



# Implementation of Media Player using Image Processing and OpenCV JavaFx library

Gautami Shingan<sup>1</sup>, Ajay Paratmandali<sup>2</sup>, Yadnyawalkya Tale<sup>3</sup>

Department of CSE  
RIT, Rajaramnagar, Sangli, MS, India

## Abstract:

Open Source Computer Vision acronym by OpenCV is a library programming function for image and video analysis, originally developed in 2000 by Intel's research lab from Russia and later maintained by Itseez lab. The library is cross-platform and free for use under the open-source BSD licensing which supports Windows, Linux, Android, iOS and almost all operating systems. It has more than 2500 optimized algorithms adopted by all around the world, OpenCV has more than 7 million overall downloads where usage ranges from interactive art, to mines inspection, stitching maps on the web on through advanced robotics [1]. In this paper we have describe way to build media player which detects, recognize color image from webcam and responses to particular functions with the help of described algorithms.

**General Terms:** Image Processing

**Keywords:** OpenCV library, Media Player, Color detection, Color recognition, JavaFx.

## 1. INTRODUCTION

The main pitfalls in the advance image processing applications are limited memory and CPU resources in order to store the images and the intermediate calculations [2]. Furthermore, webcam capturing stream has continues changing of frame. There is limitation for laptop in fast recognition, high quality image capture from webcam, etc. of each frame without including external Library.

OpenCV is an open source library, popularly used in image processing applications to attain quick acknowledgement for embedded applications. OpenCV library is written in C, a middle-level language which supports with many embedded platforms. Library includes primary image tasks, logical & arithmetic operations and also many complex operations like detections and tracking of an objects. Elementary components developed by Intel also support for many languages such as Java, C, C++, Python, Perl, Ruby, Android, etc.

This paper describes efficient way for image processing applications such as media player which runs on portable platforms. Obtained result demonstrates that video streams can be handle by color scan of webcam after converting normal images to clarified images using sharpening mechanism mentioned in paper.

Section 2, describes basic information about media player design. Section 3, describes previous work performed in OpenCV platform. Section 4, describes importance of JavaFX over Java programming language. Section 5, describes various algorithms used in in implementation of media player. Section 6, describe features of application and limitation faced during implementing media player.

## 2 BASIC INFORMATION

### 2.1 Media Player

In the era of entertainment, Media Players plays vital role to use media such as audios, videos and even Images. It helps to play mentioned media with extra functionalities. In this project, we had tried to break traditional methods by using image processing functions to Media Player. All designing features of Image Processing Media Player (IPMP) made by

JavaFX. This paper based on player which responds to color cards through color detection and recognition algorithms mentioned in Section 5. Secondary concerns of paper resides into social view. This paper talks about media player which tried to give solution to people who are disable to perform actions.

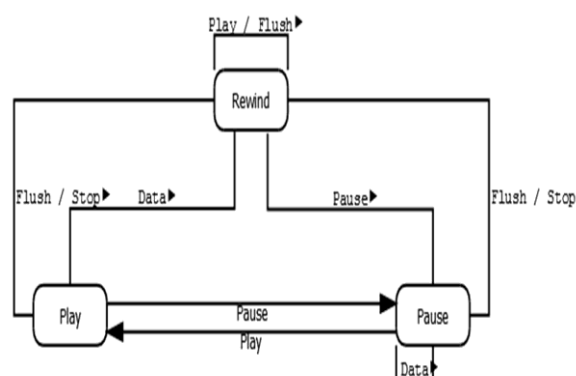


Figure 1: Data flow for media player

### 2.2 Functional Model

IPMP includes all general functionality of media player such as play, pause, rewind, forwarding, backwarding and also playlists, but all performed by image processing mentioned in, figure 3.

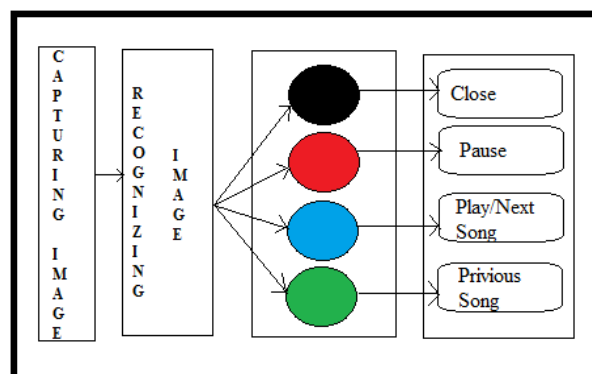


Figure 3: dataflow for image processing

### 3. PREVIOUS WORKS ON OPENCV

Since OpenCV is open source library having many function for image processing, which widely used in many application. Shao-jie Sun and Xiao-dong Yang are worked on flexible high-precision Charge Coupled Device (CCD) camera calibration method suitable for industrial field [3]. Souhail Guennouni, Ali Ahaitouf and Anass Mansouri were developed application for multiple object detection using OpenCV on an embedded platform [4]. Xianghua Fan, Fuyou Zhang, Haixia Wang and Xiao Lu are also used OpenCV for system of face detection, where they presented modified AdaBoost algorithm based on OpenCV [5]. An intelligent security system that provides a high level of home security using visual surveillance is developed and explored by Syadus Sefat, Abdullah Al Mamun Khan and Shahjahan [6]. Ashwini Awate and Damini Deshmankar introduce a technique which will diagnose and classify external disease within fruits by using color, texture analysis [7].

### 4. JAVA FX OVER JAVA

In the 1990s, Sun Microsystems created a new Computer Language called Java [8]. Many languages have issue about Memory allocation, speed and porting as well as complex syntax which not exist with Java. Thus, Java is very popular language.

Java have Swing library for creating program with graphical user interface (GUI). But, many developer face problem in swing library because of over complex syntax and Java on the desktop never really took good effect as expected. Sun Microsystems tried several ways to make it easier to create Java applications. One of these was a scripting language called JavaFX 1.0 which tried to compete with Adobe Flash and Microsoft Silverlight [8]. By, using JavaFX we can build more complex structured GUI than Java swing packages.

Initially JavaFX was not Java rather than it was just scripting language. Oracle acquired Sun Microsystems declare JavaFX not a scripting language but added these JavaFX functionality into Java. Due to this, today it is possible to integrate FXML file with JavaFX. Thus, by combining JavaFX with java you can separate the User interface from the code, you can use JavaFX designer called SceneBuilder to build complex GUI design in FXML. Thus, With JavaFX, you can build applications like complex user interface that features audio, video, graphics, and animation.

### 5. ALGORITHMS

#### 5.1 Image capturing algorithm

OpenCV library provide feature for capturing image with webcam. We have created algorithm for continues image capturing until application get closed. In one second it capture nearly 5 to 10 image and also provide delay at specific source to control speed of capturing image per second.

Figure 4: Algorithm for capturing image with OpenCV

#### 5.2 Image Sharping

Image sharpening is important term for texturing an image. Technically Image get burled every time because of low quality of web camera so sharping necessary for color identification. Sharping algorithm help image to adjust contrast and fix blur problem.

Figure 5: Algorithm for sharping image with OpenCV [9]

#### 5.3 Color detection algorithm

Color detection algorithm helps to calculate color of an image. In first stage, algorithm calculates RGB value

of each pixel depending on image height and width. To determine dominant color it take average of all RGB value for each pixel.

```
#import highgui from opencv
#load native library
System.loadLibrary( Core.NATIVE_LIBRARY_NAME );
#read image and store in Mat as source
Mat source =
Highgui.imread("sharp.jpg",Highgui.CV_LOAD_IMAGE_COLOR);
#store destination image of row,coloum and type
Mat destination = new Mat(source.rows(),source.cols(),source.type());
#blurs an image using a Gaussian filter
Imgproc.GaussianBlur(source, destination, new Size(0,0), 10);
#apply image watermark by using addWeighted
Core.addWeighted(source, 1.5, destination, -0.5, 0, destination);
#write destination image
Highgui.imwrite("sharp.jpg", destination);

#read image file
File file= new File("action.jpg");
#stored in BufferedImage as image and read it
BufferedImage image = ImageIO.read(file);
#calculate hight and width of image
int height = image.getHeight();
int width = image.getWidth();
#take avarage of image
for(int i=0; i < width ; i++)
{ for(int j=0; j < height ; j++){
#get RGB value of pixel at (i,j)
clr= image.getRGB(i,j);
red = (clr & 0x00ff0000) >> 16;
green = (clr & 0x0000ff00) >> 8;
blue = clr & 0x000000ff;
#calculate total red, blue, green color in image
sumred = red + sumred ;
sumgreen = green + sumgreen ;
sumblue = blue + sumblue ;
}}
#detect avarage red, blue, green color of image
sumred = sumred/(height * width) ;
sumblue = sumblue/(height * width) ;
sumgreen = sumgreen/(height * width) ;
```

Figure 6: A Algorithm to detect dominant color of image

#### 5.4 Color recognition algorithm

By comparing dominant color of RGB values, we can easily determine color of an image and accordingly application takes action to respective color. For example, for red image - media stream pause application, for green - media stream resumes or if already resumed then plays next song, for blue - it play previous song and for black - application closes.

```
#import highgui from opencv
#extends Thread to class for run in background
#load Native library
System.loadLibrary(Core.NATIVE_LIBRARY_NAME);
#import VideoCapture as camera with default camera 0
VideoCapture camera = new VideoCapture(0);
#import Mat as frame to store image
while(true)
{camera.read(frame);
#write image as action.jpg
Highgui.imwrite("action.jpg", frame);
}
```

```

#from color detection algorithm sumred,
sumblue and sumgreen are avarage red green
blue color of image
#recognitize red color image
if (sumred>180 && sumred>sumgreen &&
    sumred>sumblue)
{ #red color image  pause();}
#recognitize green color image
if (sumgreen>180 && sumgreen>sumred &&
    sumgreen>sumblue)
{ #green color image
  if(pause)
    resume
  else
    play_next_song(); }
#recognitize blue color image
if (sumblue>180 && sumblue>sumgreen &&
    sumblue>sumred)
{ #blue color image  play_prvious_song();}
#recognitize black color image
if (sumred<20 && sumgreen<20 &&
    sumblue<20)
{ #black color image  close(); }

```

Figure 7: Algorithm to recognize color of image

## 6. FEATURES AND LIMITATIONS

### 6.1 Features:

- **Formats:** Due to JavaFx library supported formats have limitation. It supports only those formats which support by library. Thus, supported formats for audio are mp3, mp3.1, wav as well as for video are mp4, flv. This player doesn't support compressed formats like AVI, MKV, etc.
- **Functionality Support:** Supports functionality like pause, resume, fast-forward, jump-forward, jump-backward and rewind also advanced functionality like Time sliders, Volume Slider, drag n drop, full screen etc. User have access to play any video from storage device and select playlists. Last but not least manual provides help options to use this player.
- **Attractive GUI:** JavaFX used for building attractive Graphical User Interface [GUI]. It supports FXML file which helpful for creating GUI interface. JavaFX Scene Builder 2.0 is used to create FXML file. Use of this Scene Builder is very easy and provides drag drop functionality.
- **Image processing support:** This technology helps to recognize color for Taking action. We initially provided color are Red for pause, Green for resume as well as play next song, Blue for playing previous song and Black for close application.
- **Web cam access:** Media player also access webcam for continually capturing image for image processing.

### 6.2 Limitation:

- Due to limited library, media player doesn't allow video and audio formats. According to our research, compressed formats are not supported by this media player. But, user also uses third-party software like **FormatFactory** to convert video into specific format.
- This software select black color for close application. Thus, in dark room media player probably scan black color and shutdown application.

- Sometimes, webcam capture image which not clear or low quality color may be different than expected but there probability is very low. Due to wrong color scanning, media player take wrong action. To minimize this problem use minimum 2.5 mega pixel quality webcam. Also, media player perform sharp operation to sharp every captured image which help to minimize this problem.

## 7. CONCLUSION AND FUTURE WORK

In this paper we represent, media player application that simply capture image, scan color of image and take action according to color. Here, we use image captureing method in background by simply use of multithreading. In this report we explain algorithm for capturing image, image detection and image recognition.

By this project we understand new image proccessing library OpenCV as well as JavaFx which is advanced java library for complex graphical user interface. Now, our feature work is to incresses formats of audio and vedio as well as improve range of showing cards.

## 8. REFERENCES

- [1] Accessible at <http://sourcefouge.net/projects/opencvlibrary>. (On May 20, 2016)
- [2] IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p- ISSN: 2278-8735. Volume 9, Issue 2, Ver. II (Mar - Apr. 2014), PP 73-79 [www.iosrjournals.org](http://www.iosrjournals.org)
- [3] Accessible at [http://ieeexplore.ieee.org/xpl/abstractAuthors.jsp?arnumber=7373784&queryText=opencv&ranges=2012\\_2016\\_Year](http://ieeexplore.ieee.org/xpl/abstractAuthors.jsp?arnumber=7373784&queryText=opencv&ranges=2012_2016_Year) (On May 20,2016)
- [4] Accessible at [http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7016649&queryText=opencv&ranges=2012\\_2016\\_Year](http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7016649&queryText=opencv&ranges=2012_2016_Year) (On May 20,2016)
- [5] Accessible at [http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6242980&queryText=opencv&ranges=2012\\_2016\\_Year](http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6242980&queryText=opencv&ranges=2012_2016_Year) (On May 20, 2016)
- [6] Accessible at [http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6850818&queryText=opencv&ranges=2012\\_2016\\_Year&pageNumber=2](http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6850818&queryText=opencv&ranges=2012_2016_Year&pageNumber=2) (March 20,2016)
- [7] Accessible at [http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7380603&queryText=opencv&ranges=2012\\_2016\\_Year&pageNumber=3](http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=7380603&queryText=opencv&ranges=2012_2016_Year&pageNumber=3) (March 20, 2016)
- [8] Accessible at <https://blog.idrsolutions.com/2014/11/difference-java-javafx/> (March 20, 2016)
- [9] Accessible at [http://www.tutorialspoint.com/java\\_dip/enhancing\\_image\\_sharpness.htm](http://www.tutorialspoint.com/java_dip/enhancing_image_sharpness.htm) (March 20,2016)