retrieves contours from the binary image using the algorithm.
contourArea( )- The function computes a contour area. (area computed using Green's formula).
arcLength( ) - The function computes a curve length or a closed contour perimeter.
approxPolyDP( ) - This function approximates a curve or a polygon with another curve/polygon with less vertices so that the distance between them is less than or equal to the specified precision. It uses the Douglas-Peucker algorithm.
inRange( ) - Checks if the array elements lie between the elements of two other arrays.
VideoCapture( ) - Open video file or image file sequence or a capturing device or an IP video stream for video capturing.

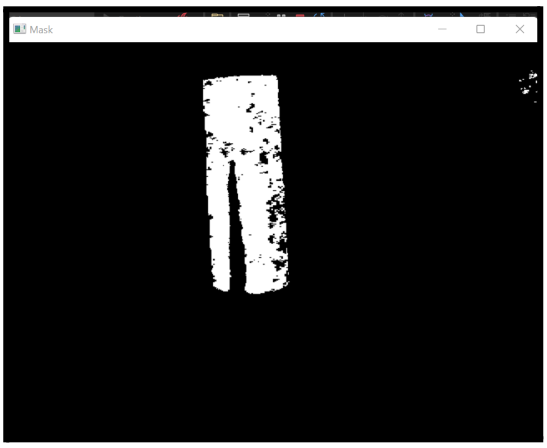
cvtColor( ) - Method is used to convert an image from one color space to another. In our program , we convert the image to HSV color space.

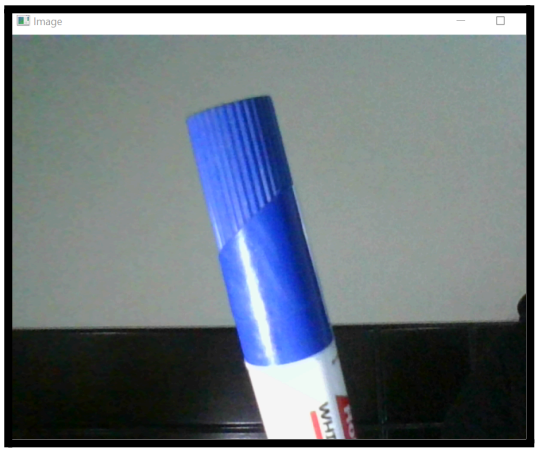
```
cvtColor(img, imgHSV, COLOR_BGR2HSV);
```

Color TrackBar

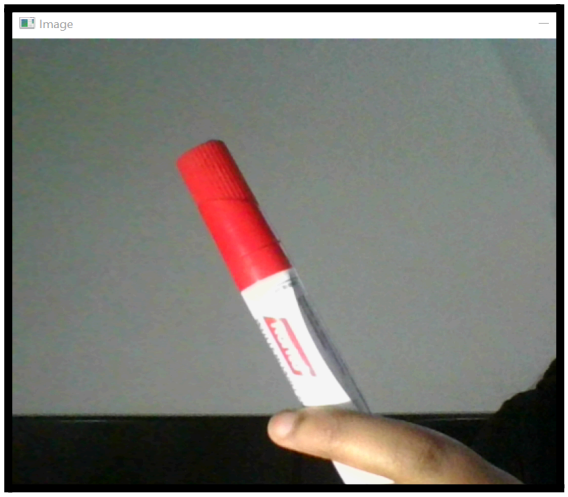


Mask Detection

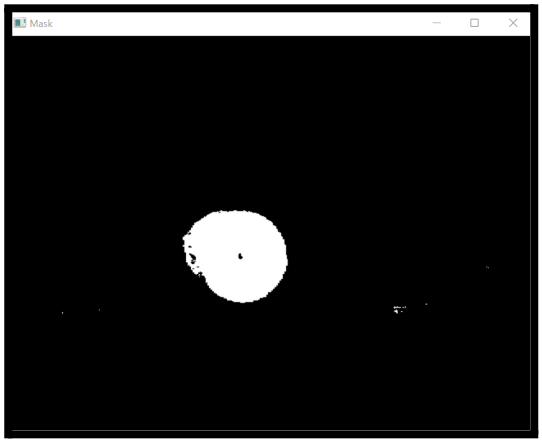


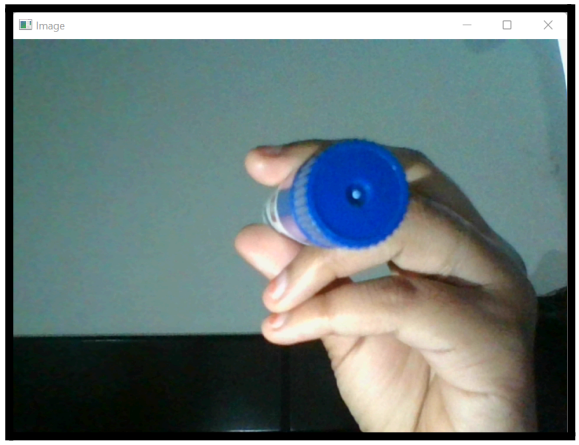



Mask

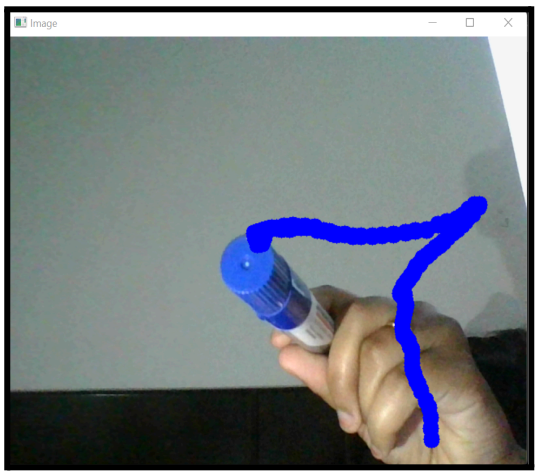


Image

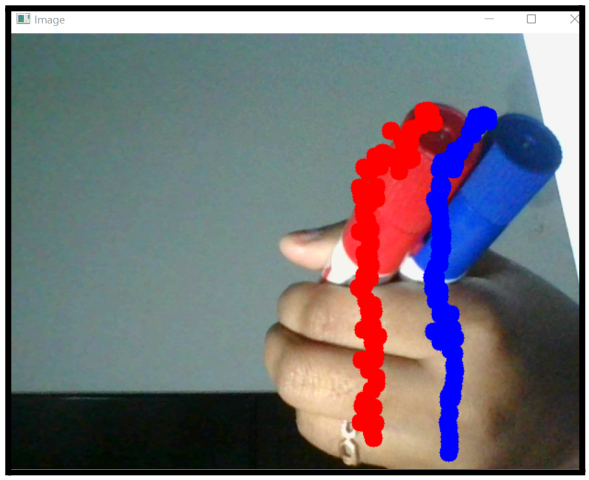

Mask



Image



Image



Image

License Plate Detector

The main aim in this section of the project is to build a mechanism to detect the License plates in a given image or in a video, captured using a webcam in this case. This idea plays a very important role mainly for security purposes.

The code has been designed such that it has various commands that can be entered in the terminal and each command performs a specific functionality to help in the process of detection of the plates.The functions implemented are as follows:

showImage(string imagePath)

Displays the image from the name provided by the user in the terminal. It must be ensured that the image must be present in the same folder as the one in which the code resides.

The argument ‘ imagePath ’ simply refers to the name of the image file.

detectOnly(string imagePath)

This function detects the license plates from a given image. OpenCV provides a training method or pre-trained models which can be read using the cv::CascadeClassifier::load method. The ‘CascadeClassifier’ function of OpenCV is used to point to the location where the XML file is stored. Afterwards, the detection is done using the cv::CascadeClassifier::detectMultiScale method, which returns boundary rectangles for the detected parts.

The Haar Cascade classifier is an effective object detection approach which was proposed by Paul Viola and Michael Jones in their paper, “Rapid Object Detection using a Boosted Cascade of Simple Features” in 2001. This is basically a machine learning based approach where a cascade function is trained from a lot of images both positive and negative. Based on the training it is then used to detect the objects in the other images.

The main arguments to be passed in the function detectMultiScale() are as given below:

image	Matrix of the type CV_8U containing an image where objects are detected.
objects	Vector of rectangles where each rectangle contains the detected object, the rectangles may be partially outside the original image.
scaleFactor	Parameter specifying how much the image size is reduced at each image scale.
minNeighbors	Parameter specifying how many neighbors each candidate rectangle should have to retain it.

detectLicensePates(string imagePath)

Here we call the ‘detectOnly’ function and then take care of the visual output of the detected plates in the image using the rectangle() function available.

The arguments to be passed to the function rectangle() are as follows:

img	Image.
pt1	Vertex of the rectangle.
pt2	Vertex of the rectangle opposite to pt1 .
color	Rectangle color or brightness (grayscale image).
thickness	Thickness of lines that make up the rectangle. Negative values, like FILLED, mean that the function has to draw a filled rectangle.
lineType	Type of the line.
shift	Number of fractional bits in the point coordinates.

cropLicensePlate(string imagePath)

This function is used to crop the detected license plate numbers in the image and display each of them separately.

saveCropImage(string imagePath)

This function is used to save the cropped images detected previously by the cropLicensePlate() function.

saveDetect(string imagePath)

This has a similar functionality like the saveCropImage() function but the only difference being that it saves the whole image with the detected plates.

cameraPlateDetection()

Here the detection of the license plates in a video is implemented. A webcam was used to capture video in this project.

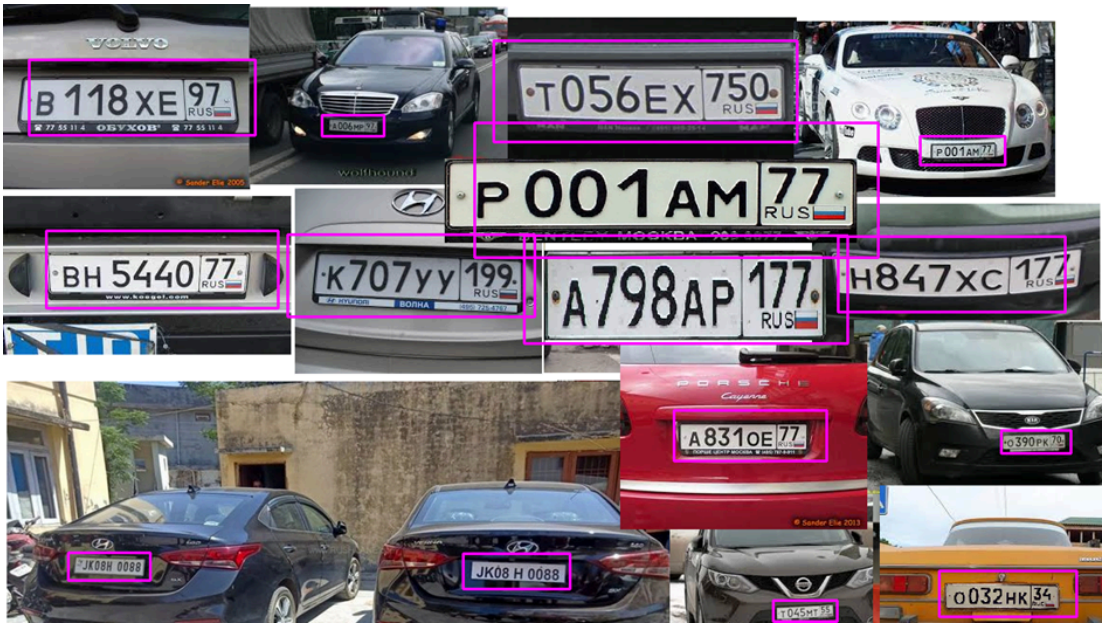
Videocapture() is used for video capturing from video files, image sequences or cameras. This class provides C++ API for capturing video from cameras or for reading video files and image sequences.

The plates are detected in every frame of the video captured and displayed on the screen in a similar fashion to detection in a static image.

The options displayed on the terminal for the user appear as below.

```
image or camera (or type help to know more)------> help
----------------License Plate Detector-------------
1. show
2. detect
3. crop
4. save_detect
5. save_crop
6. exit
-----------------------------------------------------
```

The output is as shown below.



The cropped images are saved in a folder as shown.

Bibliography

https://learnopencv.com/

Saoji, Saurabh & Vidyapeeth, Bharati & Choudhary, Akash & Phogat, Bharat. (2021). Basic Paint Window Application via Webcam Using OpenCV and Numpy in Python. Journal of Interdisciplinary Cycle Research. XIII. 1680-1686.