COMP0115: Coursework 2

Mingzhou Hu
April 2019

Part A

In part A, we produce the phase-randomized versions of natural images by randomizing the phase spectrums of the natural images while keeping the same power spectrums. Firstly, we computed the Fourier transform of an image, and then we set the phase spectrums to random values. Lastly, we compute the inverse Fourier transform for returning to the spatial domain. On the contrary, we produced whitened versions by keeping the same phase spectrums with natural images but flatting power spectrums.

The results of part A are shown below with Fig 1 and Fig 2. The first row shows the natural image with its power and phase spectrums. The second and third rows show the phased-randomized and whitened images with their power and phase spectrums respectively.

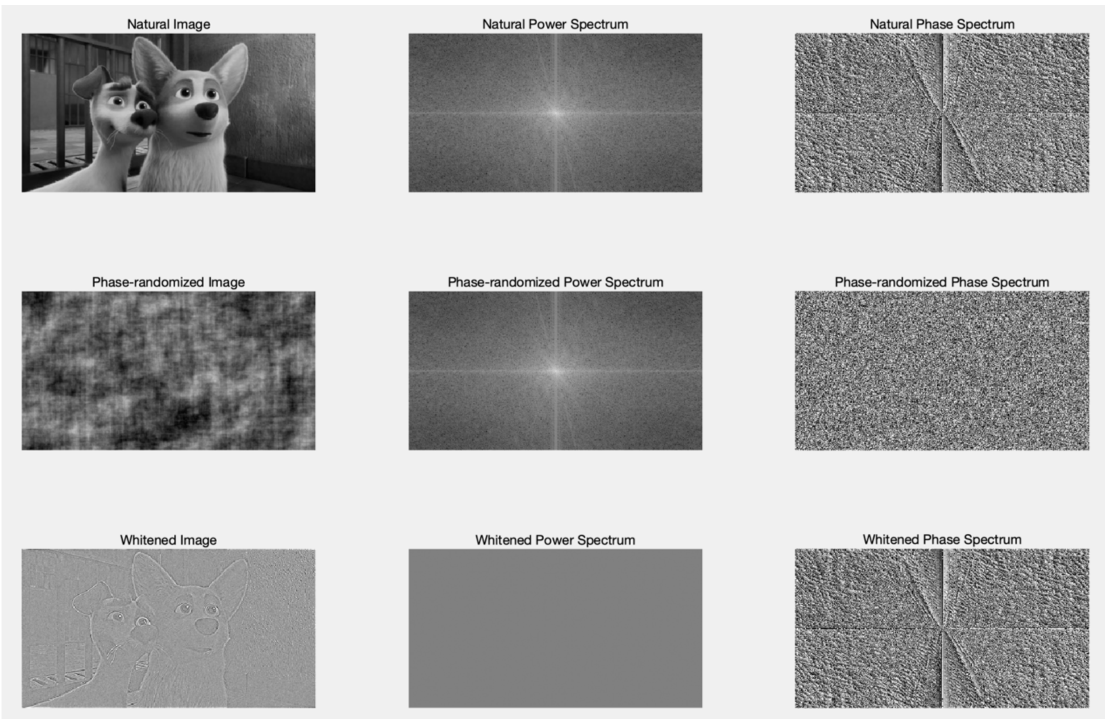

Figure 1: An example with 'dogs.jpg'

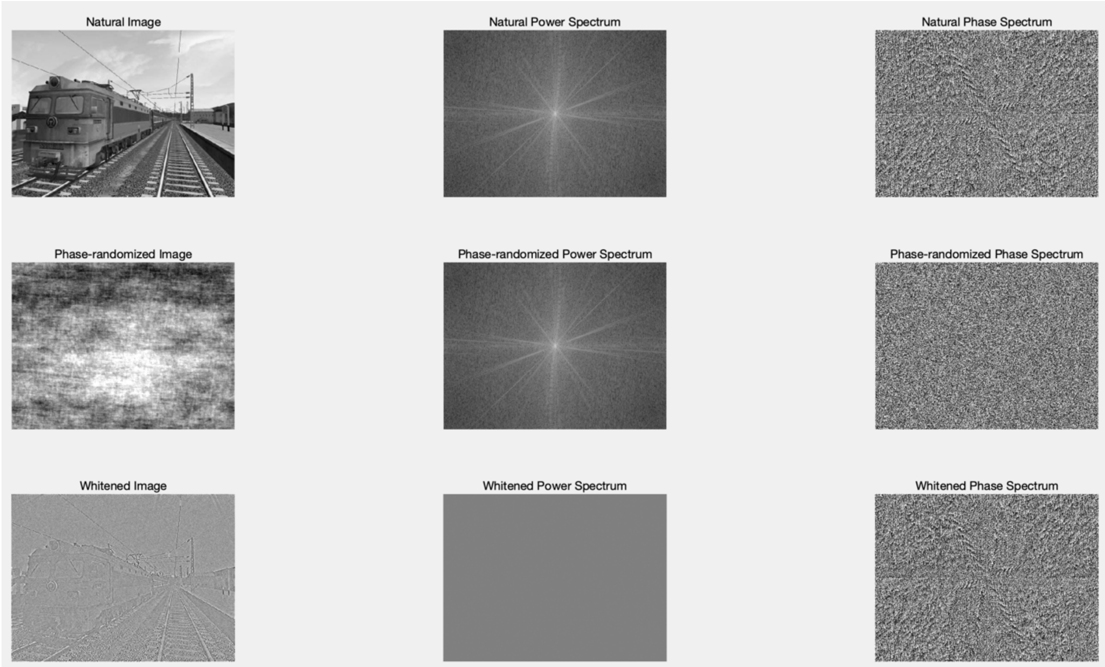

Figure 2: An example with 'train.jpg'

From the results above, we can see that the phase-randomized image loses contents of the natural image while it still keeps the intensity. The reason for the result is that the phase-randomized image keeps the same power spectrum, which is the distribution of the energy of waves used to build up the image among their different frequency components, with the natural image. However, the whitened image keeps lines and edges of contents of the natural image but loses the intensity. That is because the whitened image keeps the same phase spectrum, which is like how those waves used to build up the image are positioned, with the natural image. Therefore, from the results above, we can obtain a conclusion that in image reconstruction, power spectrum can be used to reflect the intensity of the natural image, while phase spectrum can be used to keep some important features of the natural image, such as edges and lines.

Part B

In part B, we implement a Gaussian derivative (DtG) filtering for to filter images from 0^{th} to 2^{nd} order within a cartesian basis. The derivatives of 2D Gaussian Kernel are shown in Fig 3. The top row is the 0^{th} order. The middle and bottom rows are the 1^{st} and 2^{nd} orders respectively.

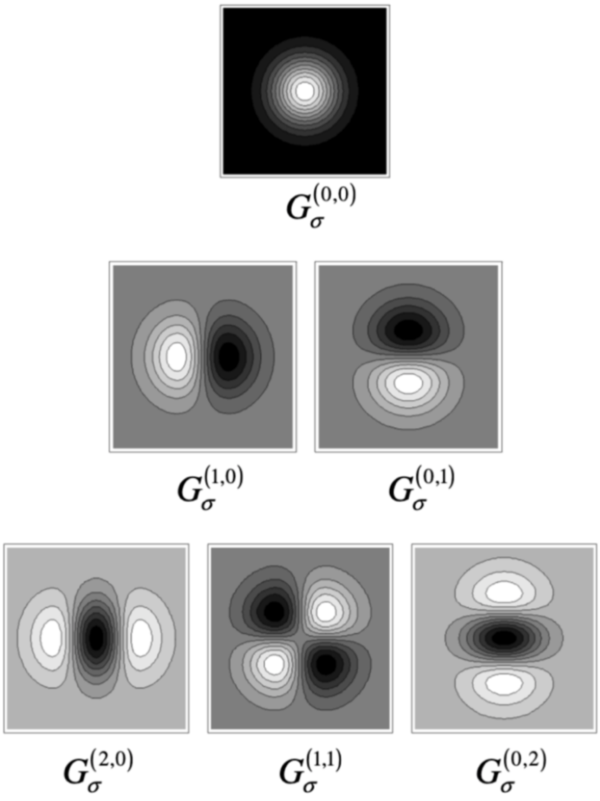

Figure 3: Derivatives of 2D Gaussian Kernel

Fig 4 shows the filtered results of the image from 0^{th} to 2^{nd} order by DtG single filter. The image shown in the top row is filtered by DtG filter in 0^{th} order. The two images shown in the middle row are filtered by two DtG single filters from horizontal and vertical directions respectively in 1^{st} order. The three images shown in the bottom row are filtered by three DtG single filters from horizontal, diagonal and vertical directions respectively in 2^{nd} order.

Fig 5 shows the filtered results of the image from 0^{th} to 2^{nd} order by DtG family filter. The first image is original image. The second image is filtered by DtG filter in 0^{th} order. The third image is filtered by DtG family filter from horizontal and vertical directions together in 1^{st} order. The last image is filtered by DtG family filter from horizontal, diagonal and vertical directions together in 2^{nd} order.

From the results above, we can see that the image becomes blurred after filtered by DtG filter in 0^{th} order, because the result image is the convolution result between original image and 2D Gaussian filter. The 1^{st} order result keeps the edge information of the original image, which is obtained by applying DtG family filter in horizontal and vertical directions together in 1^{st} order. The 2^{nd} order result keeps the most contents of original image, including edge and intensity information, which is obtained by applying DtG family filter in horizontal, diagonal and vertical directions together in 2^{nd} order.

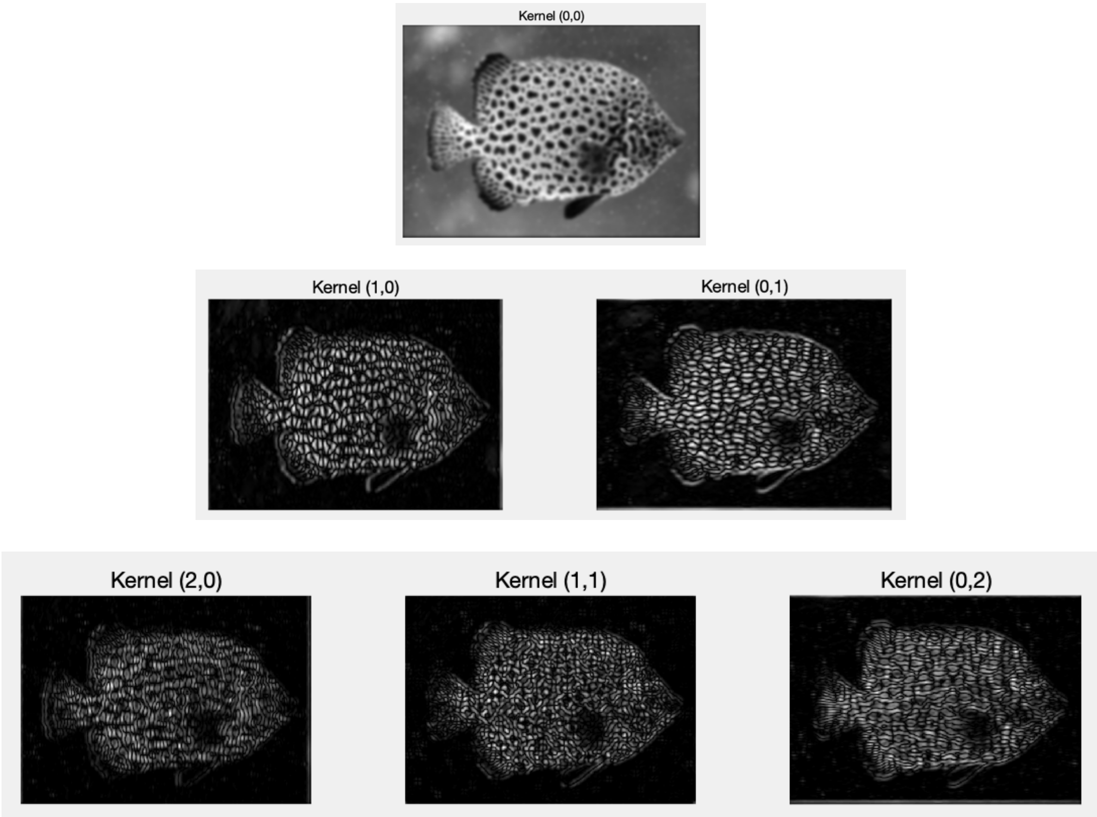

Figure 4: Filtered results of 'fish.jpg' using DtG single filter

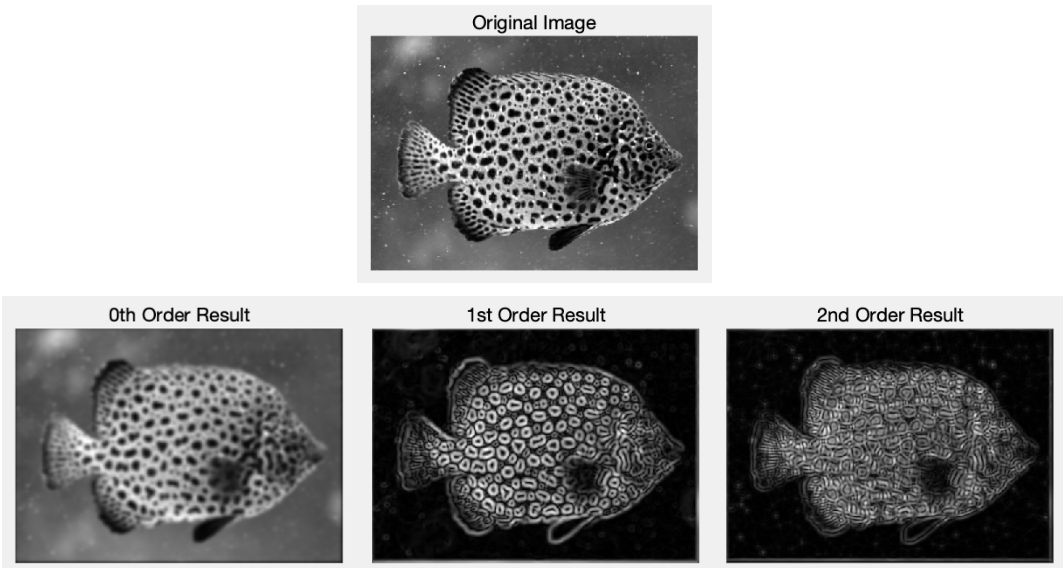

Figure 5: Filtered results of 'fish.jpg' using DtG family filter

Part C

In part C, we use the derivatives from part B to compute the Basic Image Features (BIF) classification of the pixels. BIFs can be classified to seven symmetry types by coloured pink, gray, black, white, blue, yellow, and green respectively. In this case, we set the scale of Gaussian filter to 40 to obtain a good visibility.

The BIFs results for the natural image and the comparison results for different σ and ε are shown in Fig 6, from which, we can see that if ε increased, the pink area will increase as well and other colours areas will decrease, while the image will lose features details gradually. On the contrary, if σ increased, the pink area will decrease and other colours areas will increase, while the image will lose features details gradually.

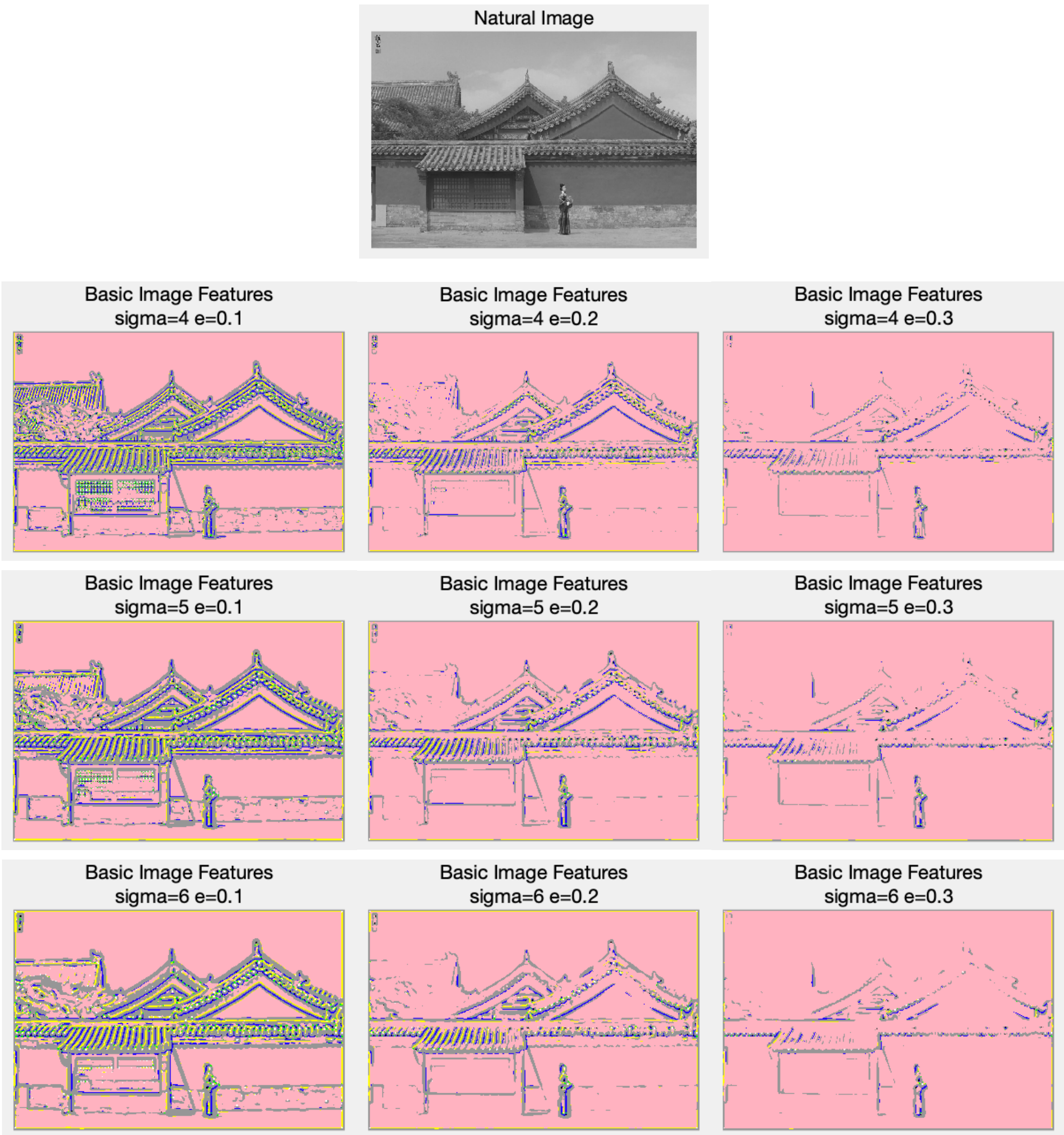

Figure 6: BIFs classification results for "house.jpg"

Fig 7 shows the comparison of BIFs classification results for natural and phase-randomized images. From the results above, we can see the result for phased-randomized image cannot be used to illustrate the basic image features, because the phased-randomized image loses most of feature details only keeps intensity information of the natural image due to the random values for phase spectrums.

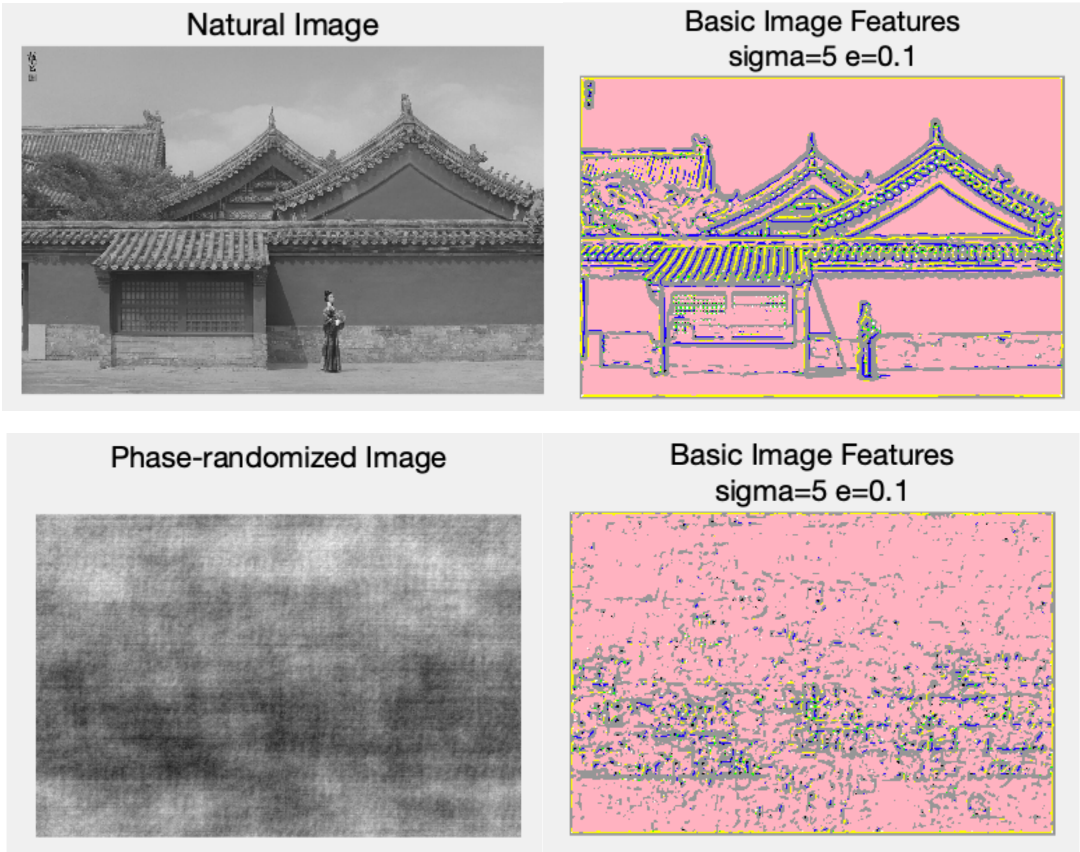

Figure 7: BIFs classification results for natural and phase-randomized images

Fig 8 and Fig 9 shows the histogram of BIFs for natural image and phase-randomized image respectively (σ=4, ε=0.1). From the figures below, the numbers of pixels for different colours in the histogram of BIFs for natural image are different from that in phase-randomized image. It means that the BIFs for natural image and phase-randomized image are different and the phase-randomized image losses most of basic image features contained in the natural image.

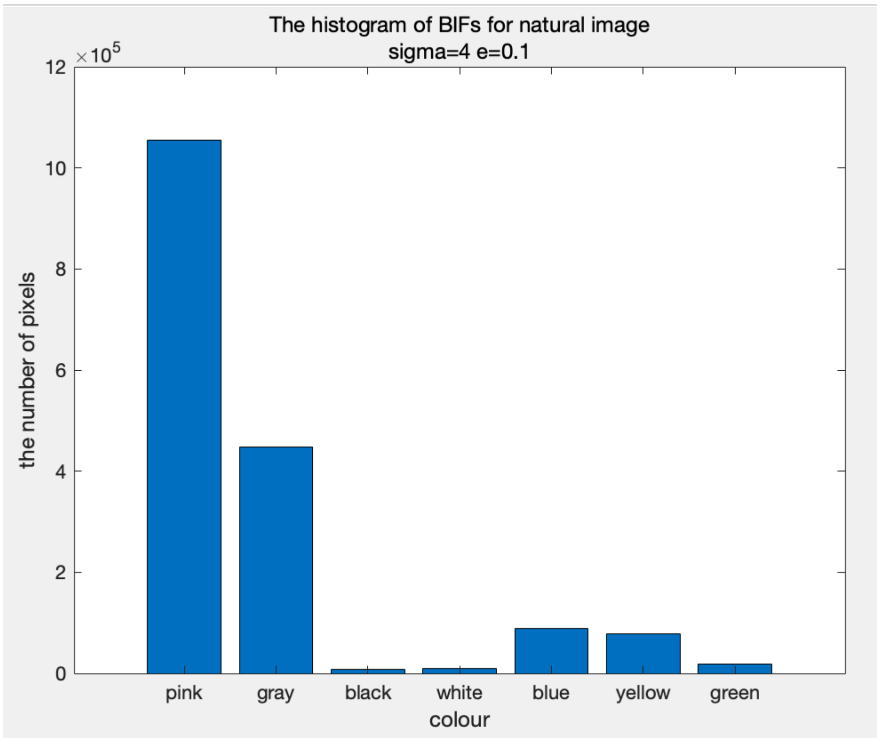

Figure 8: Histogram of BIFs for natural image

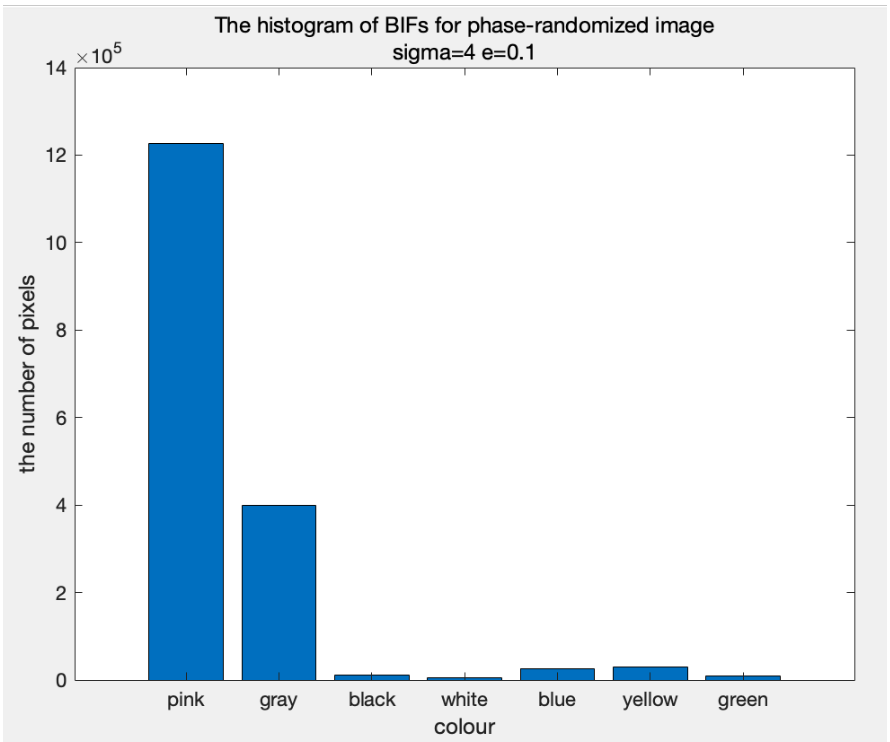

Figure 9: Histogram of BIFs for phase-randomized image