

Image Processing through JavaScript

Vivek Yadav

Abstract: Images are everywhere on the internet, making them an essential content of web page therefore it would be very useful to have the image processing technique within/on the web page itself. In this paper, a novel method is presented to implement the image processing technique at browser's end. The image processing is conducted through JavaScript using various features of Web Component, comprising of Canvas and JavaScript Classes pertaining to image.

Keywords: Image processing implementation on browser

1. Introduction

Images are everywhere, our surrounding, our devices and our mind. Images have become an integral part of our life, life without images is unthinkable. What is an image? It is an artefact that depicts visual representation, a 2 dimensional picture. We are in technology revolution; new things come up on daily basis. Internet is part of this revolution where contents and information are uploaded at large number and accessible to all. Web content constitutes of many things, but most important of them is textual and image representation. Image being an essential content of the web page therefore it would be very useful to have image processing technique on web.

Web application or web site containing the web page can be 2 tier, 3 tier or n tier. If 2 tier then client (browser) and server entities are involved, if 3 or n tier then database comes into picture. One way to implement the image processing is on the server side, the server components can process the images easily. In order to do that the image should be transferred on the server through the network call causing excessive utilization of network and server resources. Another way of implementing the image processing would be on the client side that is browser. In order to process the image on browsers, the technique must be compatible with the browsers. As we all know JavaScript is an integral part of web technology, available on all modern browsers, providing the dynamic functionality to web page. If the technique is developed in JavaScript, then image can be processed efficiently on browsers. This paper expatiates, the technique to use internal JavaScript Classes and Functions to apply the image processing algorithm on an image. Image processing algorithms are mathematical formulas/functions imposed on the pixels, changing the pixel value, within the image.

To work with pixel it is very important to understand what pixel comprises of. Each pixel contains 4 channels, RGB (Red, Green and Blue) are 3 channels while the fourth one is alpha channel. The values in RGB channel range from 0 to 255 whereas value in alpha channel range between 0 to 1. The RGB channels are responsible for colour and alpha channel is responsible for transparency.

2. Image Representation in Webpage

2.1 HTML

The tag contains an image in an HTML page.

2.2 Image (JavaScript Class)

Used for creating the instance of image.

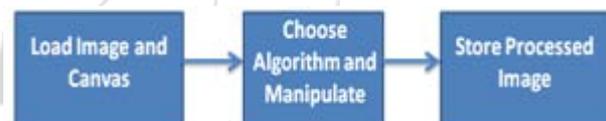
It accepts two optional parameters: Image ([unsigned long width, unsigned long height])

Returns an HTML Image Element instance just as `document.createElement('img')` would.

2.3 HTML5 Canvas (Supported on all the modern browsers)

The HTML <canvas> element is used to draw graphics, on the fly, via scripting (JavaScript). The <canvas> element is only a container for graphics. JavaScript is used to draw the graphics.

3. Method and Implementation



The image processing consists of 3 phases.

3.1 Load Image and Canvas

Select the desired image and store it within tag or as an instance of Image Object (JavaScript Class). In order to store the image as an instance, use FileReader Class.

To play with the image, the image replica must be stored as an instance of Image Object and later imprinted within a <canvas> tag that acts as a container of the modified image. In order to print the image to canvas use 'ctx' and JavaScript functions of Canvas.

```
ctx = canvas.getContext('2d');  
ctx.drawImage(img, 0, 0, canvas.width, canvas.height); //  
img represents Image Object
```

3.2 Choose Algorithm and Manipulate

Choose the desired image processing algorithm to be applied on the image. Define components to perform the image manipulation process compatible with the browser, if you wish to provide control to the user. The image processing formula is applied on the pixels within the image. The pixel value is extracted using the 'ctx' and `Uint8ClampedArray`.

```
var imageData = ctx.getImageData ( 0 , 0 , canvas.width ,
    canvas.height );
```

```
var pixelData = imageData.data
// represents object containing image data
```

```
var uint8pixelData = new Uint8ClampedArray
( pixelData.length )
// array representing 8-bit unsigned integers clamped to 0-
255
```

The 'uint8pixelData' object can be used by conditional statements and 'pixelData' can be used to manipulate image values, but they point to the same address.

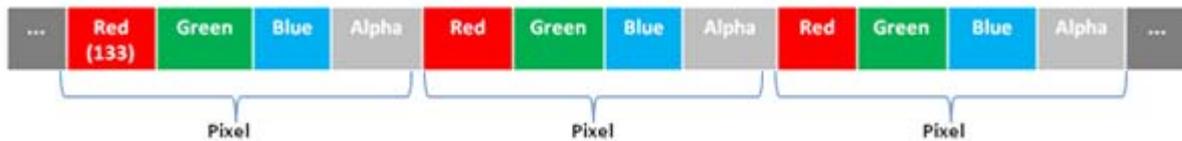
```
if ( uint8pixelData[i+j] > threshold // threshold can be any
    input for eg: 200
```

```
    pixelData[i+j] = 255;
```

```
else
```

```
    pixelData[i+j] = 0;
```

The data within the 'uint8pixelData' array contains the 4 channels in linear fashion as depicted below. This data must be handled properly to get the correct outcome.

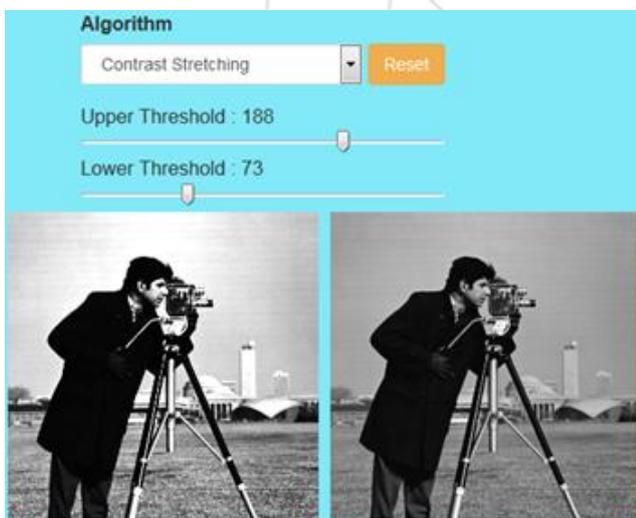


3.3 Store Processed Image

After image processing algorithm has been applied on pixels of the image, the modified pixels must be united and formed as a modified image. It is possible to perform this task using Canvas

```
ctx.putImageData(imageData, 0, 0); // Paint the new image in
    Canvas
```

The implementation of the process is shown in screenshot below. The image on the left hand side is modified image and right hand side image is the original one.



To view the implementation, please visit <http://imagerefining.com>

4. Conclusion

This paper represents the steps to process an image in browser, hence image processing task can be handled on the client side unless there exists some issues like security concerns.

Using the same process, other tasks related to image can be accomplished easily like sending image over the network, copying image data, manipulating the image data to send messages, etc.

References

- [1] Book to understand algorithms: Image Processing Using MATLAB Codes by Dhananjay K. Theckedath
- [2] <https://en.wikipedia.org>
- [3] <https://developer.mozilla.org/en-US/docs/Web>
- [4] <http://www.w3schools.com>