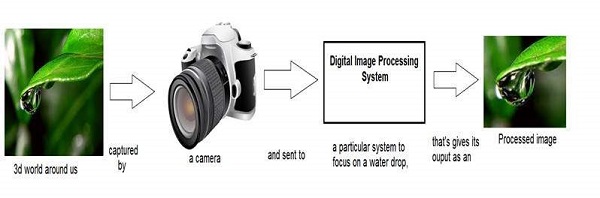
# **Project Title**

# **Image Processing in java**

***Introduction:***

Digital Image Processing (DIP) deals with manipulation of digital images using a digital computer. It is a sub field of signals and systems but focuses particularly on images. DIP focuses on developing a computer system that is able to perform processing on an image. The input of such system is a digital image. The system processes the image using efficient algorithms, and gives an image as an output.



Java is a high level programming language that is widely used in the modern world. It can support and handle digital image processing efficiently using various functions and classes and the most widely used class is **BufferedImage** class which is a subclass of Image class.

**BufferedImage Class:**

*It is used to handle and manipulate the image data. A BufferedImage is made of ColorModel of image data.*

Functions of this class which have most used in the project are:

* **getRGB(int x, int y)**: It returns an integer pixel value in the default RGB color model (TYPE\_INT\_ARGB) and default sRGB colorspace.
* **getType()**: It returns the image type
* **getWidth()**: It returns the width of image.
* **getHeight()**: It returns the height of image.

***Algorithms Implemented:***

The purpose of our project is to implement some of the image processing algorithms. The above 2 to 8 UDF’s are Image Filters and rest of the UDF’s are image modifying operations.

1. **Gray Scale Conversion**

We have implemented two algorithms for Gray Filters:

1.1 Gray Luminous

1.2 Gray Average

The difference between these is that Gray Luminous retains the brightness of the image.

1. **Primary Color Image**

We have implemented three algorithms for Primary Color Filters:

2.1 Red Color Image

2.2 Blue Color Image

2.3 Green Color Image

For 2.1, it takes all the pixels of image and removes the Green and Blue pixels of image.

For 2.2, it takes all the pixels of image and removes the Green and Red pixels of image.  
 For 2.3, it takes all the pixels of image and removes the Red and Blue pixels of image.

1. **Sepia Image**

It is one of the image filters which are very popular.

1. **Mirror Image**

It takes an image and return an image which is a mirror form of that image.

1. **Negative Image**

Negative Image Algorithms takes the pixel of image and subtract its red, green and blue component from 255 for each and every pixel.

1. **Translation**   
   Translation Algorithm requires two inputs from user to translate image on x and y- axis parallel from top left bottom of image.
2. **Scaling**

For scaling, we have used Graphics2D class. It resizes the image. For resizing, the current dimensions of the image will be displayed to user, and user will be asked to enter the new dimensions for image.

1. **Rotation:**

8.1 Rotation\_90\_clockwise

8.2 Rotation\_270

8.3 Rotation\_180

8.4 Rotation

For Rotation, we have implemented more than two algorithms. 8.4 works for any angle given by user, but it produces black spaces in image. User usually wants to rotate image at 90 and 180 degree. Therefore, we have also added 8.1, 8.2 and 8.3 which will not produce black spaces in image.