1. About

Vital Pixels

Reference Guide v 4.6.7

Website: http://www.vitalpixels.com

Mohsen Shakibafar

ABSTRACT

Vital Pixels is the Integrated Development Environment (IDE) for developing and analyzing image processing algorithms. This software has the following tools and features to edit, debug, processing and analyzing algorithms.

1. You can open and edit your sample images
2. You can use simple tools for general editing such as crop, filters, changing colors balance, brightness and gamma, etc. …
3. You can view and save images and algorithm's output information in a property grid.
4. You can use zoom in and out of your images
5. You can mix images with many method such as XOR, OR, And, etc. …
6. You can use statistics tools for analyzing your algorithms.
7. This software calculates mathematical parameters for your resulted image, such as SNR and PSNR after each run.
8. You can visit and save histogram of input/output images.
9. You can use external resources and libraries such as your programs generated by Visual Studio.
10. You can build your own plugin and send it to another program.
11. You can use any free third party plugins.
12. You can analysis color and brightness.
13. You can use strong editor and debugging tools for editing your algorithms.
14. You can use the MATLAB code and programming in your algorithms.
15. This software saves result of your algorithms in xml so you can analysis result of your algorithms with other programs such as excel and access.

CONTENT

١ About 1.

٢ ..... ABSTRACT

٣ ..... CONTENT

Install and Download 2.
٥

٥ ..... System Requirement

٥ ..... Install

٨ ..... How to Update Vital Pixels

٩..... Starting First Program 3.

٩ ..... Writing first program

١١..... Working with Images 4.

١١..... Open an Image

١٢..... Capture Screen

١٣..... Save Edited Image

١٣..... Image Properties

١٣..... Print Input and Output image

١٥ Image Menu 5.

١٥..... Image Mode

١٥..... Invert Color

١٦..... Make Grey style

١٦..... Colors to Red

١٦..... Colors to Green

١٦..... Colors to Blue

١٧..... Change Brightness

١٨..... Change Contrast

١٨..... Change Gamma

١٩..... Flip Image

١٩..... Commit Image

١٩..... Histogram Window

٢١..... Set Result as a Source

٢١..... Mix Images

٢٢ Filter Menu 6.

٢٢ ..... Blue Filter
٢٣ ..... Find Edge Filter
٢٣ ..... Add Noise Filter
٢٤ ..... Mosaic Filter
٢٥ ..... Programming in Vital Pixels 7.
٢٥ ..... SemiCode Programing
٢٧ ..... FullCode Programming
٢٩ ..... Run and Compile
٢٩ ..... Using MATLAB
٣١ ..... Import another library
٣١ Plugin 8.
٣١ ..... Install and Uninstall Plugin
٣٢ ..... Build a Plugin

2. Install and Download

Users can download Vital Pixels and install it competently free. Also this software has some facility to update automatically from internet.

System Requirement

Component	Minimum Requirement
Processor	4 cores for small deployments
Ram	4GB
Hard disk	1GB free

Table 2-1 . Hardware requirement

Component	Requirement
OS	Windows 7, 32-bit Windows 7, 64-bit Windows 8 Windows 10 (It is not compatible with Windows xp)
.NET Framework	Microsoft .NET Framework 4.0 (Free install)
MATLAB	Optional (It is necessary if you want to import script from MATLAB)
Library	Vital Pixels is based on Microsoft GDI+ library and it has supported in any version of windows

Table 2-2. Software Requirement

Install

Vital Pixels is a free software and you can download it from http://www.vitalpixels.com/download and install it. For installing you should do following step as you can see in figures.

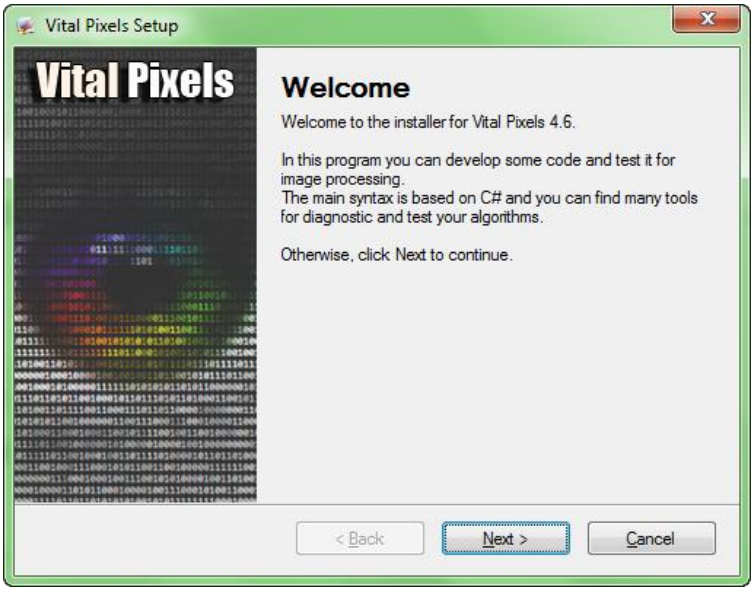

Figure 2-1. First step after for install

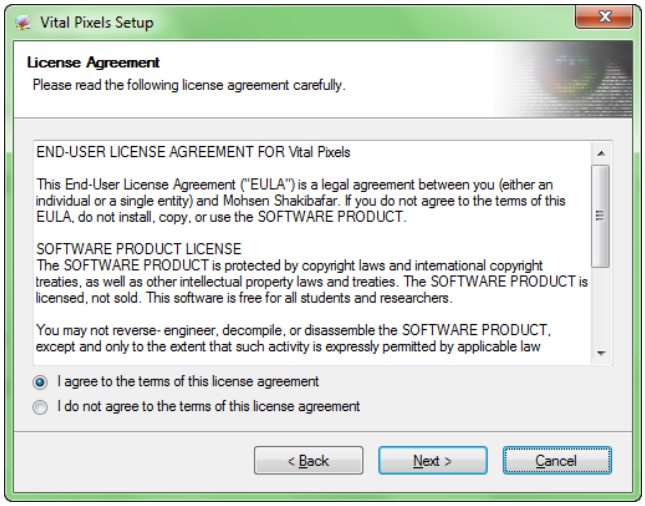

Figure 2-2. You should accept out license agreement in install

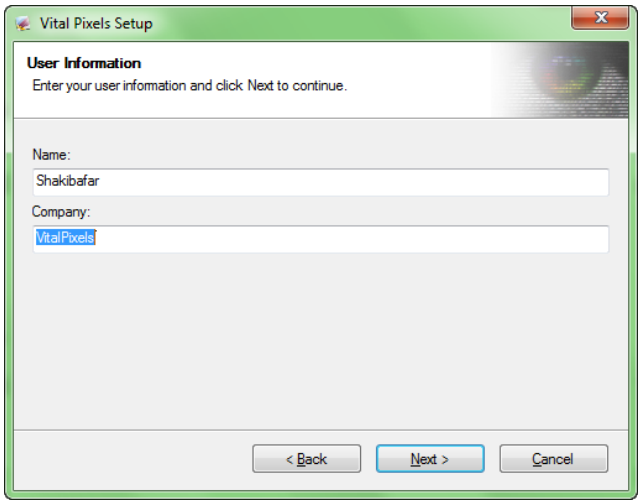

Figure 2-3. You can choose your name and company name in install

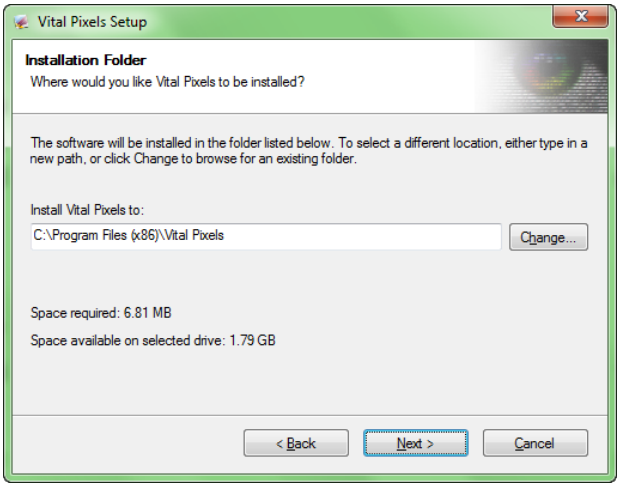

Figure 2-4. You should select installation folder (it needs less than 10MB)

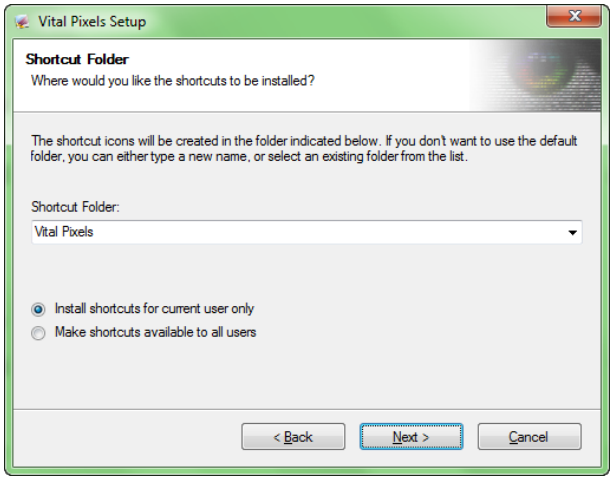

Figure 2-5. You can type shortcut name

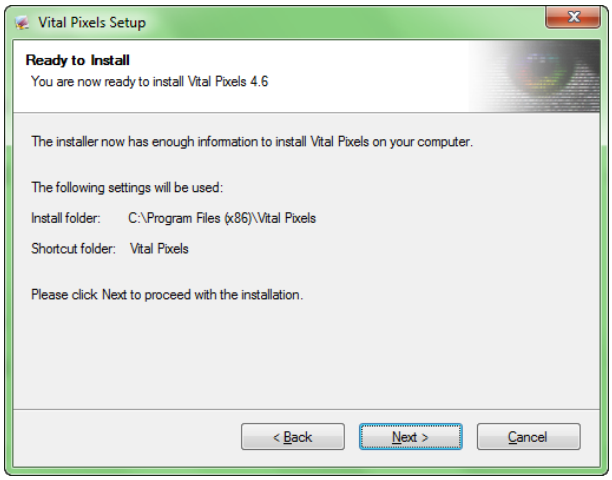

Figure 2-6. For start installing press next button

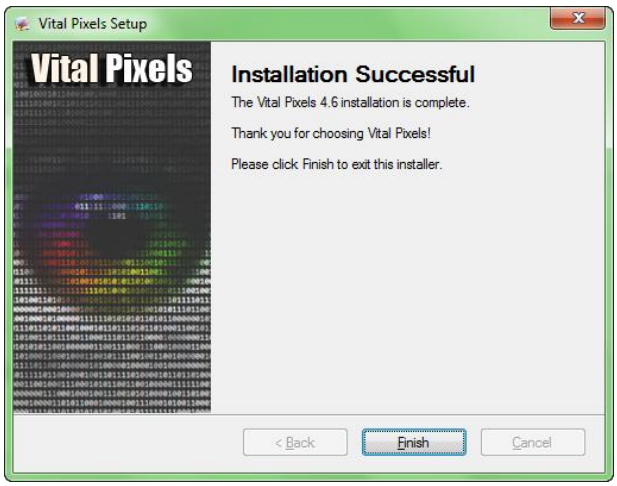

Figure 2-7. This window show installation was successful

How to Update Vital Pixels

If Vital Pixels installed successfully, you can see two expectable program. One of them is main program and another one is UpdateVP (Fig 1-1). UpdateVP program is designed for checking and updating Vital Pixels to latest version.

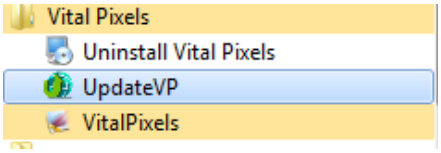

Figure 2-8. UpdateVP program for updating Vital Pixels in start menu

After running UpdateVP, you can see Update Windows that shows in Fig 1-2.

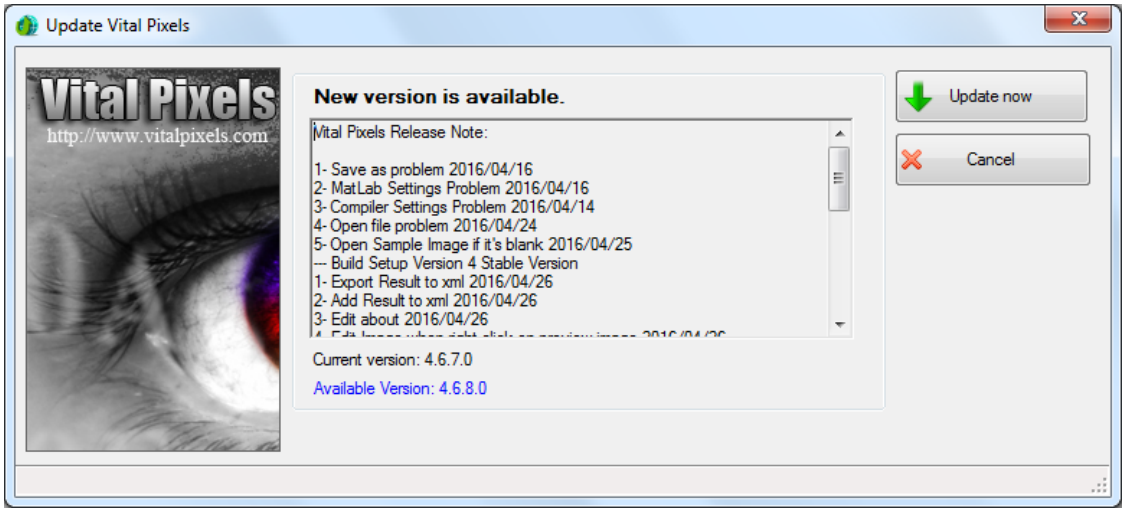

Figure 2-9. Update window

As you can see in Fig 1-2, UpdateVP check latest version and if any new update was available, it would show release note and “Update Now” button would be enable. If you click on “Update Now”, download would be started and automatically install would start.

In Vital Pixels program you have shortcut to access UpdateVP program. As you can see in Fig 1-3 after clicking on “Help->Update “, software run UpdateVP.

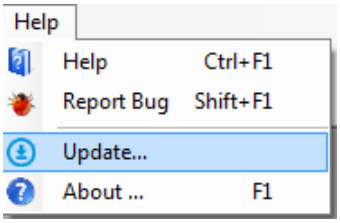

Figure 2-10. Run UpdateVP from Vital Pixels

Vital Pixels after any run, can check new version. But if you want to enable this ability you should go to “Tools->Preference” and check “Check for update”. As you can see in Fig 1-4 it is disable, by default.

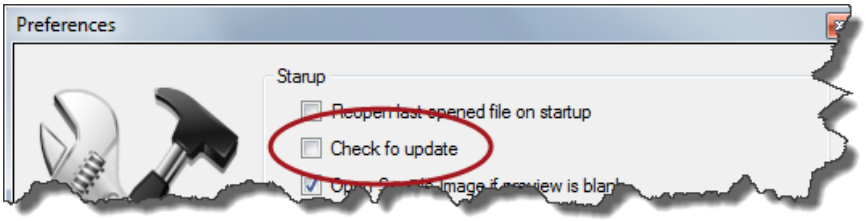

Figure 2-11. You can set Vital Pixels to check new version in any run

Please notice that this update system doesn't support any proxy settings so it connect to internet directly.

3. Starting First Program

You can see main window in Fig 2-1. As you can see its GUI is very friendly and you can start your algorithms very fast and easily.

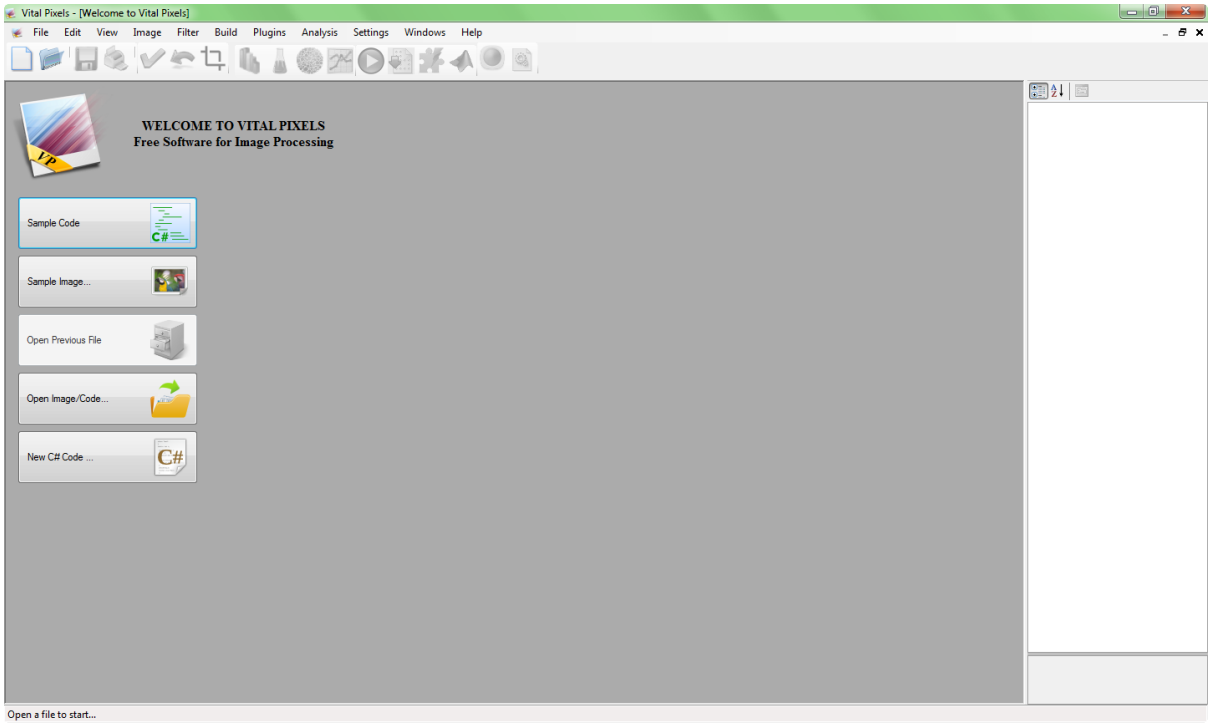

Figure 3-1. Main window of Vital Pixels

Main window can divided in 3 parts:

1. Main Document window
2. Property grid and Preview Image
3. Menus and toolbars

Writing first program

For starting first program you should click on "Sample Code" button. After that software will open a very sample code and sample image for starting. For more information about windows you can see following figures.

```
#region a

for (int y = 0; y < OutputImage.Height; y++)
{
    for (int x = 0; x < OutputImage.Width; x++)

    {

        Color c = ((Bitmap) InputImage).GetPixel(x, y);

         int luma = ((c.R)+(c.B)+(c.G))/3;

         //OutputImage.SetPixel(x, y, Color.FromArgb(c.G,c.R,c.B));
         OutputImage.SetPixel(x, y, Color.FromArgb(luma,luma,luma));
    }
}




argMessage = "Output Message";

#endregion
```

Figure 3-2 . Editor window for editing and debugging your codes

ColorsInfo	
NumberofColors	86363
FileInfo	
FileName	**Sample.jpg**
Size	84498
ImageInfo	
Height	**512**
TotalPixels	262144
Width	**512**
Output	
argMessage	Output Message
Resolution	
HorizontalReso	72
VerticalResoluti	72
Result	
ElapsedTime	725 ms
OutputColor	250
PSNR	25.15 dB
SNR	19.19 dB

Figure 3-3. Property grid of image and result

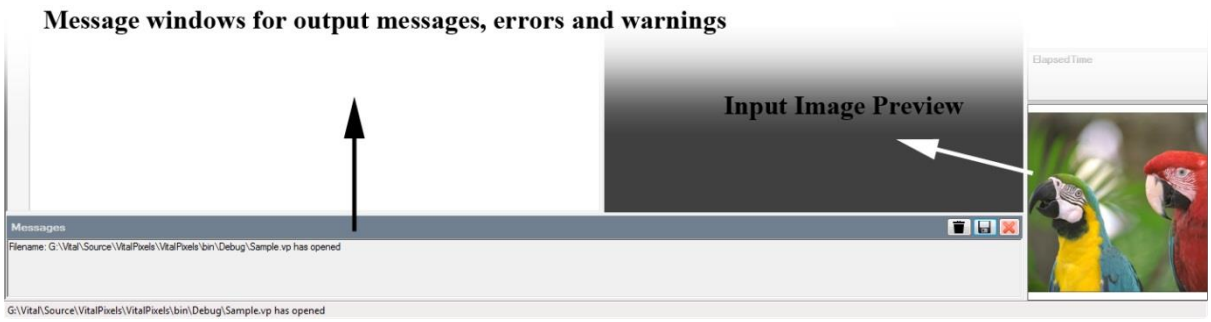

Figure 3-4 . output window and preview image

After editing your program you can build and compile your program with "Build->Compile" menu and you can study your errors in "Messages Windows". If you have no errors you can "Run" your program with "Build-> Run". As you can see in Fig 2-5. After running you can see result of your program in output windows.

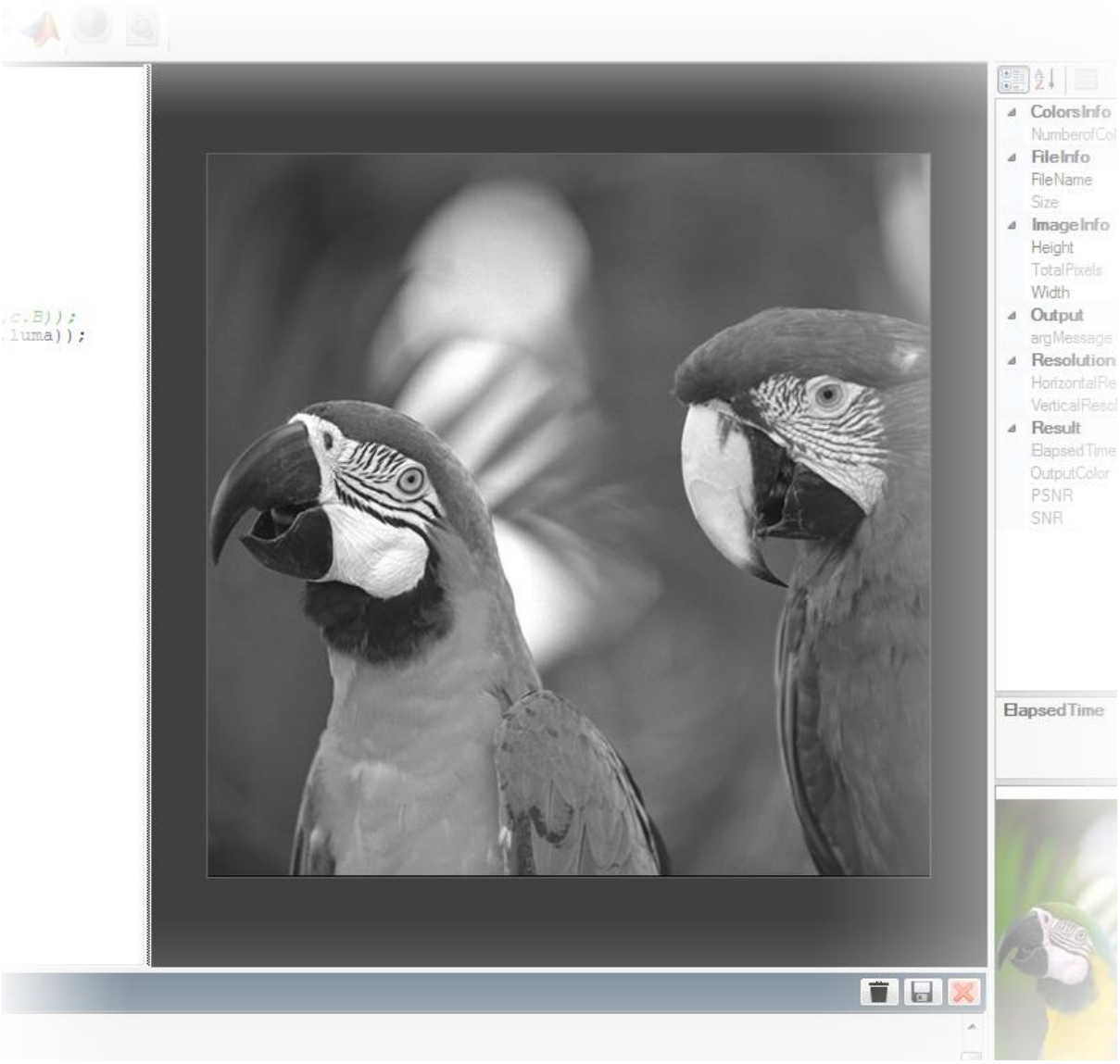

Figure 3-5. Output image after runnging program

4. Working with Images

Vital Pixels has some ability to open and editing images for testing in your algorithm.

Open an Image

You have three ways to open an image for your work. First one in using "File->Open" menu. After that you can use every image with any extension and use it for edit and running algorithms. Best property for images in algorithm is shown on table 3-1.

Property	Value
Width	512 pixels
Height	512 pixels
Resolution	96 dpi
Other	Without alpha and animation

Table 4-1 . Best Property of images

As you know in Image Processing we have some standard image for testing the algorithms. If you use Vital Pixels, it doesn't need to search those images on internet because Vital Pixels provide standard and test images for users. For using them you can use "File-> Open Sample Image". After using this menu Fig 3-1 will be shown.

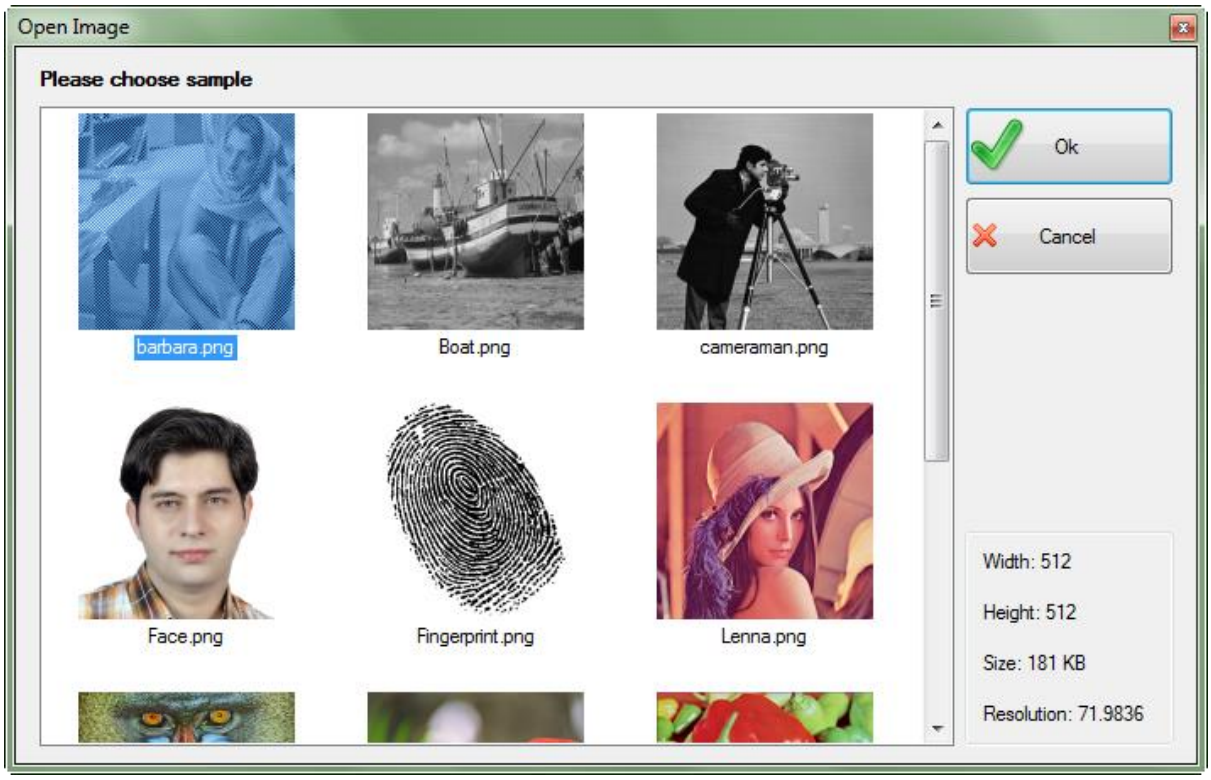

Figure 4-1 . Open Sample Image

On another hand you can use another method for opening images. If you right click on "Preview Image" you will see change image (see fig 3-2). After clicking on this item you can change your source image.

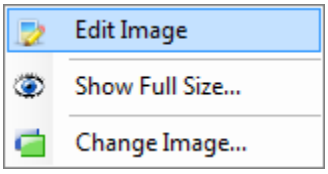

Figure 4-2 . Change Source Image

Capture Screen

If you want to work on desktop images you can capture your screen and use it for your program. For

using this ability you should use "File->Captures Screen" menu. After click, you should save your screenshot and it will be opened in software.

Save Edited Image

If you use editing tools for image in Vital Pixels you can save it and use it again. Such as another program we have "Save", "Save as" and "Save all" tools for saving your edited images.

Image Properties

In image processing you need to know about your input images, such as height, width resolution and etc. In Vital Pixels you can see all of information about your input image in a property grid (Fig 2-3).

This table is divided into six parts (Table 3-2):

Title	Description
ColorInfo	It shows number of all colors in input image
It shows	It shows file name of input image and its size in bytes.
ImageInfo	It shows width and height of input image
Output	In program you can define a output parameters and calculate it. This item shows value of your output parameter.
Resolution	It shows vertical and horizontal resolution of input image in DPI.
Result	It show result of program such as Elapsed time , Number of color in output image, PSNR and SNR in output image.

Table 4-2 . Content of property grid

You can save information in property grid with "File->Save Result as XML" menu. This item exports all data as a XML file and you can use it in Excel program and another database. With another item (File->Add Result to XML) in File menu you can add information of property grid in existed file and new data will be added to previous data. This ability is very useful for Excel. For example in fig 3-3 we show imported data in Excel.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Filename	Size	Width	Height	TotalPixels	HorizontalResolution	VerticalResolution	NumberofColo	SNR	PSNR	OutputColor	ElapsedTime	MyProperty
2	peppers.png	538749	512	512	262144	1199.998	1199.998	183525	18.49 dB	24.24 dB	225	1903 ms	null
3	Parrot.png	390758	512	512	262144	72.009	72.009	60558	19.18 dB	25.14 dB	244	1611 ms	null
4	Lenna.png	473831	512	512	262144	96	96	148279	25.4 dB	31.08 dB	200	1508 ms	null
5													
6													
7													
8													
9													

Figure 4-3 . Imported xml data from poperty grid

Print Input and Output image

Such as another standard program you can print you document in Vital Pixels. When an image is active you can print both input and output result with "File->Print". Fig 3-4 shows a sample of document that was printed in Vital Pixels when an image was open and active.

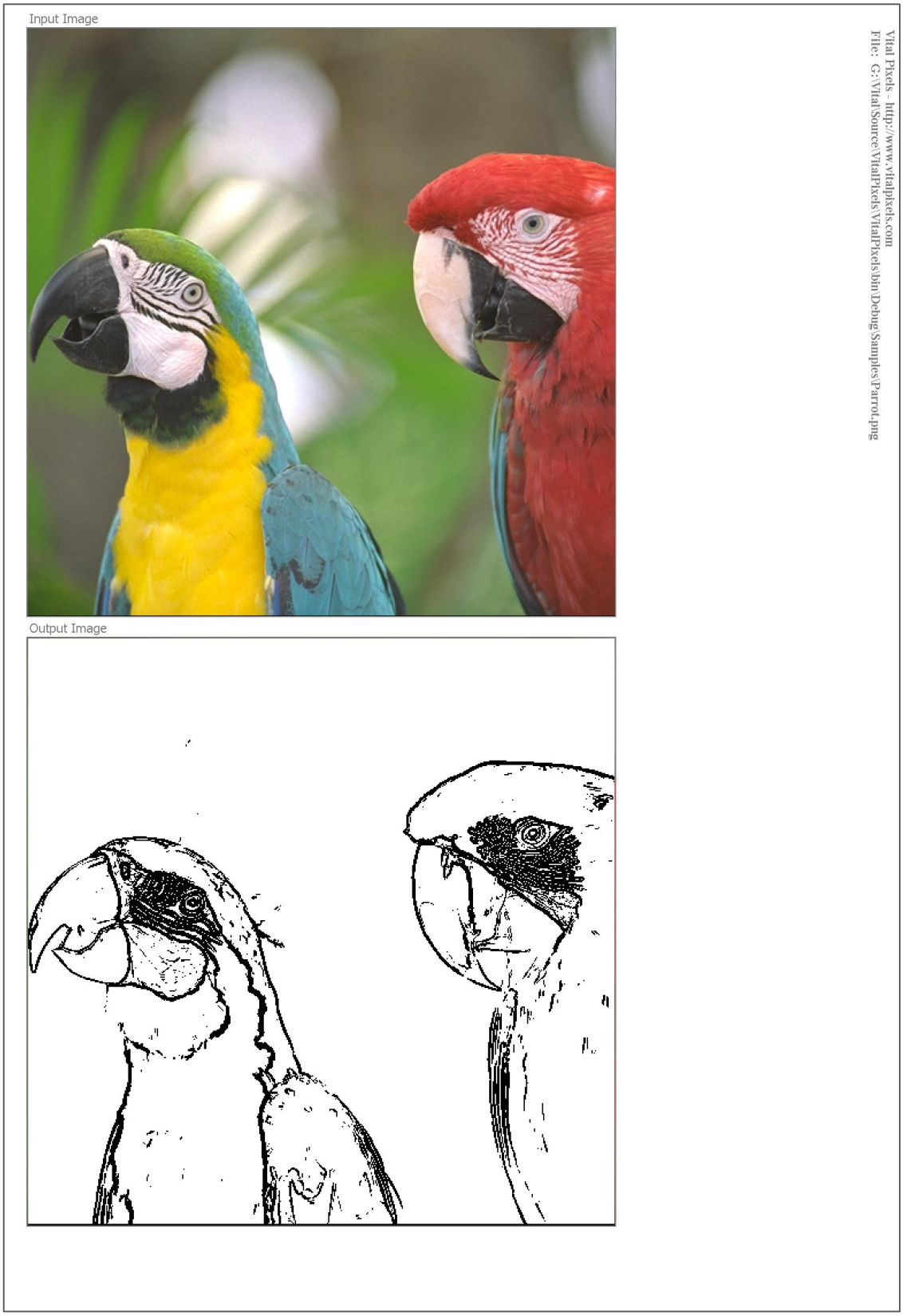

Figure 4-4. Sample of print

5. Image Menu

In Vital Pixels you can work on your input image to change its features for testing your algorithm in many conditions. Menu item is one of the ability to change image properties.

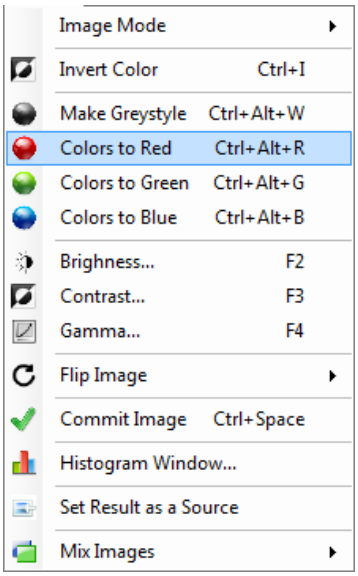

Figure 5-1. Image menu

Image Mode

With this menu you can change mode of image in 4 modes. As you can see in fig 4-2 this menu can reduce color and size of image in 4 modes.

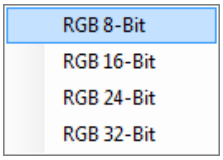

Figure 5-2 . Image -> Image Mode

Invert Color

With "Image->Invert Color" you can make you image negative and change all of colors to inverted one. For example you can its result in Fig 4-3.

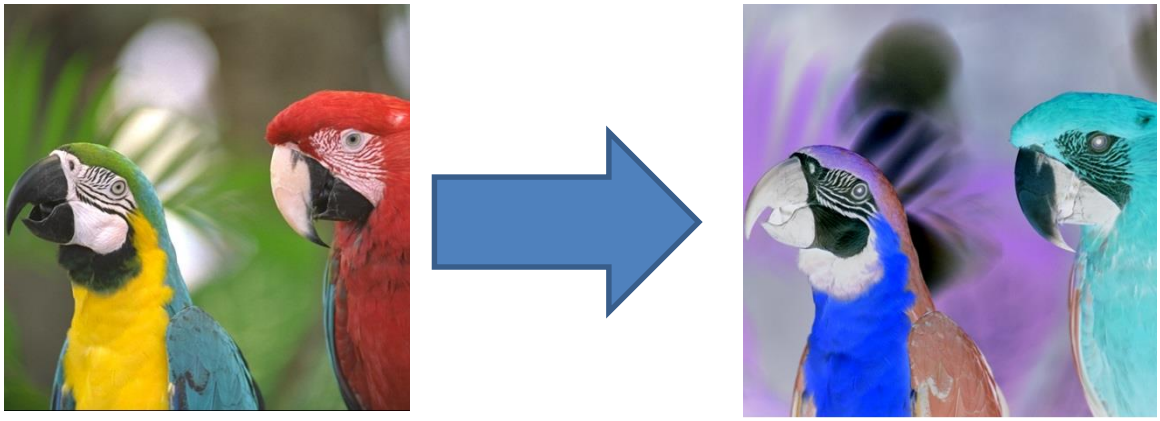

Figure 5-3. Image->Invert Color

Make Grey style

With "Image->Make Greystyle" you can convert your colorful image to a black and white image. Fig 4-4 shows a sample result of this menu.

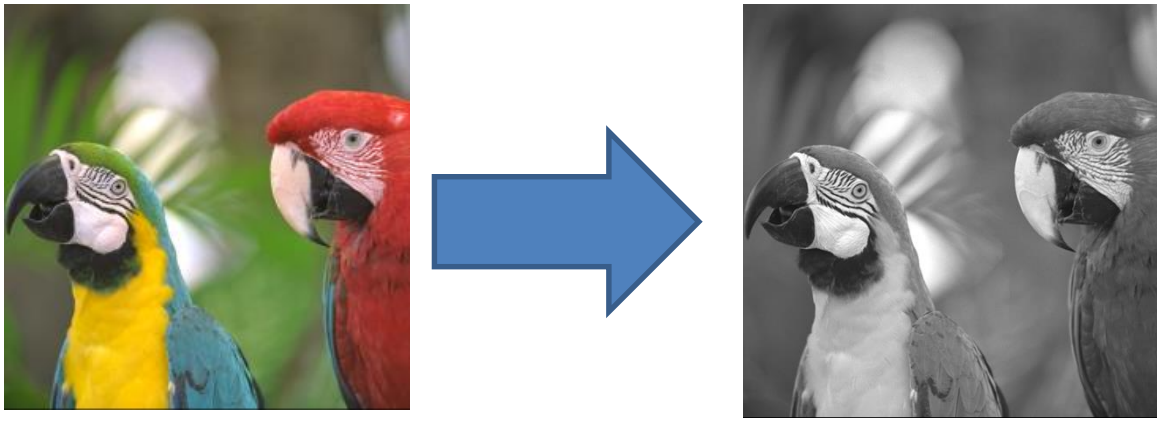

Figure 5-4. Image-> Make Grey Style

Colors to Red

With "Image->Colors to Red" you can convert all colors to red style. In this action all of other colors (Green and Blue) will be converted to zero (Fig 4-5).

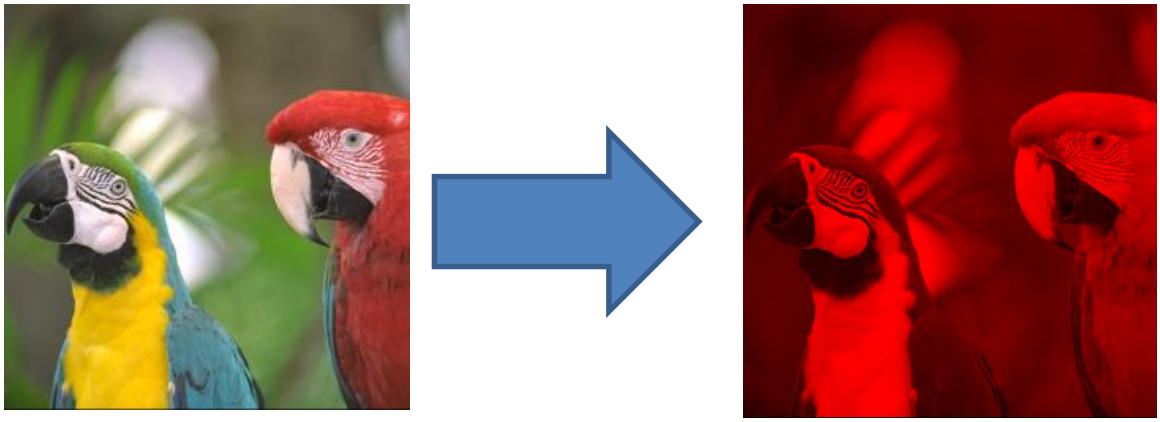

Figure 5-5 . Image-> Colors to Red

Colors to Green

With "Image->Colors to Green" you can convert all colors to red style. In this action all of other colors (Red and Blue) will be converted to zero (Fig 4-6).

Colors to Blue

With "Image->Colors to Blue" you can convert all colors to red style. In this action all of other colors (Red and Green) will be converted to zero (Fig 4-7).

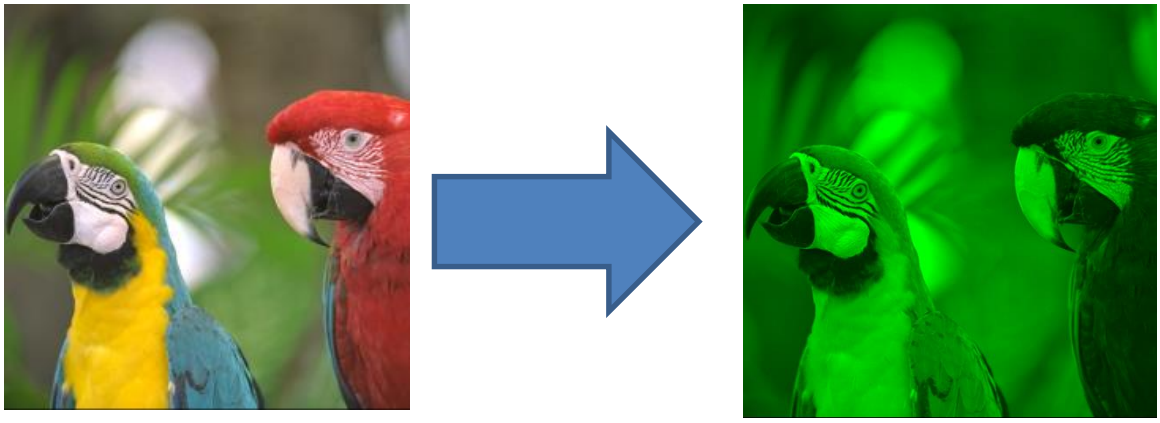

Figure 5-6 . Image-> Colors to Green

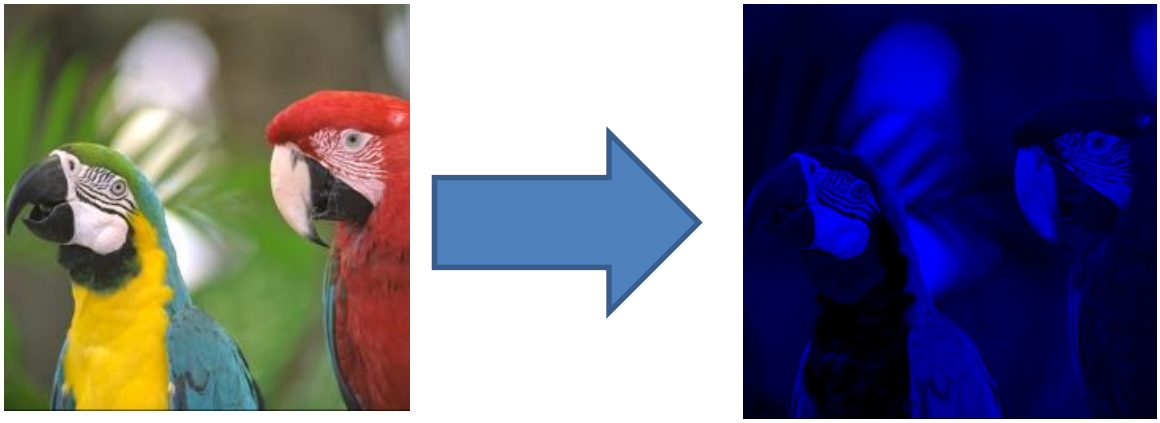

Figure 5-7. Image-> Colors to Blue

Change Brightness

With "Image->Change Brightness", you can change brightness of input image. As you can see in Fig 4-8, Vital Pixels shows a windows to configure brightness in image.

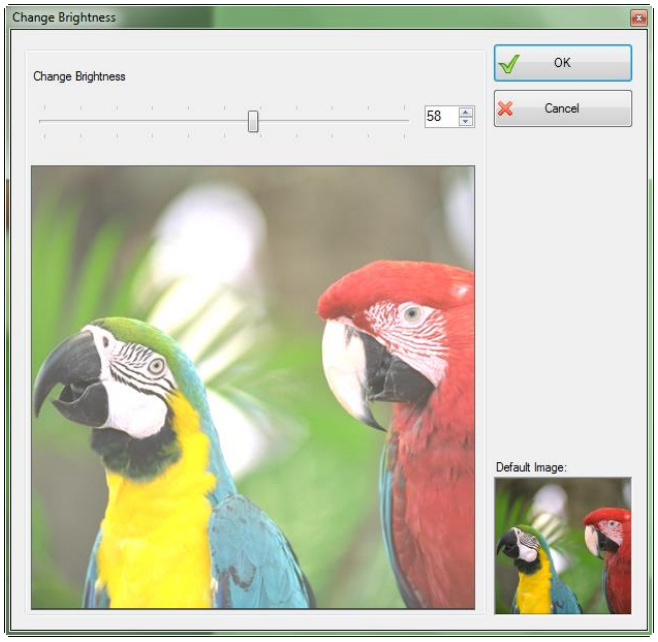

Figure 5-8. Imgae -> Chnage Brighness

Change Contrast

If you want to test your program with another contrast you can change it in "Image->Change Contrasts". After click on this menu a window like Fig 4-9 will be appeared and you can change contrast.

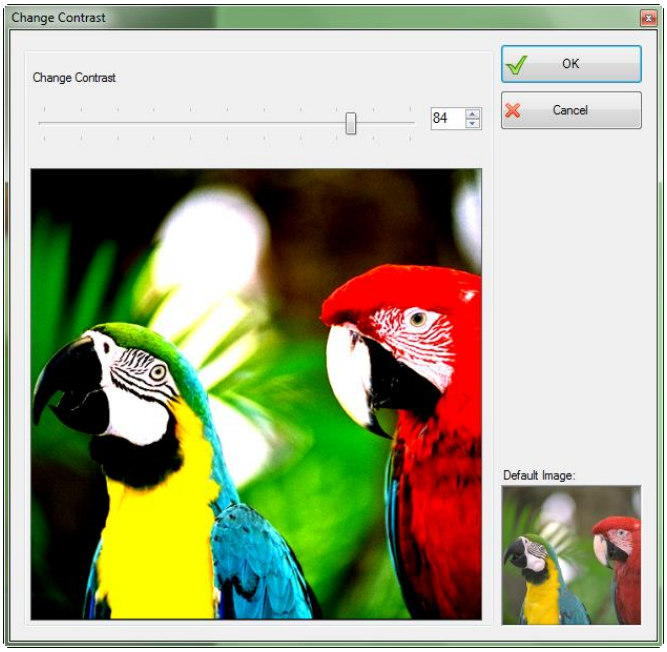

Figure 5-9 . Image->Change Contrast

Change Gamma

With "Image->Change Gamma" and its window (Fig 4-10) you can change Red, Green and Blue gamma.

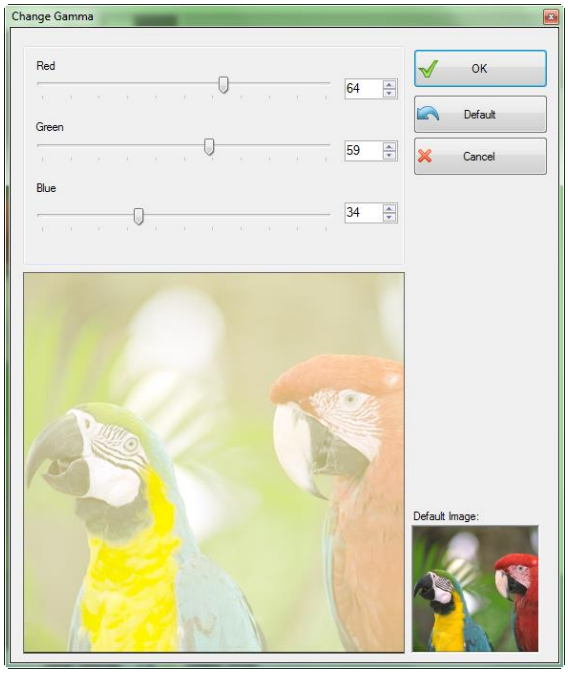

Figure 5-10 . Image-> Change Gamma

Flip Image

If you want to rotate your source image you can use this menu. This menu has 4 sub menu that you can see in Fig 4-11. As you can guess, this item rotate image in 4 degrees and directions.

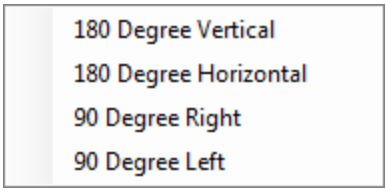

Figure 5-11 . Imgae->Flip Image

Commit Image

If you do any change in source image, its border will be red (Fig 4-13). In this situation can undo your change with "Reset to Default" button in toolbar (Fig 4-12). After commit you couldn't commit you change and its red border will be disappeared.

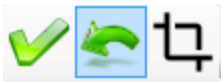

Figure 5-12. Reset to Default

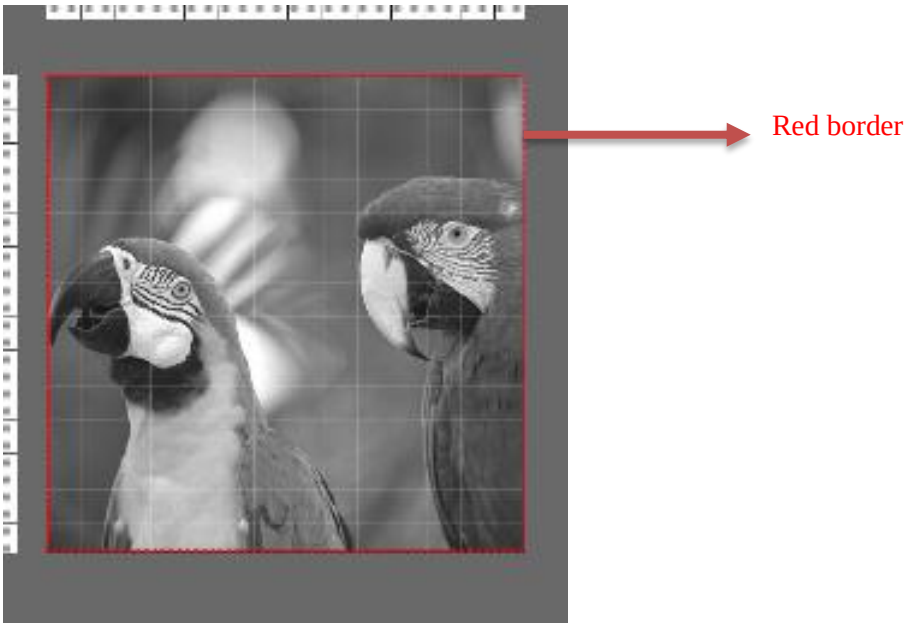

Figure 5-13 . Red border shows that you can undo chnages.

After commit your change couldn't be restore.

Histogram Window

Histogram is so important in Image Processing and Vital Pixels has a powerful tools to show, save and analysis histograms. You have two ways to see histogram. First one is "Image->Histogram Windows". After click on this menu you will see a window (Fig 4-14).

In this window you can see histogram of input and output image in the same time. In this window you can see 4 types of histogram (Red, Green, Blue and Grey).

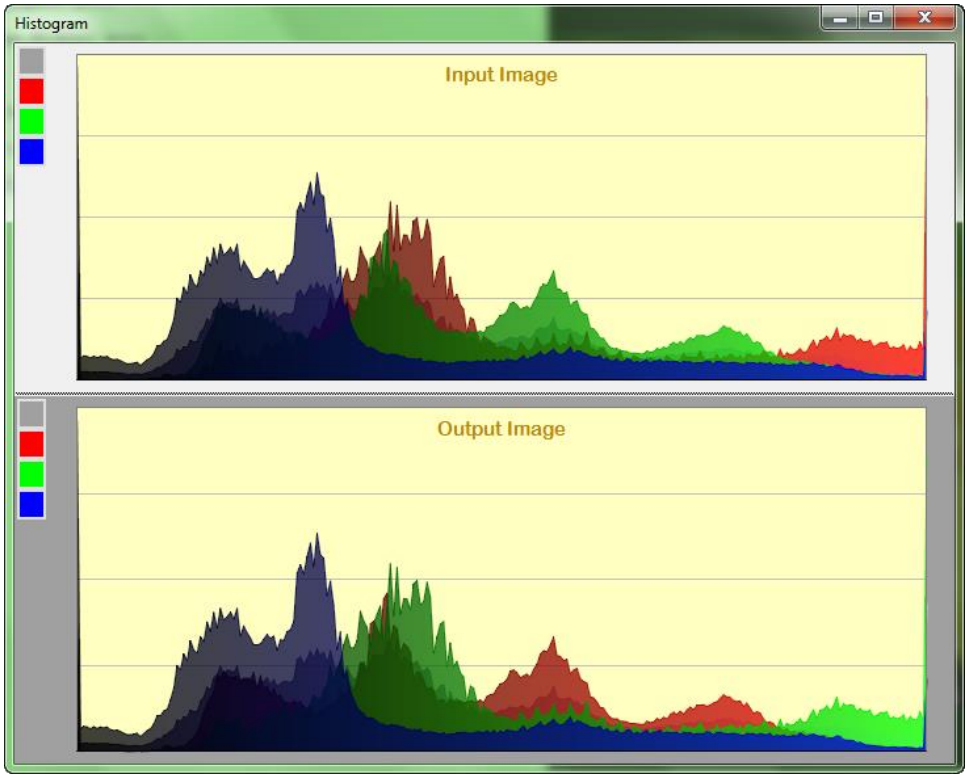

Figure 5-14 . Histogram of input and output image

On the left of window you see 4 buttons that with click on those you can visible and invisible any type of histogram.

For more action you can right click on any histogram (Fig 4-15). In this menu you can save histogram as an image or print it and you can change its background color.

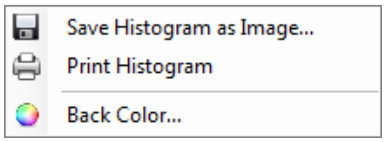

Figure 5-15 . Menu in histogram after right click

If you need to see histogram in any change and any rung immediately, you can use immediately windows on the button of any image. As you can see in Fig 4-16 you can choose histogram window on the bottom of input and output image. By default, immediately window is set to “Zoom” window.

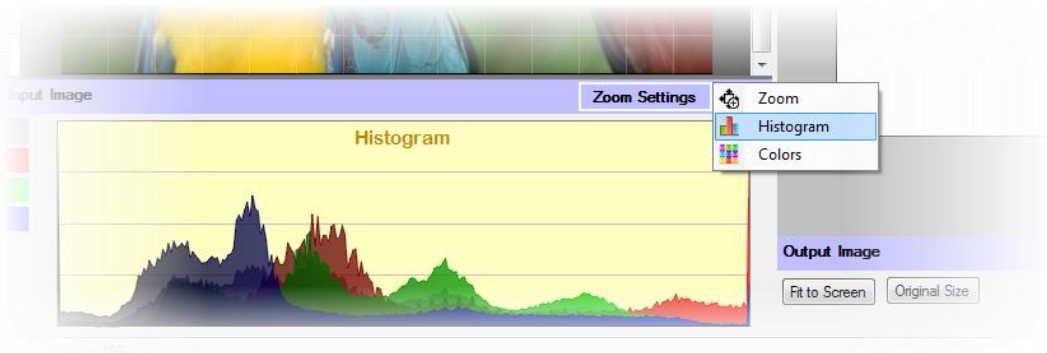

Figure 5-16. Histogram in immediately window

Set Result as a Source

With this menu (Image->Set Result as a source), you can move result of program as a source for another program or plugin.

Mix Images

In this menu you can mix some images in input image (Fig 4-17). As you can see in Fig.25 we have many calculation to mix two images, such as average, add, sub, OR, XOR, Max, Mix, etc.

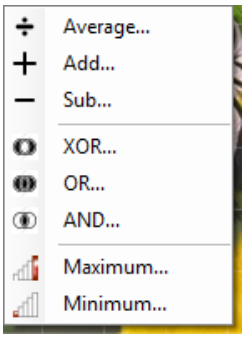

Figure 5-17. Mix Images

As you know in any item colors of any pixels will be mix to its corresponding. For example if you choose "XOR" any software will mix colors like this:

```
For any x,y in Image1

        Image1.Color(x,y).R= Image1.Color(x,y).R XOR Image2.Color(x,y).R

        Image1.Color(x,y).G= Image1.Color(x,y).G XOR Image2.Color(x,y).G

        Image1.Color(x,y).B= Image1.Color(x,y).B XOR Image2.Color(x,y).B
```

Fig 4-18 shows the result of mixing two images.

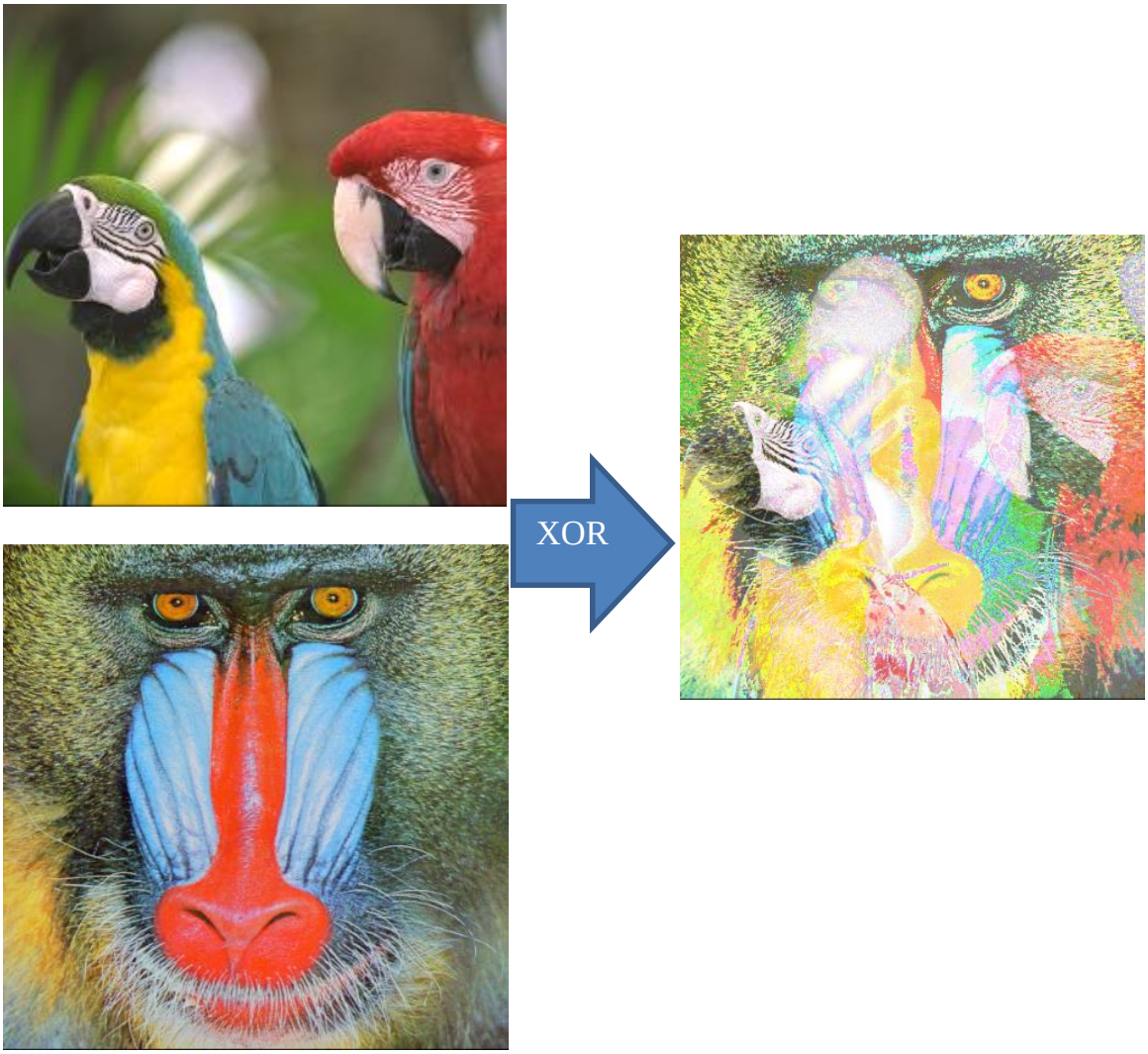

Figure 5-18. Image->Mix Images->XOR

6. Filter Menu

As you can see in Fig 5-1, Vital Pixels has some basic filter for images. This filters will be applied in input image.

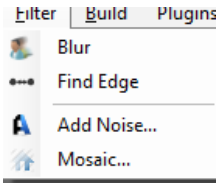

Figure 6-1. Filter menu

Blue Filter

If you only want to blur the image a little bit — to soften it, as it were — you might use the simple "Blur" filter. This filter runs automatically, without creating a dialog. The effect is subtle enough that you might not even notice it, but you can get a stronger effect by repeating it. In Vital Pixels you can

repeat filter to do it stronger (Fig 5-2).

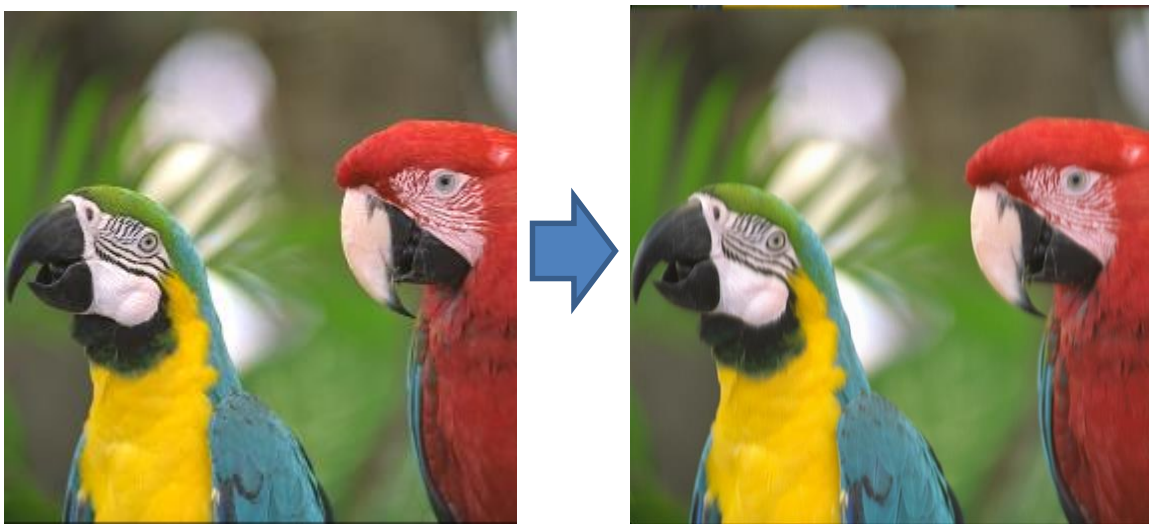

Figure 6-2. Blur Filter

Find Edge Filter

Edge detect filters search for borders between different colors and so can detect contours of objects. They are used to make selections and for many artistic purpose. Fig 5-3 shows the operation of this filter.

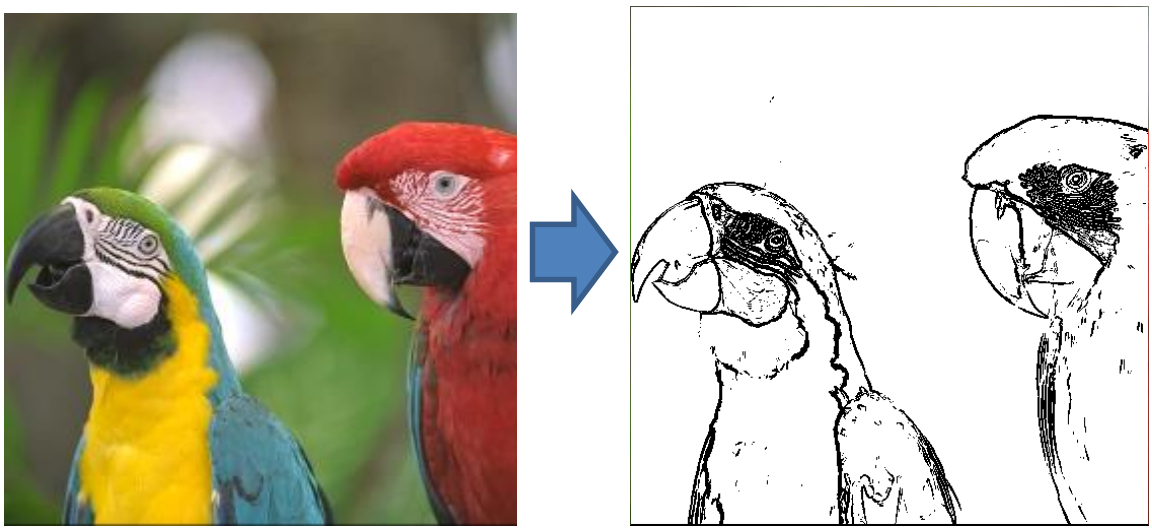

Figure 6-3. Find edge filter

Add Noise Filter

After click on this menu you can see its windows (Fig 5-4) and with this you can add colorful noise to active image.

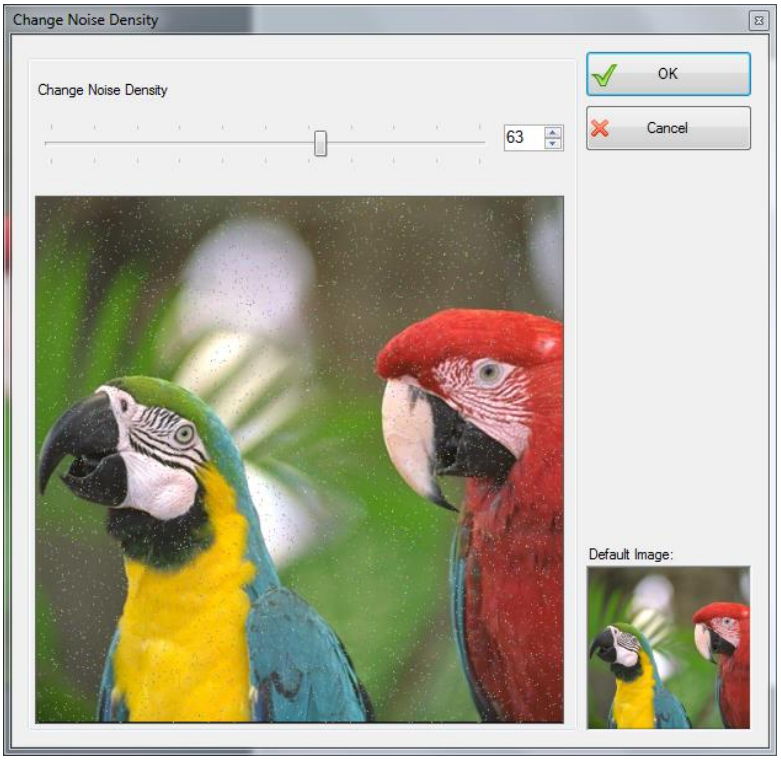

Figure 6-4. Noise Filter

Mosaic Filter

It cuts the active image into many squares or polygons which are slightly raised and separated by joins, giving so an aspect of mosaic. As you can see in Fig 5-5, you can select size of mosaic in this filter.

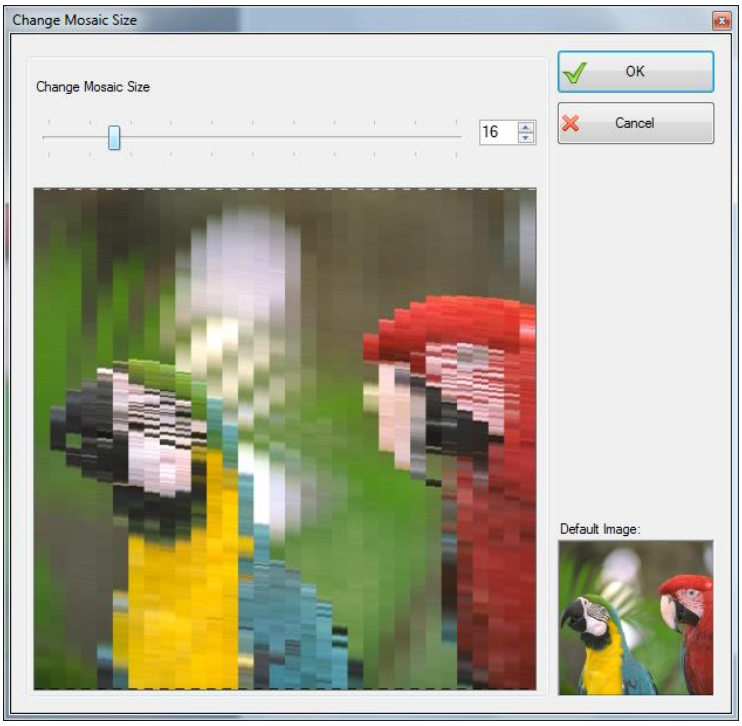

Figure 6-5. Mosaic Filter

7. Programming in Vital Pixels

In vital Pixels you can write you algorithm in C#. You have two method to write program. These are SemiCode and FullCode. At first you should know about structure of programming in Vital Pixels. In Vital Pixels we have two Main Image. One of them is Input (Source) Image that you open it from "File" menu. Second one is output image. Output image is the result of algorithm.

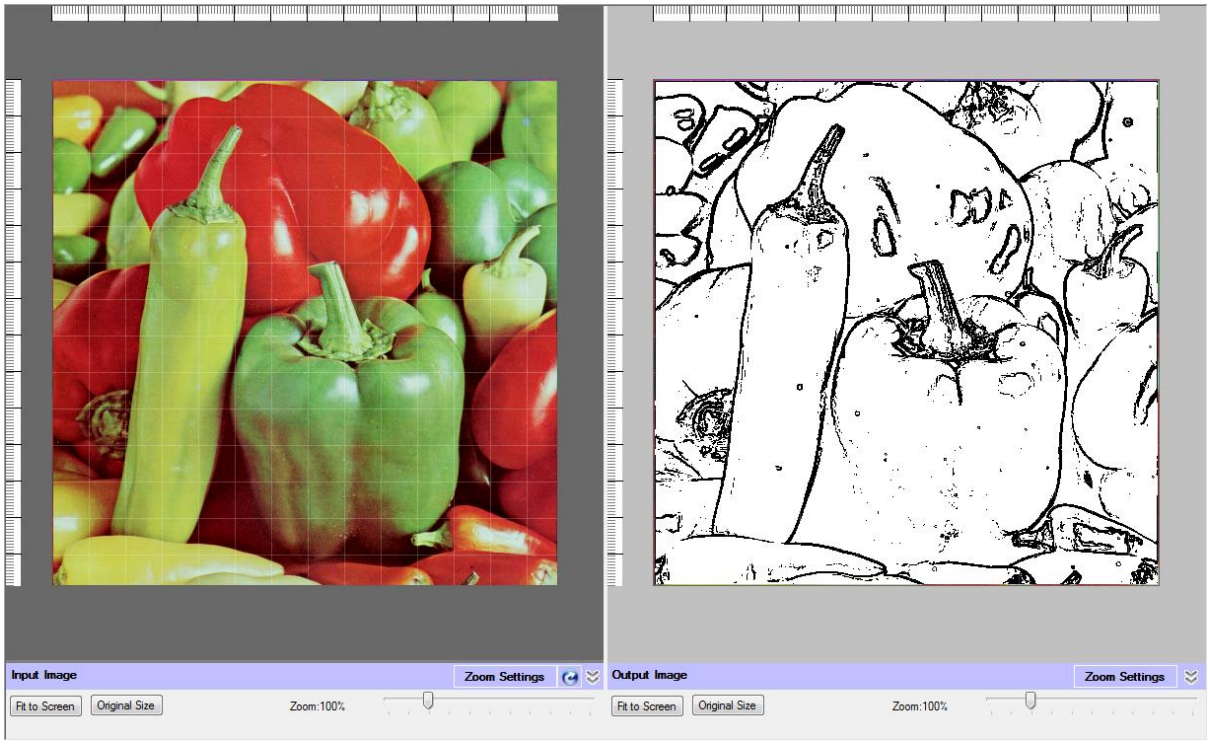

Figure 7-1. Input and Output Image in Vital Pixels

As you can see in Fig 7-2, your program will process in output image then will build output image. After each run software will compare input and output and calculate SNR, PSNR and shows it in property grid.

Figure 7-2. How your program works in Vital Pixels

SemiCode Programing

In SemiCode method you don't need to know every thinks about C# programming such as its structures, namespaces and classes. Following code shows you basic form of SemiCode. After running this program make your input image to grey style.

```
for (int y = 0; y < InputImage.Height; y++)
{
    for (int x = 0; x < InputImage.Width; x++)

    {

        Color c = ((Bitmap) InputImage).GetPixel(x, y);

         int luma = ((c.R)+(c.B)+(c.G))/3;

         OutputImage.SetPixel(x, y, Color.FromArgb(luma, luma, luma));
    }
}

argMessage = "Output Message";
```

In SemiCode you have three basic parameter:

1- InputImage as Image (Image is a type in C#). Input image is implemented in InputImage and with this parameter you have access to any pixels of input image. You can use GetPixel(x,y) to get Color of any pixels. For example:
`Color c = ((Bitmap) InputImage).GetPixel(x, y);`

2- OutputImage as Image. Every pixels that you set in OutputImage will be implemented in output image. You can use SetPixel(x,y, Color) to set color of output image in (x,y).
`OutputImage.SetPixel(x, y, Color.Red);`

3- argMessage as String; If you want to calculate or send a message to user you can use this parameter and you should set a value for it. If you set a value for argMessage, Vital Pixels will should its value in a Message Box and in property grid.

Other parameters and syntaxes are same as C#. In SemiCode you cannot define methods and classes. Because it is not real environment for C#. If you need to define another class or method you can develop it in Visual Studio and import it into Vital Pixels as a reference or you should use FullCode.

For starting SemiCode you can use "Sample Code" button on the "Welcome Window" or click "File->New" and choose SemiCode (Fig 7-3).

SemiCode will be save with "vp" extension.

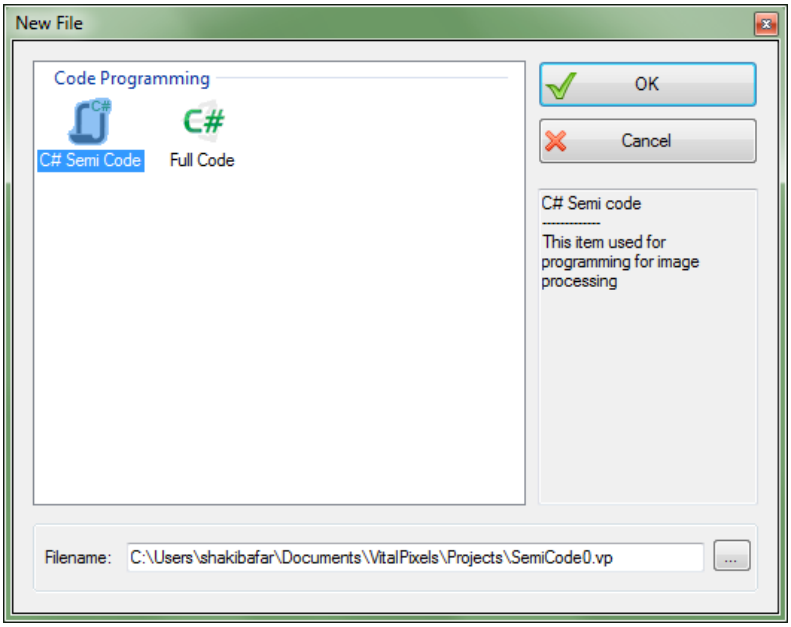

Figure 7-3. Create new C# SemiCode

FullCode Programming

In FullCode programming you can write your program same as C#. But following rules must be done.

1- Main namespace is VPPlugin
2- Main class MyPlugin
3- Main method must be defined such as following line:

```
public Image OutPlugin(Image argImage,out string argMessage)
```

If you use more argument as double, Vital Pixels will define those as input parameters and you can use those for analyzing and flexibility. For example if you define OutPlugin as following line:

```
public Image OutPlugin(Image argImage,out string argMessage ,
double alpha)
```

After running as a plugin, Vital Pixels ask the value of Alpha parameter (Fig 7-4).

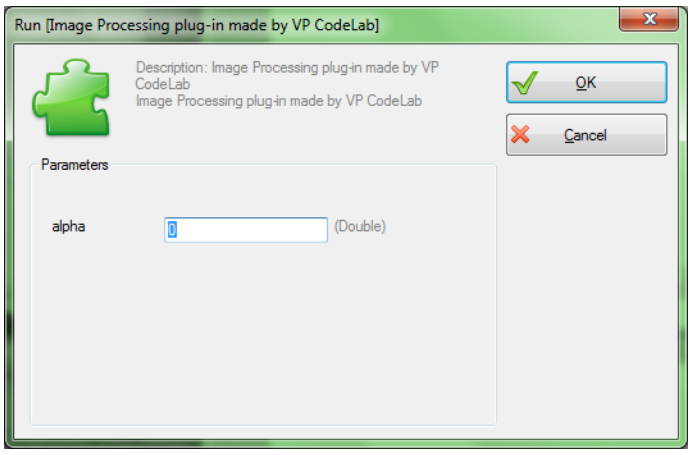

Figure 7-4. Run plugin with more parameters

4- If you return any Image in OutPlugin, Vital Pixels will show it as output image and calculate

necessary parameters or it.

5- In FullCode you can define another method and class.

6- In FullCode you should define all of necessary namespaces with "using" keyword. Some namespaces is necessary for any Vital Pixels program.
They are:
using System.Reflection;
using System.Drawing;
using System.Drawing.Imaging;
using System.Drawing.Drawing2D;
using System;
using System.Collections.Generic;
using System.Text;
using System.Threading.Tasks;

For example:

```
using System.Reflection;
using System.Drawing;
using System.Drawing.Imaging;
using System.Drawing.Drawing2D;
using System;
using System.Collections.Generic;
using System.Text;
using System.Threading.Tasks;

namespace VPPlugin
{
   public class MyPlugin
   {
     public Image OutPlugin(Image argImage,out string argMessage)
     {

          Bitmap OutputImage = new Bitmap(argImage.Width, argImage.Height);
         /// TO DO Program

        for (int y = 0; y < OutputImage.Height; y++)
        {
            for (int x = 0; x < OutputImage.Width; x++)

            {
                Color c = ((Bitmap) argImage).GetPixel(x, y);

                 int luma = ((c.R)+(c.B)+(c.G))/3;

                    OutputImage.SetPixel(x, y,
                       Color.FromArgb(luma, luma, luma));
            }
        }
         argMessage = "Output"; // TO DO External Message
         return (Image)OutputImage;
     }

   }
}
```

Run and Compile

After editing program you can compile it with “Build->Compile” menu or press F9 to find errors. You can see compiler and system message on “Messages Window” in the bottom of the page (Fig 7-5).

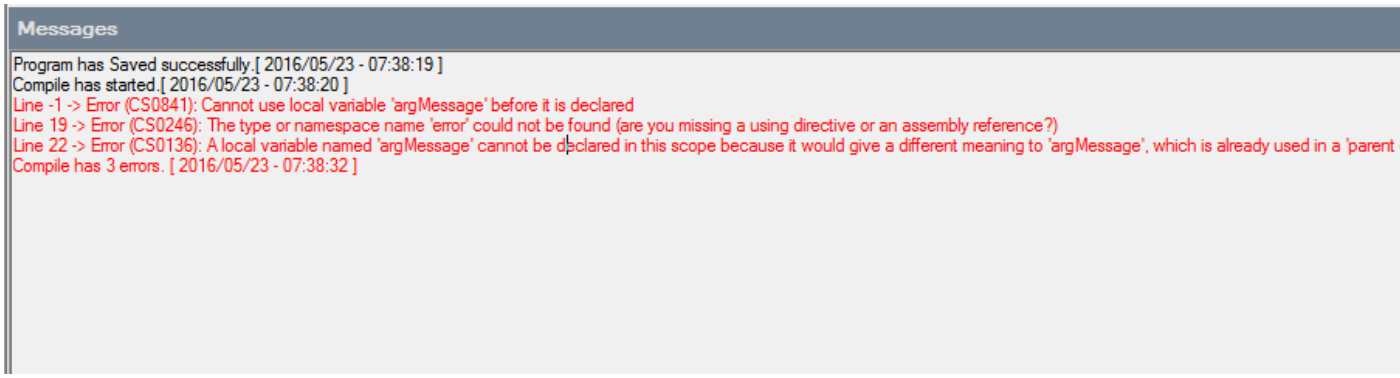

Figure 7-5. Messages window after compile

Some of these messages are built by Vital Pixels and some of them are generated by C# compiler. If you double click on any error line cursor in editor would went to its position. Also after any run you can see value of argMessage in message window. With buttons on the top of window you can save, clear and close messages.

After compile you can run your program with “Build->Run” or press F5 and it will be applied to input image and will be shown in output image.

In you start running a program software automatically start saving and compiling, so compile and save it is not necessary for any run. Many important point for run is selecting input image, so if you didn’t select it you program couldn’t be run and Vital Pixel would show error to user.

Using MATLAB

In Vital Pixels you can import MATLAB script in M-File and use it in your algorithm. If you have M-File you can use “Build->Import MATLAB Script”. As you see in Fig 7-6 you should select installation folder of MATLAB and then choose your M-File. For using MATAB Script you should check “Import MATLAB Script to Compiler”.

After that you should press “Import” button and waiting for some minutes.

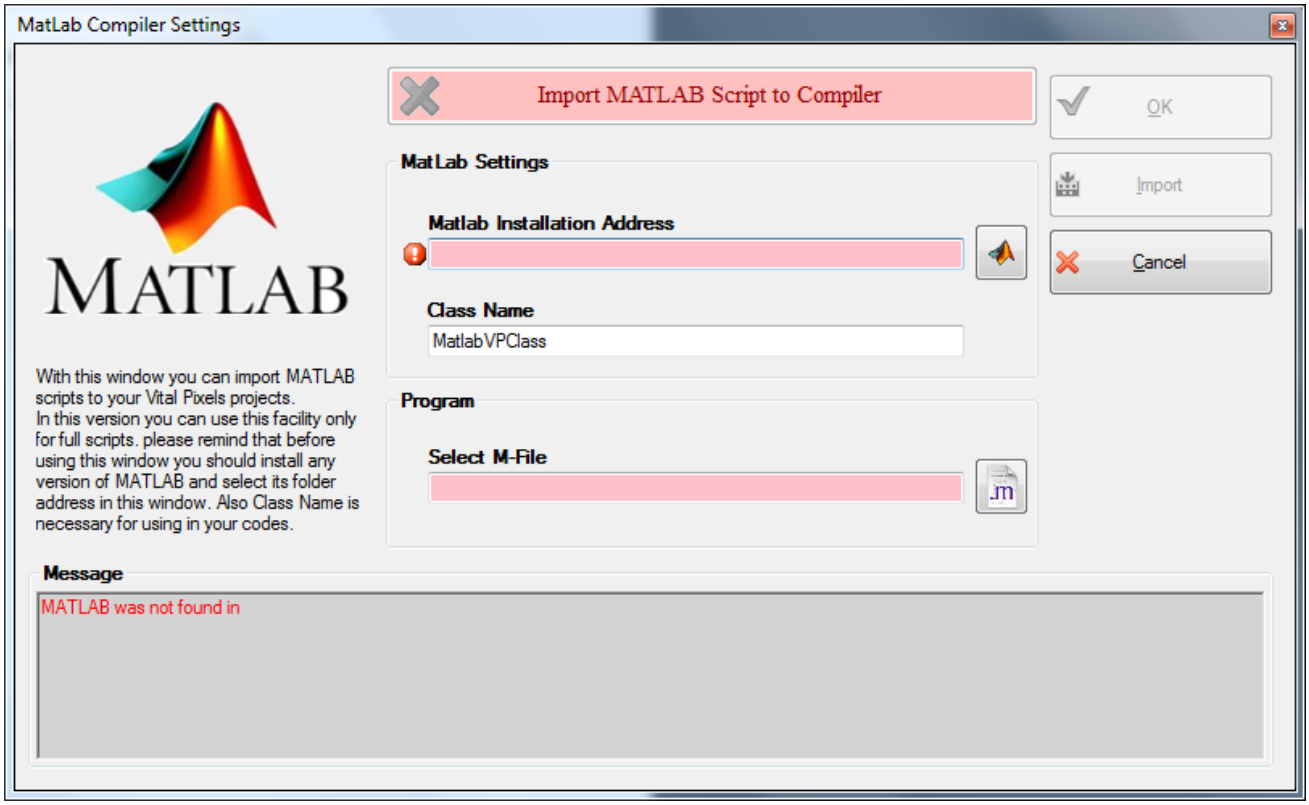

Figure 7-6. Import MATLAB script to Vital Pixels

This ability is supported only in FullCode . For example in following code you can see a simple code witch use m-file. MWNumericArray is a type that you should use to import value from MATLAB to C#.

// Mohsen.m

```
function [z] = add(x,y)

z=x+y;

%%
```

// Full Code

```
using System.Drawing;
using System.Drawing.Imaging;
using System.Drawing.Drawing2D;
using System;
using System.Collections.Generic;
using System.Text;
using System.Threading.Tasks;
using System.Reflection;

namespace VPPlugin
{

   public class MyPlugin
   {
     public Image OutPlugin(Image argImage,out string argMessage)
     {
```

```
            Bitmap bm = new Bitmap(argImage.Width, argImage.Height);
          /// TO DO Program

              Class1 v           = new Class1();
              MWNumericArray aa = new MWNumericArray(15);
              MWNumericArray ab = new MWNumericArray(6);
              aa                 = 12;
              ab                 = 1;
              double a           = Convert.ToDouble(v.mohsen(aa,
ab).ToString());

              argMessage = "Message"; // TO DO External Message
          return (Image)bm;
      }

    }
}
```

Import another library

If you need to use another .NET library in you program Vital Pixels can solve your problem. For using this ability you should use “Settings->Compiler Settings”. In this window you can import another library from GAC (Global Assembly Cash) or choose it directly. If you use FullCode you should know their namespaces but in SemiCode it will be added automatically.

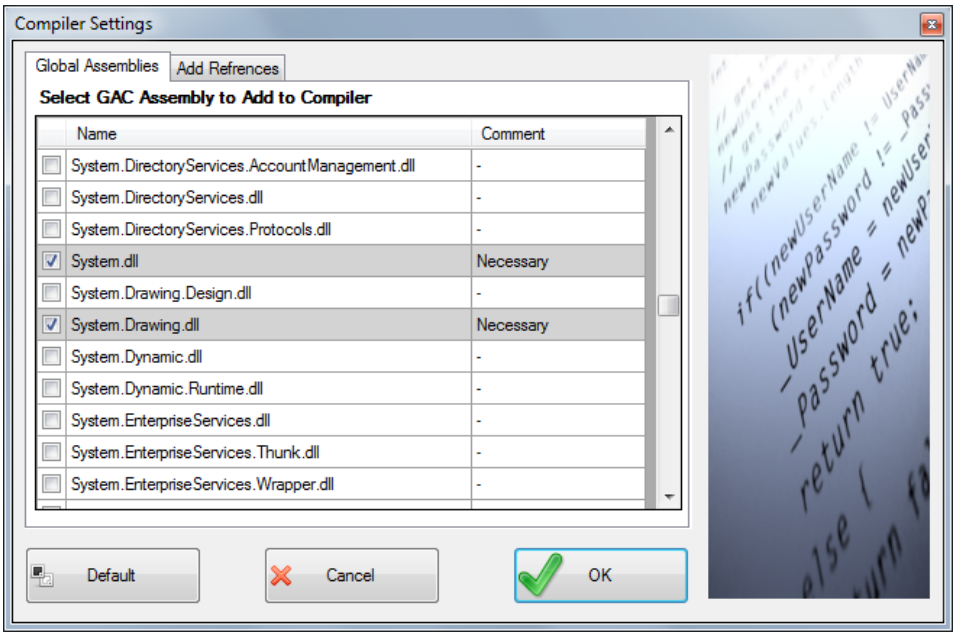

Figure 7-7. Use another library in your program

8. Plugin

One of the important ability in Vital Pixels is plugins. You can make plugin easily in software also you can build it in another IDE such as Visual Studio.

Install and Uninstall Plugin

Installing plugin is so easy and you can use “Plugin->Plugin management”. As you see in Fig 8-1

you can browse your plugin as DLL or EXE , and press install button.

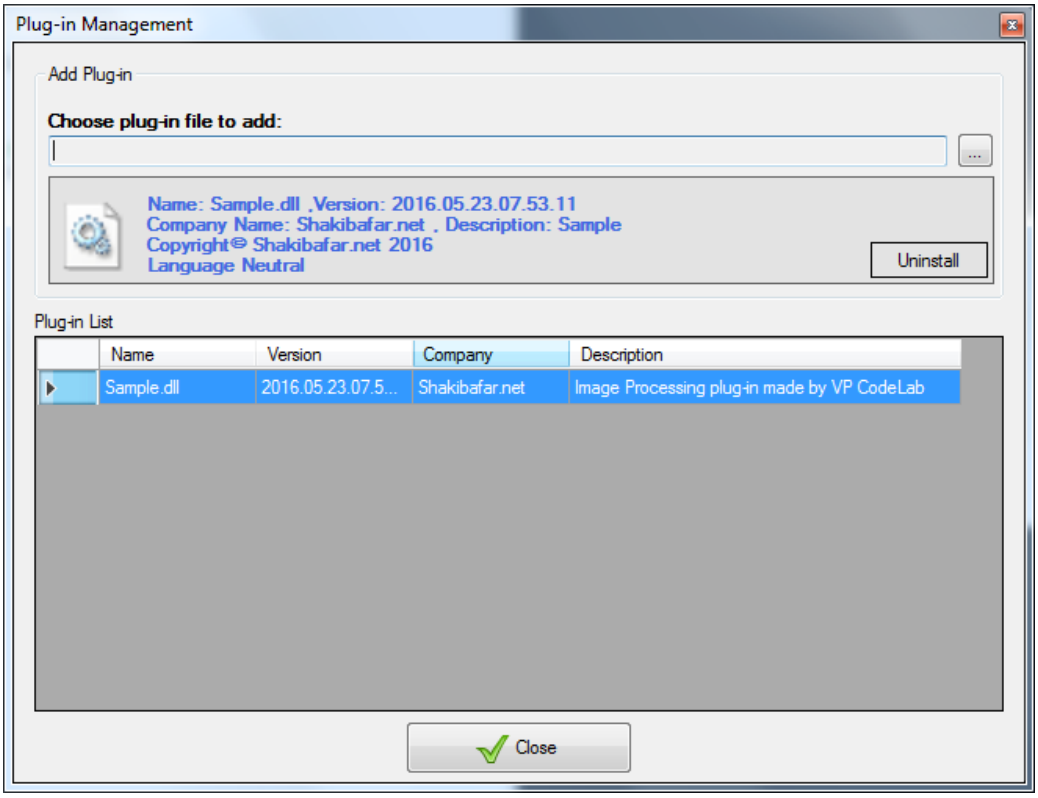

Figure 8-1. Plugin Management

If you use another IDE you must write your program such as a FullCode. If you your plugin are not compatible with Vital Pixels, you would see following error message (Fig 8-2).

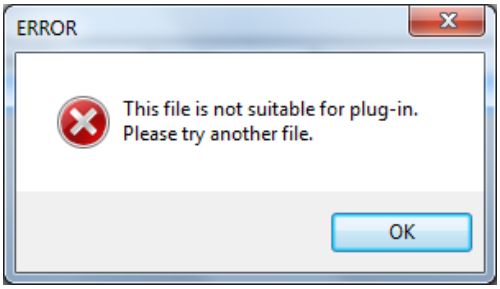

Figure 8-2. Error message if plugin was no compatible

If your plugin was not useful you can uninstall it with right click and choosing “uninstall”.

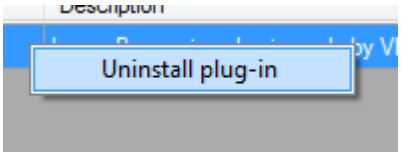

Figure 8-3. Uninstall plugin

Build a Plugin

Building a plugin in Vital Pixels is so easy. You should instead of compiling your program you should use “Build->Build as plugin” and if your program has no error a window that you see in Fig 8-

4 will be appeared.

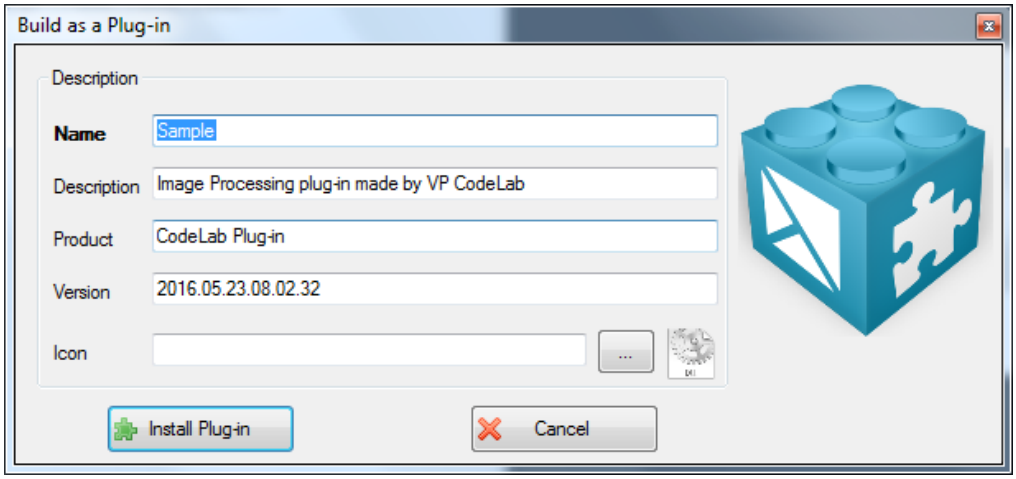

Figure 8-4. Build as plugin window

In this window you can select Name, Description Version and even an icon for your program. After that you can see you plugin in plugin menu and plugin toolbar. After installation you can run plugin for any image or send it for another users. For example you can make a plugin that can find Tumor in MRI photo and publish it for neurologists.