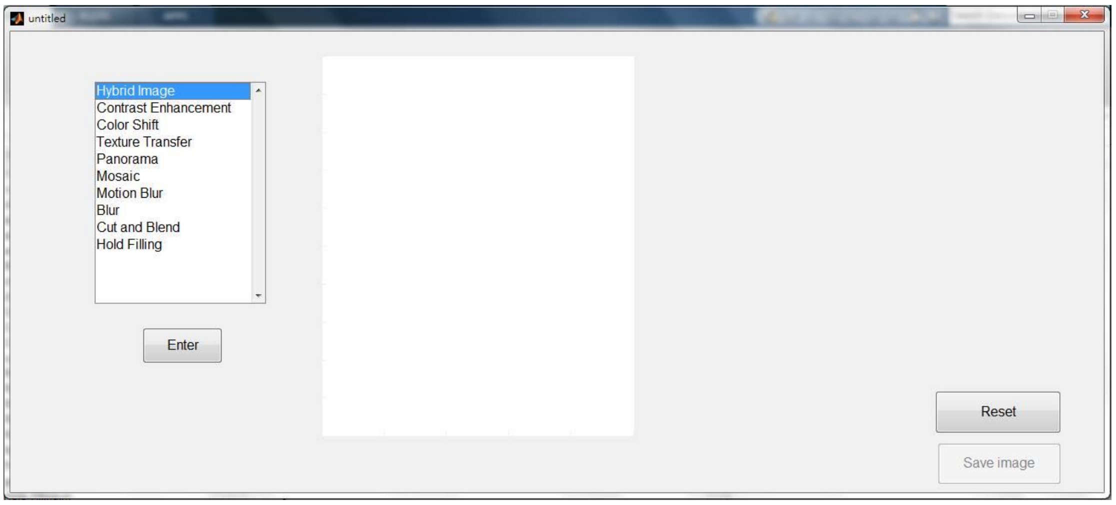
CS 445 - Final Project: Interactive Image-Editing Program

Ching-Yu, Liu　Yao-Tsung, Hsu　Yu-Ching, Huang

Introduction

The main purpose of the final project is to create an interactive program, which provides users to apply several functions. Some functions are we have learned in this semester while others are useful functions when a user wants to edit an image. In addition, the program has an advanced function “hole filling”, which is an extension from previous project.

Our Interface and Approach



At first, the user can choose a function in the left-side bar. After choosing the function and clicking enter, the right-side bar would show corresponding workspace, where the user can key in the parameters for the function.

After clicking “Run”, the program will start to run and the result will be displayed in the middle after a while. The user is able to save the result as jpg.

This is the main part of our projects, and it is also the most time-consuming one. We spent much time realizing how to create a GUI tool on Matlab. During the process, we learned how to create GUI element and set it on the program. Once we added a new element, Matlab will generate “create function” and “callback function” automatically, and we were able to write what to do in the function. We also had to understand how parameters are passed from GUI to the functions. If we need to change the value of GUI element, we can use handles of that element to "get" the value or "set" the value.

As there is variety of functions, we had to set up the limitation and scale of the parameters for each function to avoid the program crashing. For example, we need to display the value whenever the bar of the slider changes, or when the value from the slider is not an integer, we might need to reset the value and store it back to the slider.

Functions

1. Hole Filling: This is another achievement in our project.

   We implemented this function by referencing the algorithm described in Region Filling and Object Removal by Exemplar-Based Image Inpainting by Criminsi et al.

   There two key points about filling problem. The first is the order that we should fill the hole region. We choose the target point on the boundary by computing $P(p) = C(p)D(p)$, where p is the points on the boundary. $C(p)$ is the confidence value, the greater confidence, the point should be filled earlier. $D(p)$ encourages our algorithm to fill high gradient regions first. Hence, the point with highest $P(p)$ means the highest priority to filling.

   After choosing the point, we use this point as the center to decide a patch to fill. (the patch size is given by the user)

   The second is to find the fill front. We divided the image into four parts.

   Then calculate the exemplar patch, which is the fill front, from these four images by finding the minimum SSD.

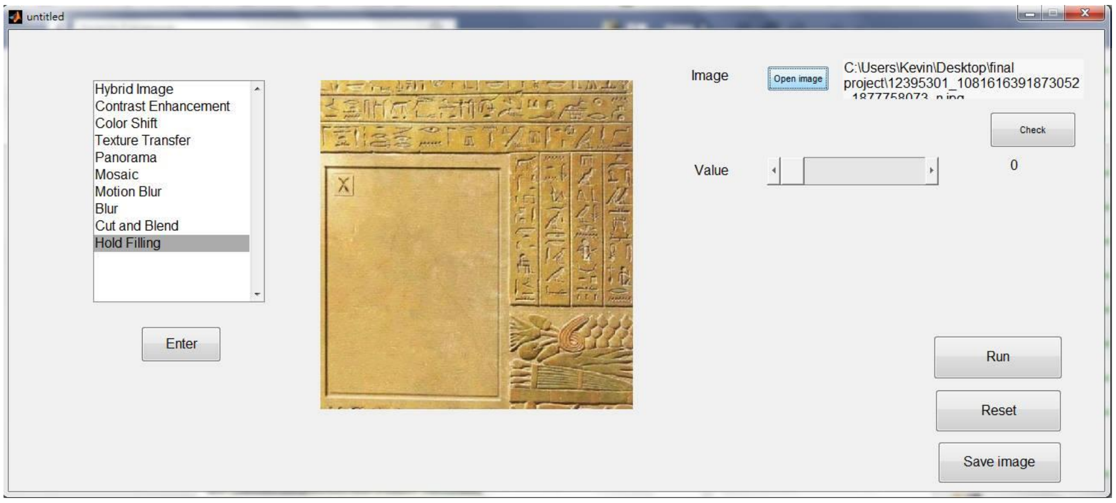
   The steps we used.

   1. User input image and select a region to cut.
   2. Get the mask for the hole and boundary.
   3. Choose the point p on the boundary that has the highest $P(p)$.
   4. Find the fill front to fill the patch which centered at p.
   5. Replace the patch by the fill front and update mask and boundary
   6. Repeat step 3~5 until the hole is filled.

   Actually we encountered two main problems.

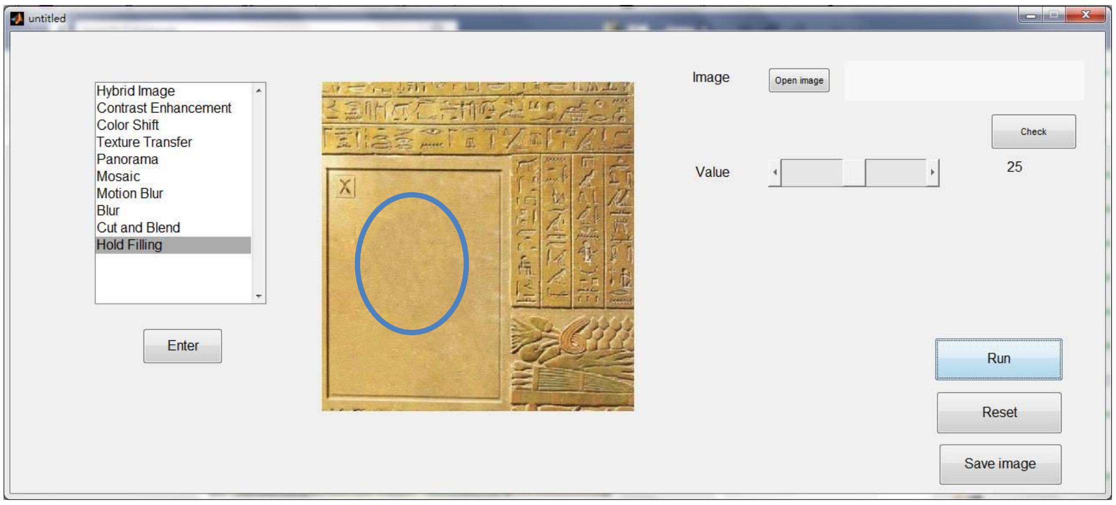
   The first is computing $D(p)$. As the paper said, $D(p)$ equal the inner product of the norm vector and the isophote at point p. However, isophote is what we didn't learn before. Finally we solve this problem by figuring out that Isophote_intensity = $\sqrt{\mathrm{fx}^2 + \mathrm{fy}^2}$, Isophote _theta = $\cot^{-1}\frac{\mathrm{fy}}{\mathrm{fx}}$, where fx, fy is gradient in x and y direction . So Isophote = $[\mathrm{Isophote_{intensity}} * \cos(\mathrm{Isophote\ \_theta})\ , \mathrm{Isophote_{intensity}} * \sin(\mathrm{Isophote\ \_theta})]$

The second is that even we used SSD to find the most similar fill front to the patch that needed to fill, it is still not always good to every input images. We guess the reason is that we used the method which is implemented in project2, but the situation is a little bit different in hole filling problem. However, we didn't modify the method since we can still get some good results with the current method.



C:\Users\Kevin\Desktop\final project\12395301_1081616391873052

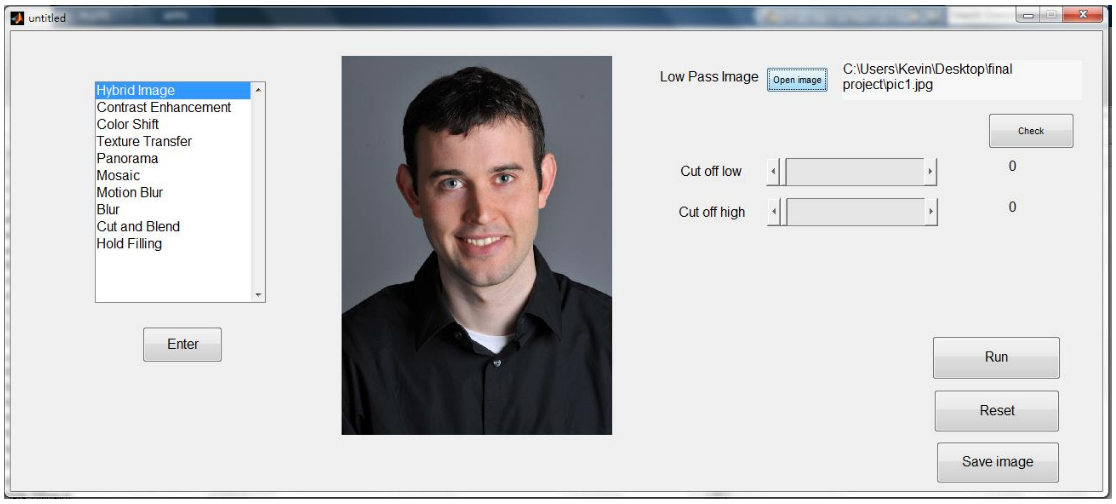
0

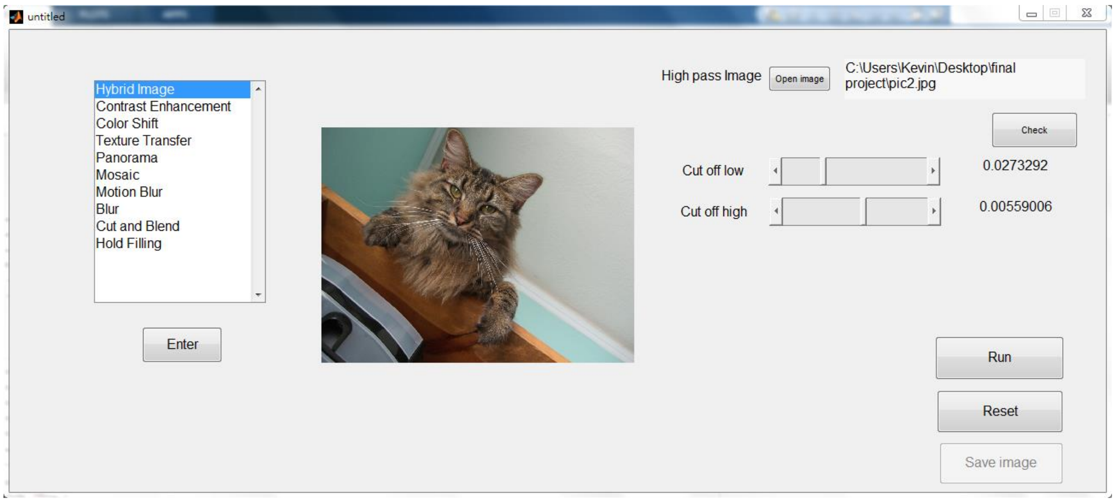


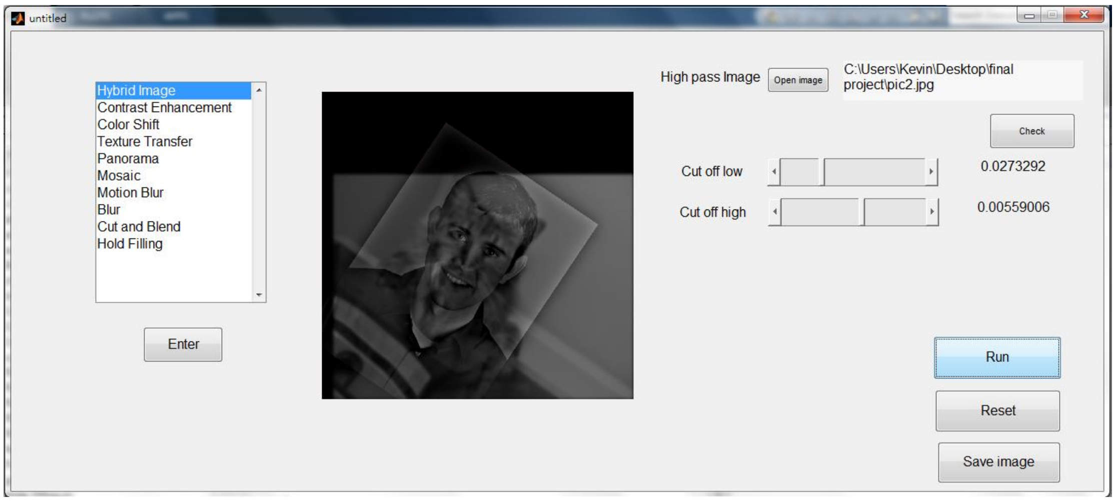
25

Circle part is where has been filled after cutting off from the original image.

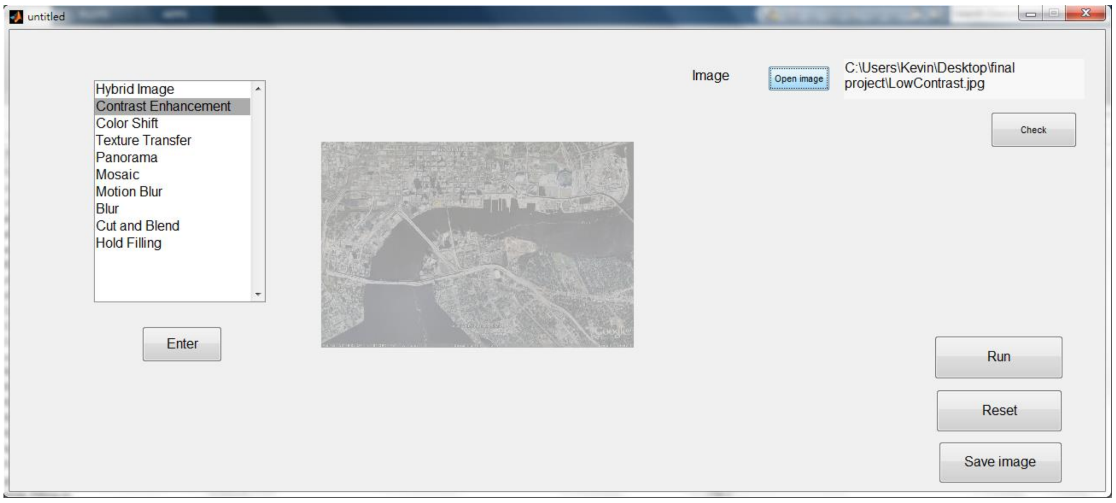
2. Hybrid Image: Hybrid two images by providing corresponding cut-off value.

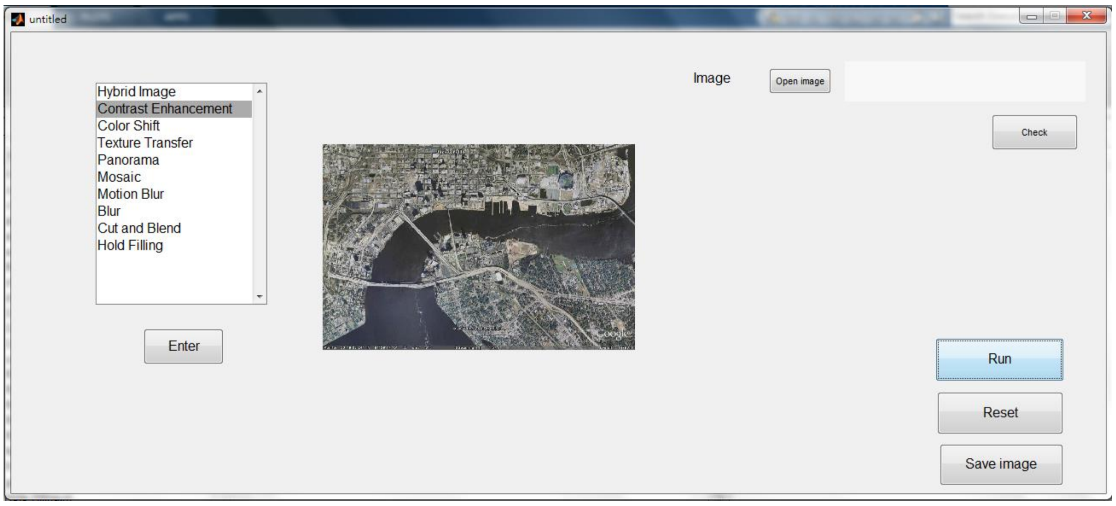




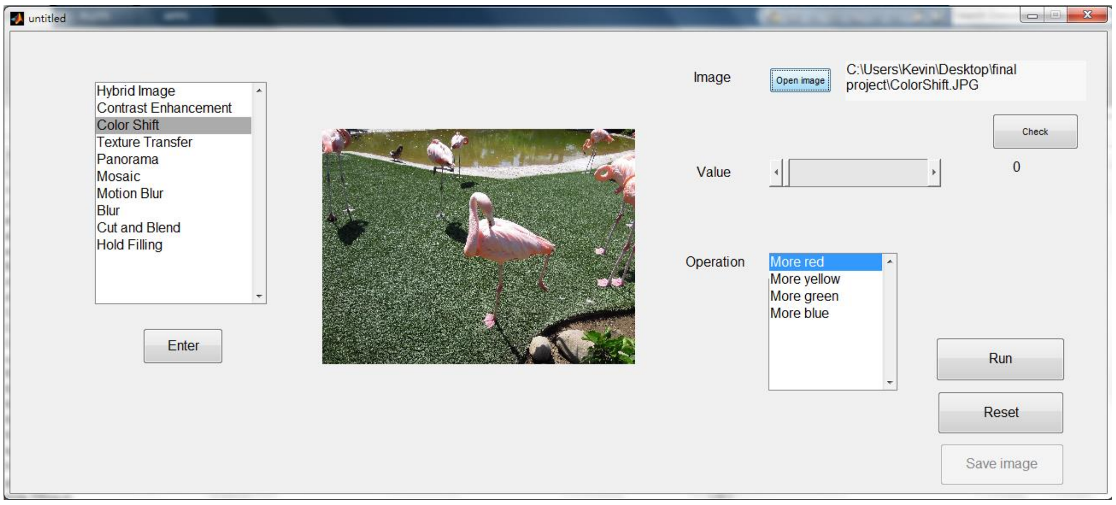


3. Contrast Enhancement: Make the image more vivid by enhancing the contrast.





4. Color Shift: Make the color of image more red, yellow, blue or green.



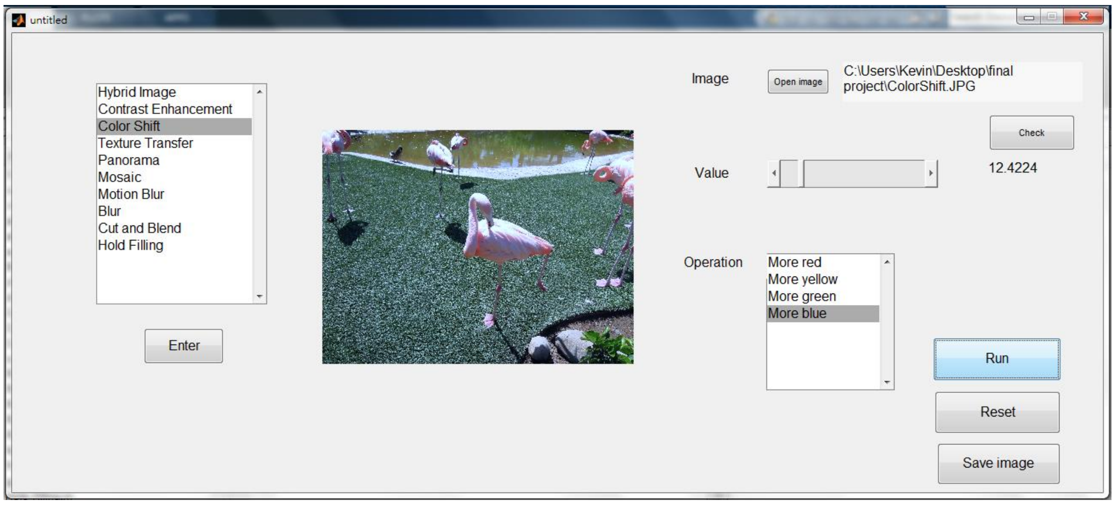
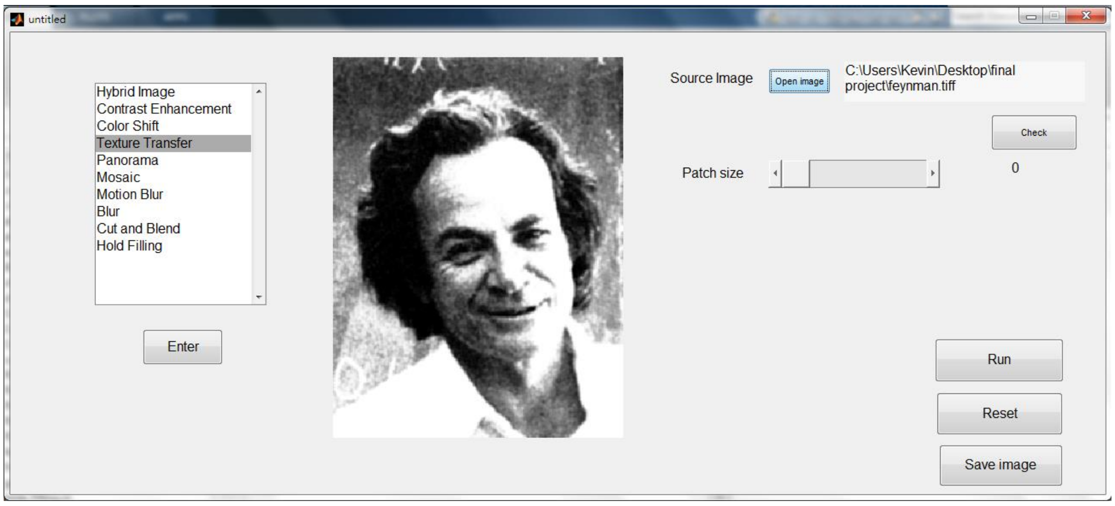


Image: C:\Users\Kevin\Desktop\final project\ColorShift.JPG

Value: 12.4224

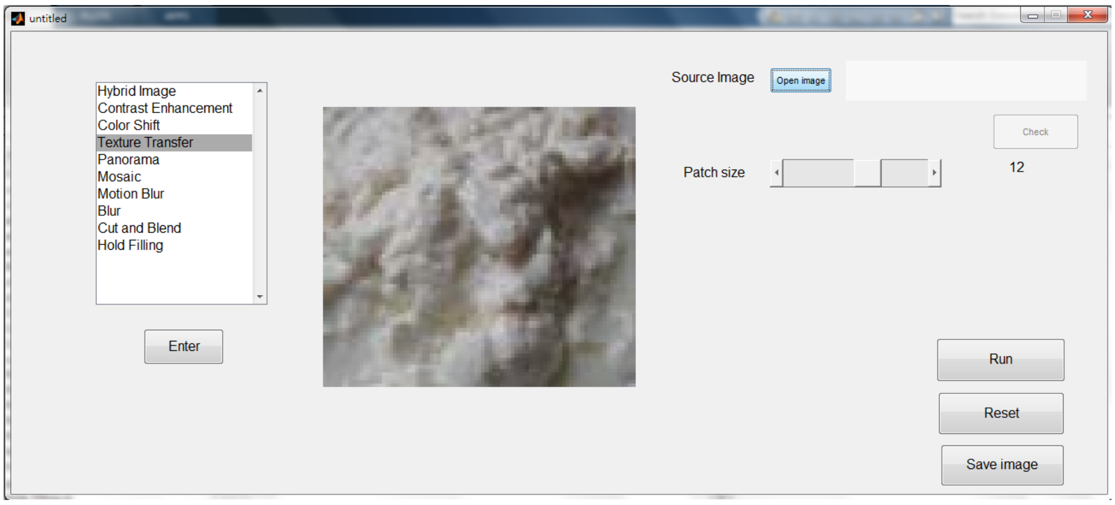
Operation: More blue

5. Texture Transfer: Giving an object, the appearance of having the same texture as a sample while preserving its basic shape.



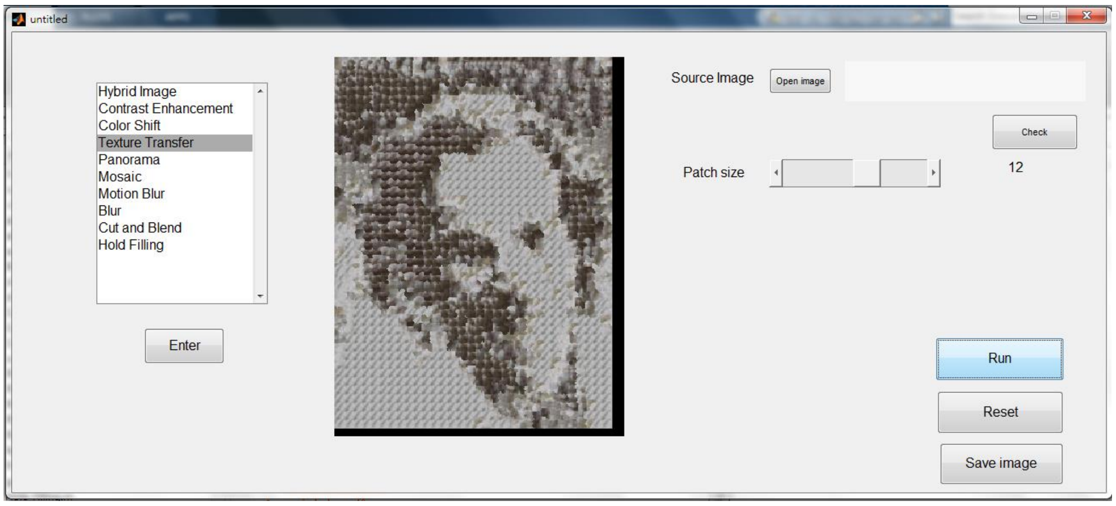
Source Image: C:\Users\Kevin\Desktop\final project\feynman.tiff

Patch size: 0

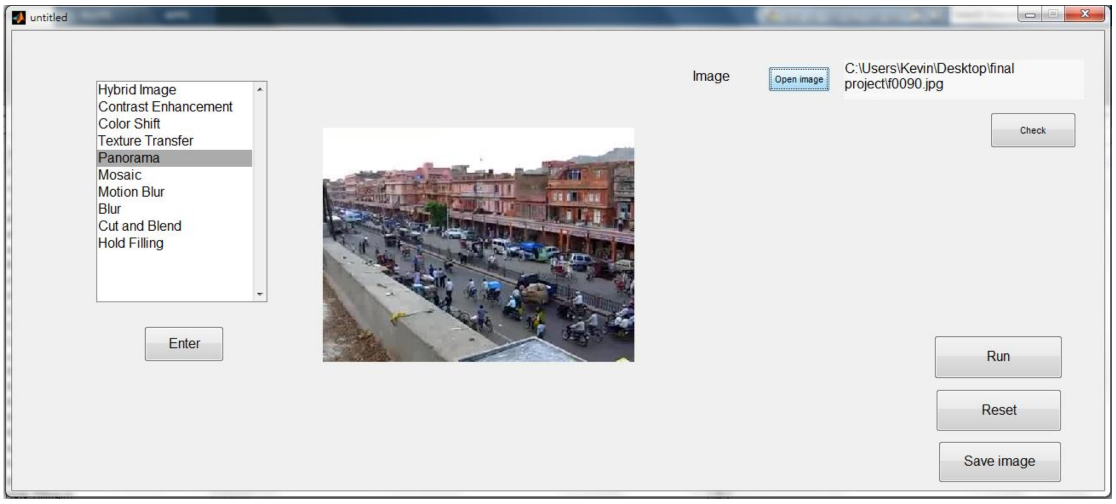


Source Image

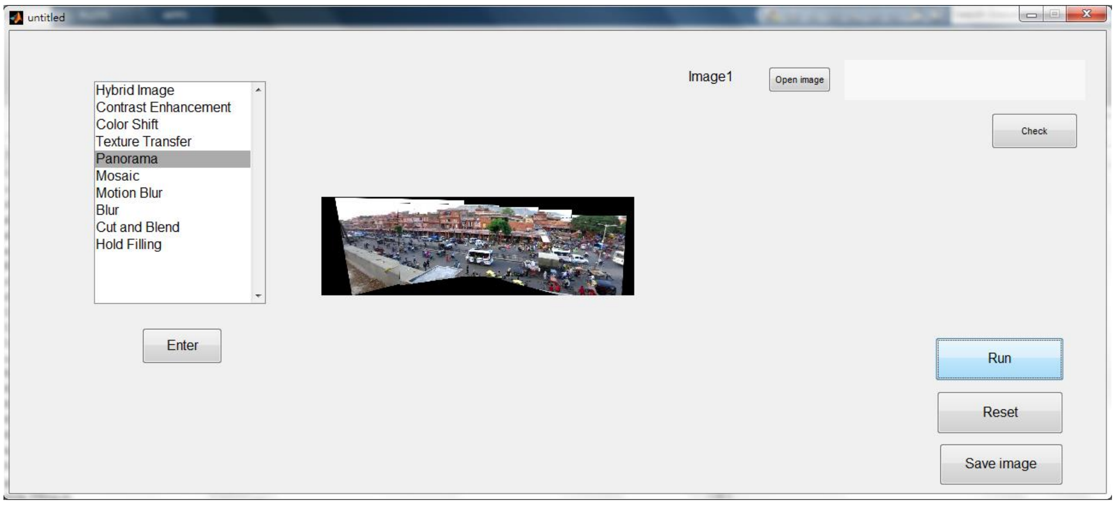
Patch size: 12



6. Panorama: Creating a panorama by providing 5 images.



C:\Users\Kevin\Desktop\final project\f0090.jpg



7. Mosaics: Pixelate the image.

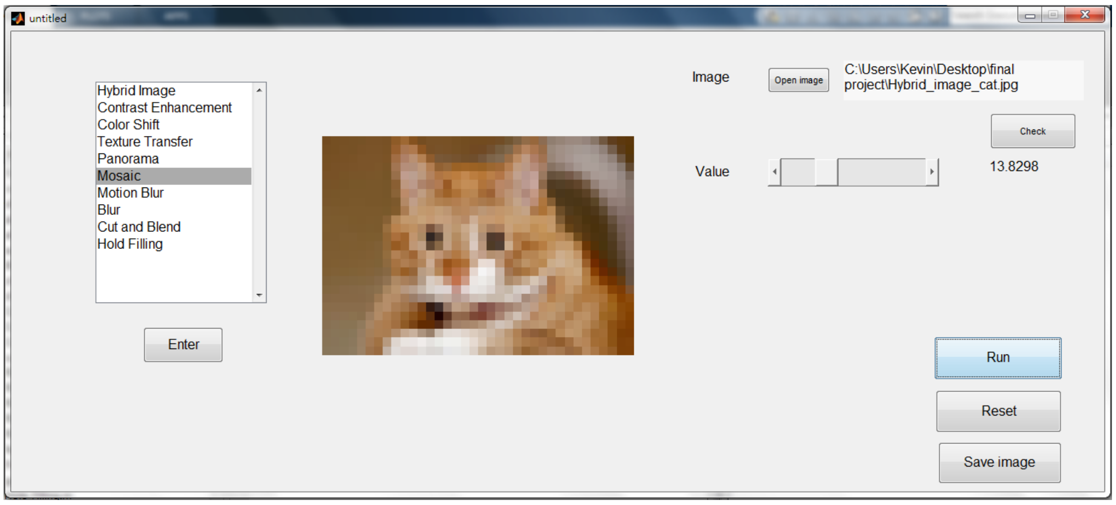


Image C:\Users\Kevin\Desktop\final project\Hybrid_image_cat.jpg

Value 13.8298

8. Motion Blur: Make the effect of motion blur.

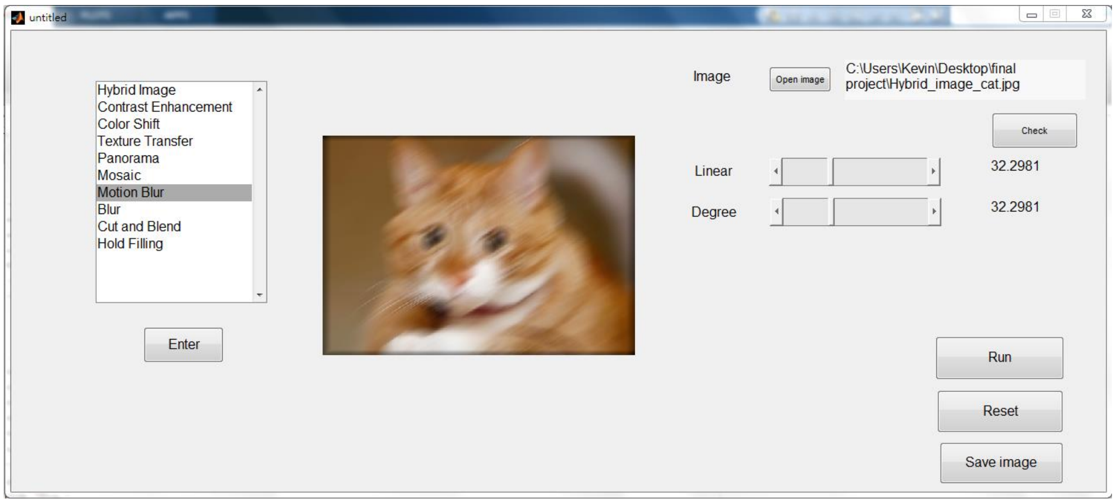


Image C:\Users\Kevin\Desktop\final project\Hybrid_image_cat.jpg

Linear 32.2981

Degree 32.2981

9. Blur: Blurring the image.

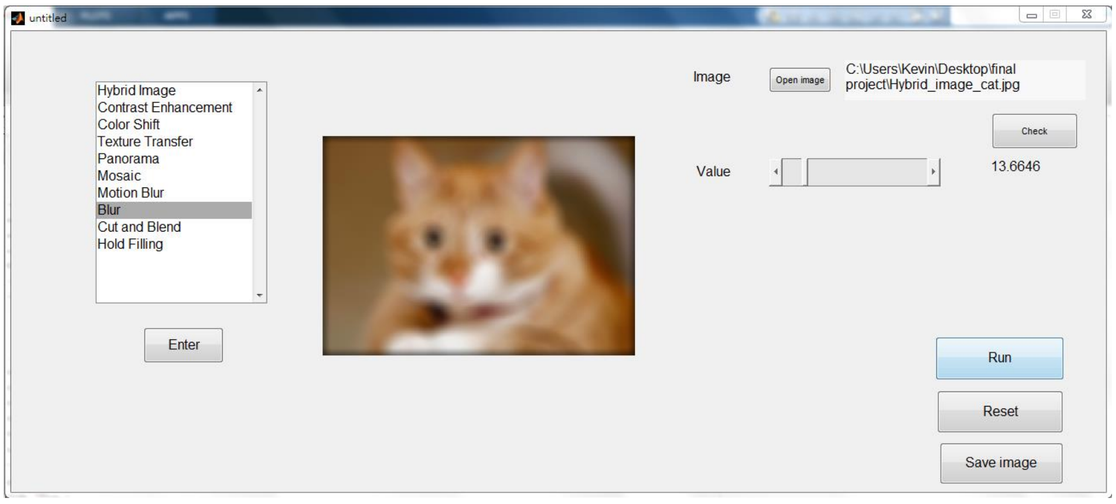
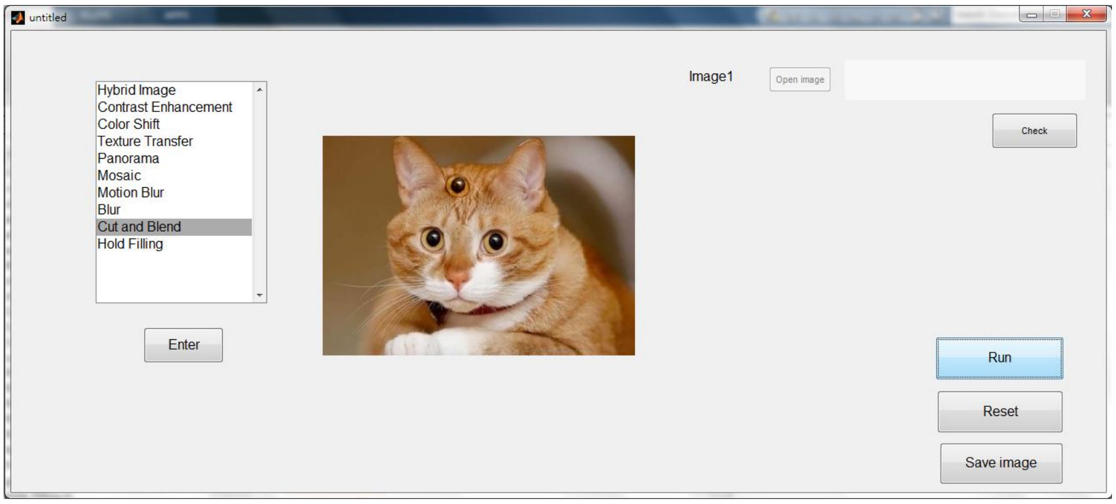


Image C:\Users\Kevin\Desktop\final project\Hybrid_image_cat.jpg

Value 13.6646

10. Cut and Blend: Cut a part of image and paste it on the other one.


untitled
Hybrid Image
Contrast Enhancement
Color Shift
Texture Transfer
Panorama
Mosaic
Motion Blur
Blur
Cut and Blend
Hold Filling
Enter
Image1
Open image
Check
Run
Reset
Save image