گزارش تمارین

سروش باقرنژاد

اطلاعات گزارش	چکیده
تاریخ:	انتقال تصویر به حوزه فرکانس میتواند محاسباتی که در حوزه مکان پیچیده و زمان بر هستند را تسریع کند. در این حوزه به راحتی میتوان فیلتر های دلخواه مانند smoothing یا sharpening یا edge detection را استفاده کرد و سپس به حوزه مکان برگشت .
واژگان کلیدی: تبدیل فوریه فیلتر پایین گذر فیلتر بالاگذر	

1-مقدمه

در سوال 5 قسمت 1 از ما خواسته شده تا تصاویر Im183 و Im184 را به حوزه فرکانس برده و یکبار طیف Im183 با فاز Im184 ترکیب کنیم و یکبار طیف Im184 با فاز Im183 ترکیب کنیم و نتیجه را به حوزه مکان برگردانیم

2-شرح تکنیکال

شرح قسمت 5.1

در هنگام انتقال یک تصویر از حوزه مکان به حوزه فرکانس دو مولفه مهم بوجود میاید ، یکی طیف که مقدار واقعی آن فرکانس در تصویر اصلی را نمایش میدهد و دیگری فاز که انحراف آن سینوسویید را از مرکز نشان میدهد .

در تصویر زیر میتوانید این دو مولفه را ببینید .

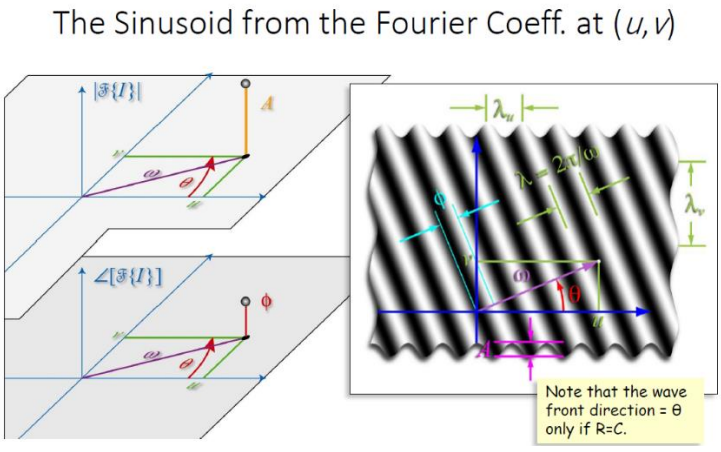


Figure 1- فاز و طیف

پس از تبدیل فوریه یک تصویر میتوانیم این دو مولفه را جدا کنیم . اما چون مقداز طیف معمولا مقدار بزرگیست ما از لگاریتم آن استفاده میکنیم . همچنین چون ما تصویر ورودی را تکرار شونده در نظر میگیریم، هنگام تبدیل فوریه باید طیف آن را شیف بدهیم تا مرکز آن در مرکز تصویر بیافتد.

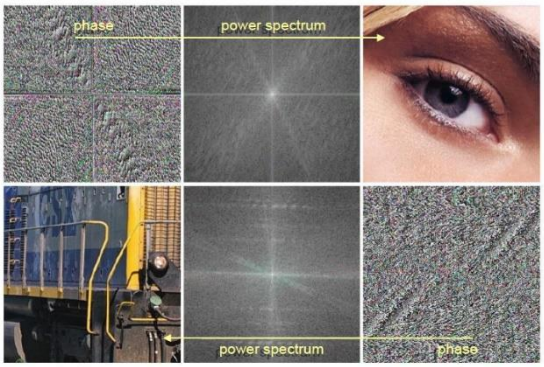


Figure 2- فاز و طیف تصاویر مختلف

فاز تصویر اطلاعات مهم تری از تصویر را در بر دارد و هنگام اعمال فیلتر در حوزه فرکانس هیچوقت نباید به فاز آن دست بزنیم و آن را تغییر بدهیم چراکه با کوچکترین تغییر فاز، تصویر اصلی بهم میریزد. ما در این حوزه فقط روی طیف فیلتر هایمان را اعمال میکنیم .

مراحل فیلترینگ در حوزه فرکانس به شرح زیر است :

ابتدا پیش پردازش های مورد نیاز را روی تصویر اعمال میکنیم. سپس تصویر را به حوزه فرکانس میبریم .

فیلتر های مورد نظر را در این حوزه اعمال میکنیم و با توجه به اینکه تبدیل فوریه یک تبدیل برگشت پذیر است ، تبدیل معکوس فوریه را انجام میدهیم تا به حوزه مکان برگردیم .

در قسمت 1 ، برای بدست آوردن طیف تصویر ابتدا تصویر را به کمک تابع fft2 از کتابخانه numpy به حوزه فرکانس میبریم سپس به کمک تابع fftshift آن را شیفت میدهیم و به مرکز منتقل میکنیم و به کمک رابطه زیر طیف را بدست میاوریم .

$$magnitudeSpectrum = 20 * \log\left(|ffshift|\right)$$

و فاز هم به راحتی به کمک np.angle که زاویه بین جزء موهومی و حقیقی را بدست میاورد ، پیدا میکنیم .

برای ترکیب فاز و طیف از رابطه زیر استفاده میکنیم :

$$|magnitude| * e^{j*\tan^{-1} phase}$$

سپس ماتریس بدست آمده از رابطه بالا را به کمک ifft که معکوس تبدیل فوریه است بدست آورده و قسمت حقیقی (Real) آن را جدا میکنیم ، این قسمت همان تصویر ترکیب شده است.

شرح قسمت 5.2

در این قسمت از ما خواسته شده تا تصاویر Im421 و Im423 را با فیلتر پایین گذر در حوزه فرکانس فیلتر کنیم .

فیلتر های پایین گذر فیلترهایی هستند که به یک تصویر در حوزه فرکانس اعمال میشوند و فرکانس هایی که از مقدار مشخصی پایین تر باشند را اجازه ورود میدهند (مقدارشان را یک میکنند) و بقیه فرکانس ها را صفر میکنند .

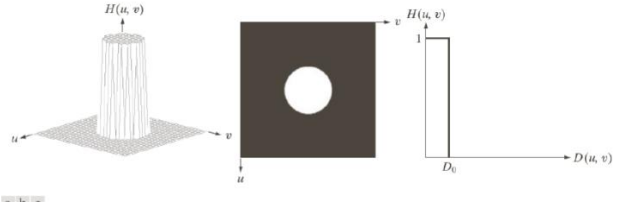


Figure 3- فیلتر پایین گذر ایده آل

همانطور که مشاهده میشود این فیلتر مرکز تصویر تبدیل شده را که حاوی فرکانس های بالا است را 0 کرده است . شعاع دایره ی جدا کننده در عکس بالا D0 میباشد .

توابع مختلفی برای اعمال فیلتر وجود دارد مانند باثروث و یا گوسی که ما در اینجا به گوسی میپردازیم .

شرح قسمت 5.3

در این قسمت از ما خواسته شده تا انچه که فیلتر پایین گذر حذف میکند در حوزه فرکانس و حوزه مکان نمایش دهیم .

به دلیل مشکل plt.subplot با تایپ complex128 مجبور شدیم با cv2_imshow عکس هارا نشان دهیم و فقط به 2 مورد عکس بجای 20 مورد بسنده میکنیم .

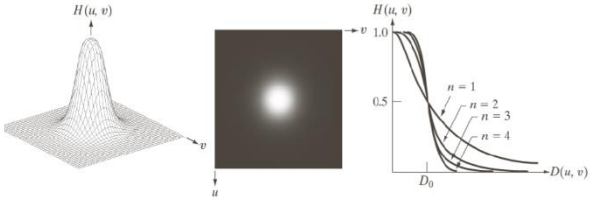


Figure 4- فیلتر پایین گذر گوسی

این فیلتر شکل بالا را دارد و از رابطه زیر بدست میاید.

$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}}$$

Figure 5- تابع فیلتر پایین گذر گوسی

در رابطه بالا D(u,v) فاصله از مرکز مدنظر است و همچنین D0 بیانگر پهنای باند ما میباشد که در این قسمت ما ده پهنای باند مختلف میدهیم.

روش کار به این صورت است که ابتدا با توجه به ابعاد تصویر و D0 یک فیلتر گوسین بدست میاوریم و این فیلتر را در تبدیل یافته تصویر در حوزه مکان ضرب میکنیم .

سپس عملیات معکوس را انجام میدهیم تا از حوزه فرکانس به حوزه مکان برگردیم .

شرح قسمت 5.4

در این قسمت ما باید دو تصویر را مانند قسمت قبل سوال با هم ترکیب کنیم به طوری که یکی از تصاویر فیلتر پایین گذر و در تصویر دیگر فیلتر بالاگذر استفاده شود .

برای این کار ما از تصاویر face1 و face2 استفاده کردیم .

چون در قسمت های قبلی بیشتر توابع پیاده سازی شده بود در این قسمت فقط فیلتر بالا گذر گوسی اضافه شد .

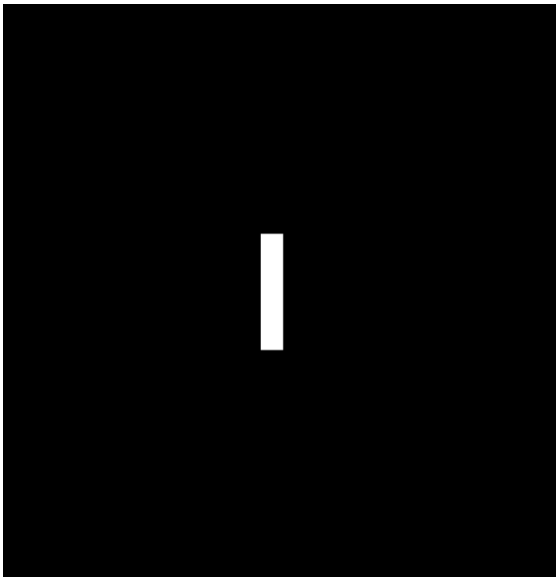
Face1 را فیلتر پایین گذر میزنیم و face2 را چون دارای لبه های بیشتری است فیلتر بالاگذر اعمال میکنیم و این با ضریب های 0.7 و 1.2 که به صورت تجربی و با ازمون و خطا بدست آمده ، ترکیب میکنیم .

3-بحث و نتایج

نتایج قسمت 5.1 :

تصویر اصلی که برای قسمت اول سوال 5 استفاده شده "Im183" و "Im184"میباشند .

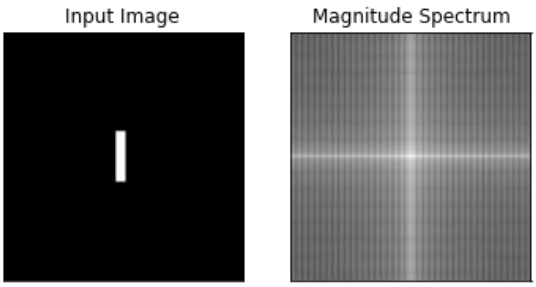
Im183 -6 Figure



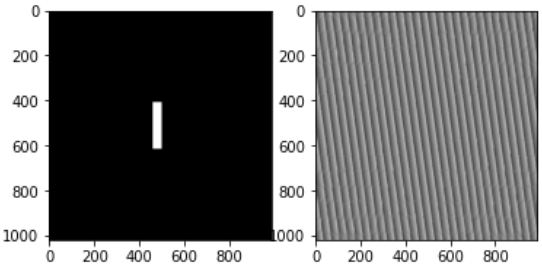
-im1847 Figure

تصویر 1 - تصویر اصلی

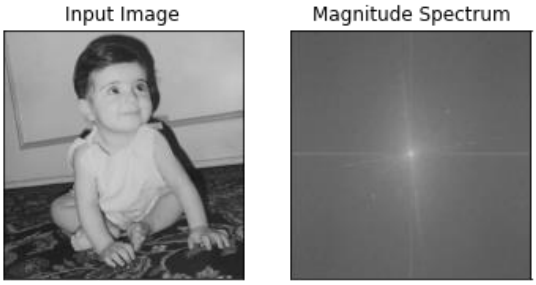
پس از انتقال به حوزه فوریه و بدست آوردن فاز و طیف :



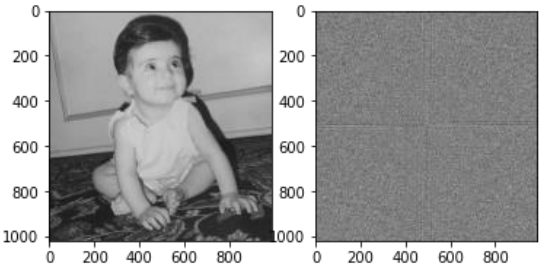
im183 طیف -8 Figure



im183 فاز-9 Figure



im184 طیف-10 Figure



im184 فاز-11 Figure

ابتدا فاز تصویر کودک را با طیف خط عمودی ترکیب میکنیم و نتیجه زیر حاصل میشود .

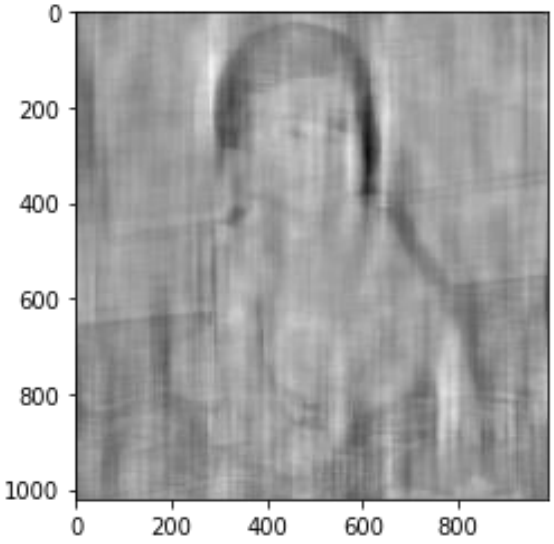


Figure 12 - فاز 183 و طیف 184

همانطور که مشاهده میشود تاثیر فاز بسیار بیشتر است و تصویر بیشتر شبیه کودک است تا خط عمودی .

حالا فاز خط عمودی را با طیف کودک ترکیب میکنیم و به تصویر زیر میرسیم .

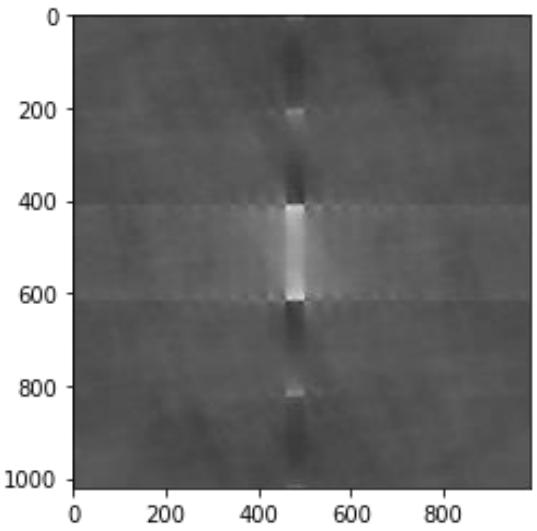


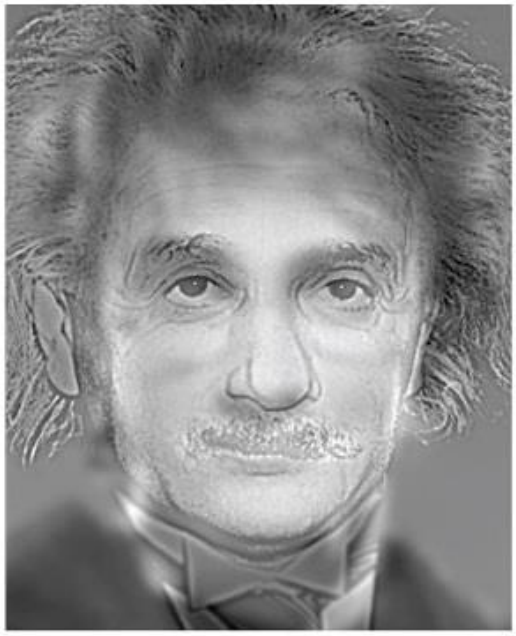
Figure 13- فاز خط عمودی و طیف کودک

همانطور که مشاهده میشود در اینجا هم قدرت فاز بیشتر است و تصویر به شکل خط عمودی در آمده است .

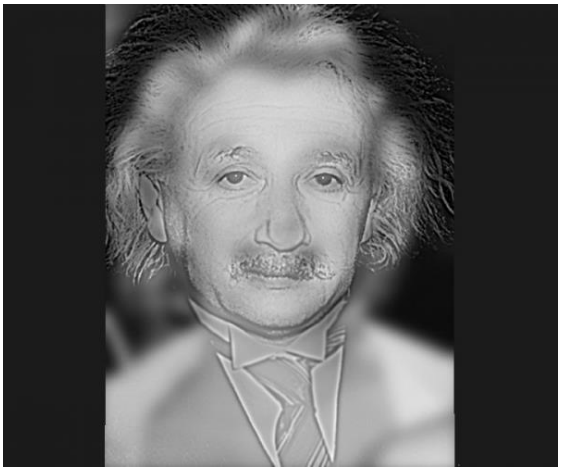
نتیجه میگیریم که هنگامی که در حوزه فرکانس قرار داریم نباید تغییری در فاز ایجاد کنیم چراکه به از دست دادن اطلاعات زیادی از تصویر اصلی منجر میشود .

نتایج قسمت 5.2:

تصاویر اصلی به شکل زیر هستند :



IM421 -14 Figure



IM423 -15 Figure

ابتدا تصویر Im421 را به کمک فیلتر پایین گذر گوسی هموار میکنیم . پهنای باند های استفاده شده به ترتیب برابرند با : 10,20,40,50,60,80,100,200,300,500

Figure 16 - پهنای باند 10

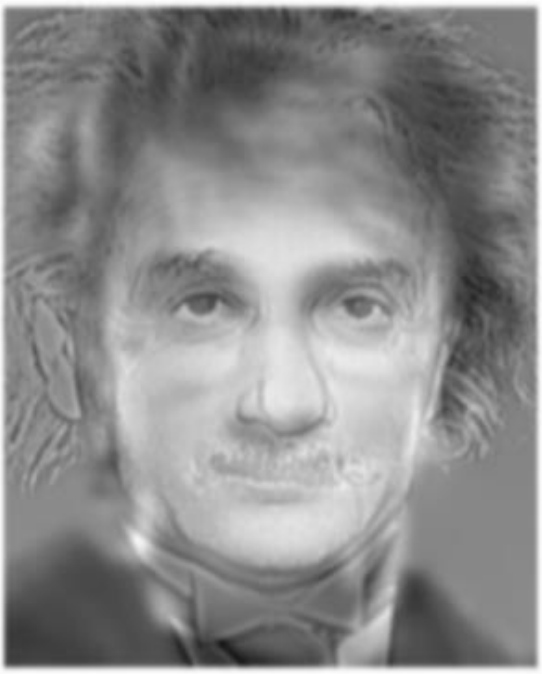


Figure 18- پهنای باند 40

Figure 17 - پهنای باند 20

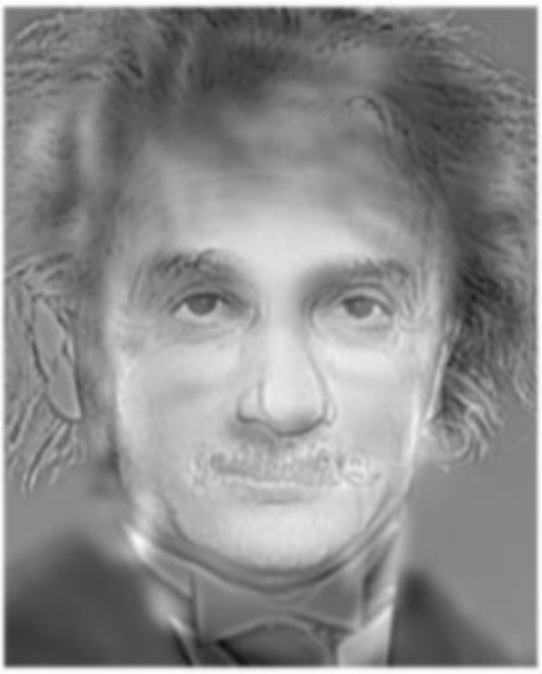


Figure 19 - پهنای باند 50

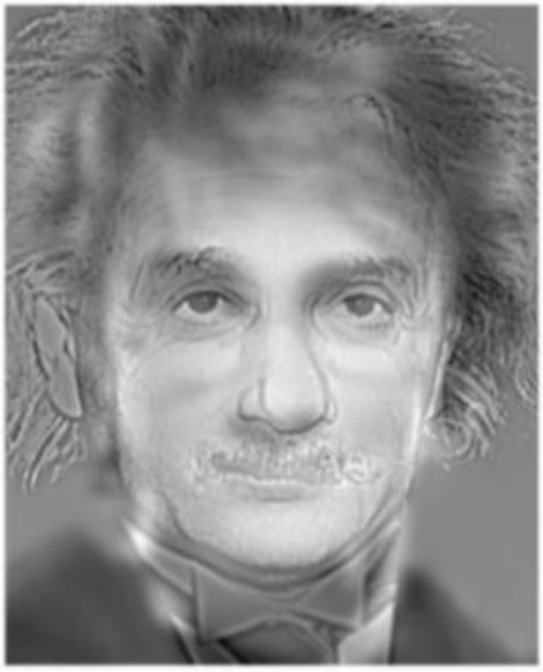


Figure 20- پهنای باند 60

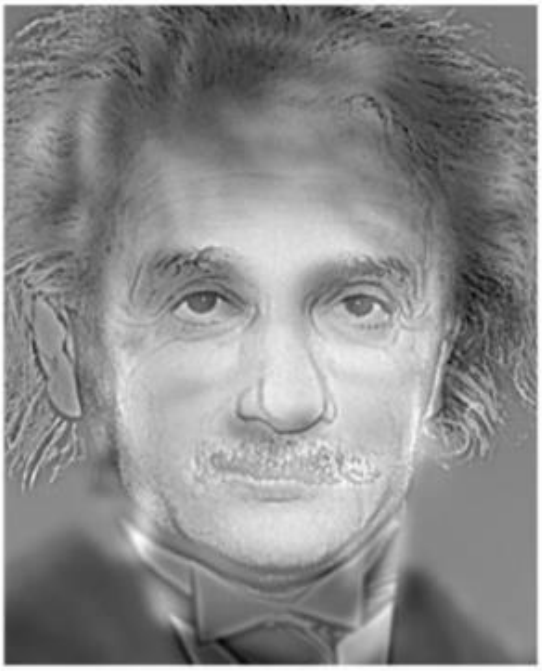


Figure 22- پهنای باند 100

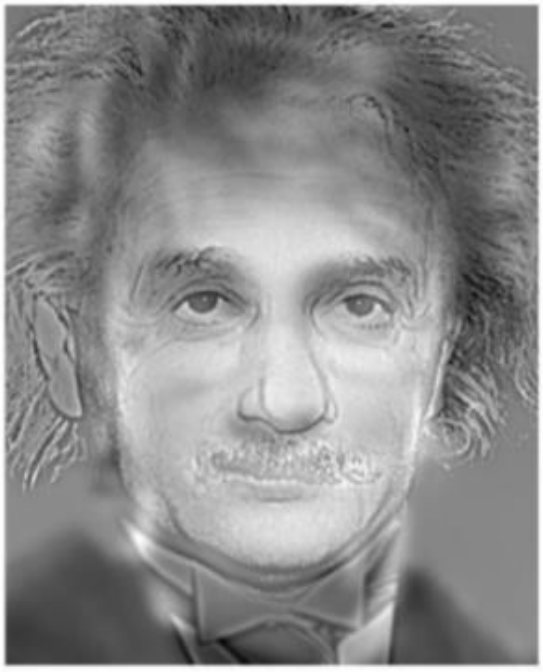


Figure 21- پهنای باند 80

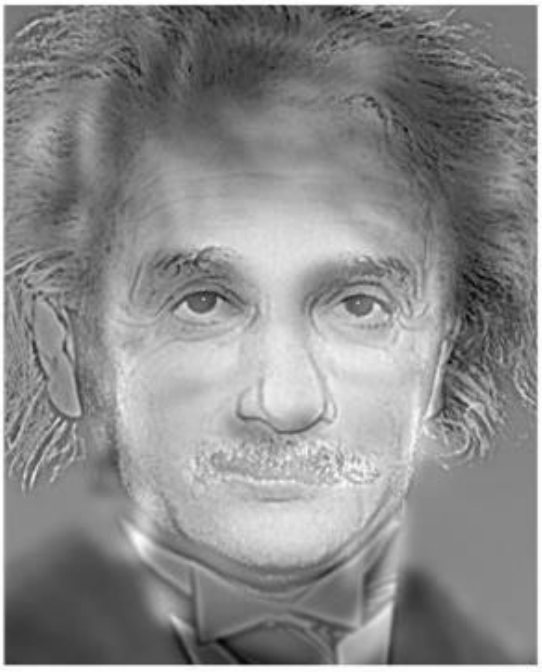


Figure 23 - پهنای باند 200

همانطور که مشاهده میشود درپهنای باند های پایین تصویر تار تر است و هرچه پهنای باند زیاد تر میشود فرکانس های بالای تصویر نیز بیشتر در تصویر نمایان میشوند .

ده عکس بعدی :

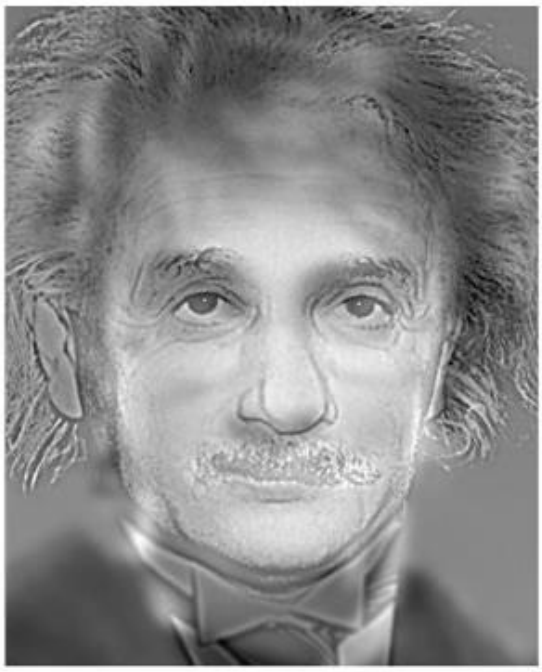


Figure 24 - پهنای باند 300

Figure 26- پهنای باند 10

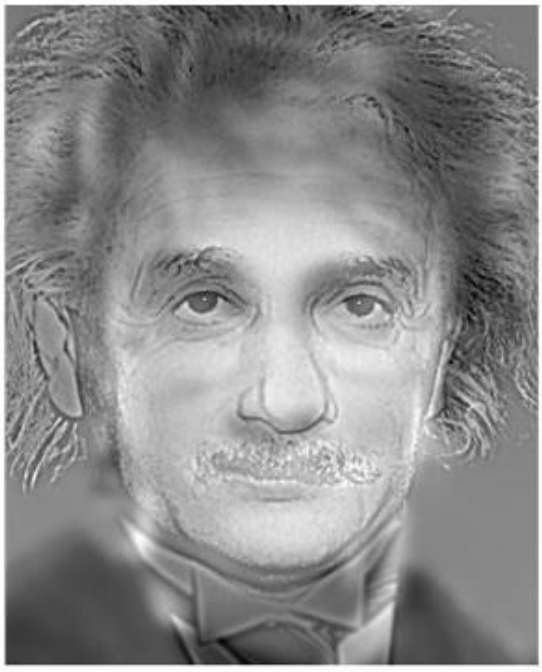


Figure 25 - پهنای باند 500

Figure 27- پهنای باند 20

Figure 28 - پهنای باند 40

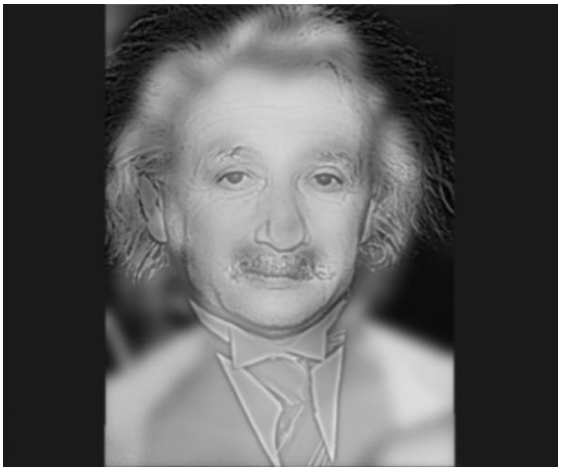


Figure 31پهنای باند 80

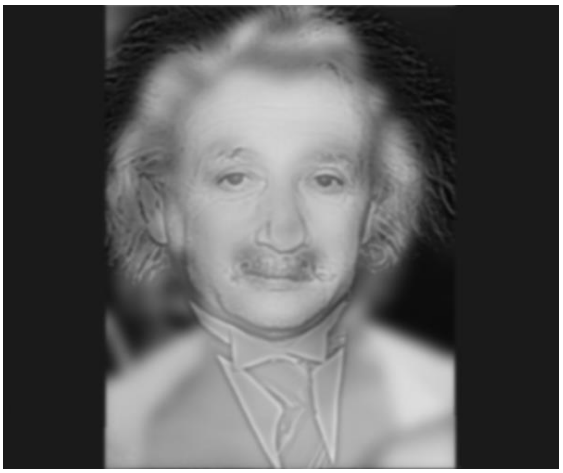


Figure 29 - پهنای باند 50

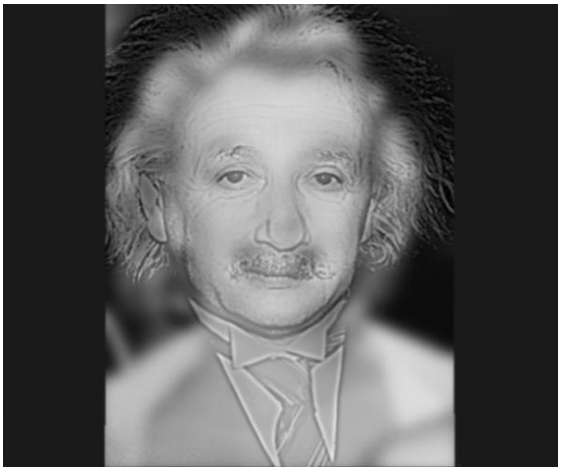


Figure 32 - پهنای باند 100

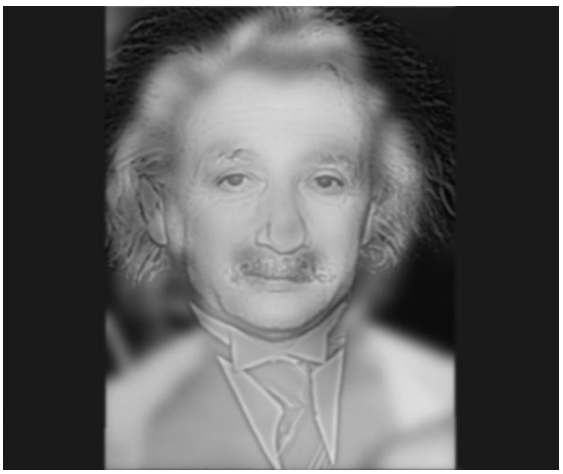


Figure 30- پهنای باند 60

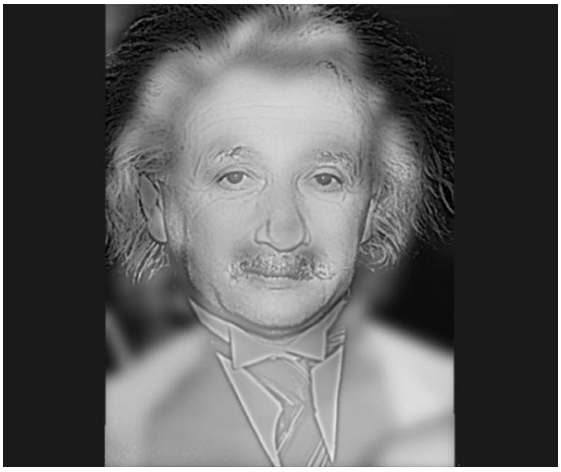


Figure 33- پهنای باند 200

نتایج قسمت 5.3 :

در این قسمت برای نمونه تصویر IM421و IM423 را به حوزه فرکانس برده و فیلتر پایین گذر گوسی را با پهنای باند 50 اعمال میکنیم . سپس در همان حوزه فرکانس تفاضل تصویر فیلتر شده و تصویر اصلی در حوزه فرکانس را بدست آورده و نمایش میدهیم . همچنین این تفاضل را به حوزه مکان برده و نمایش میدهیم .

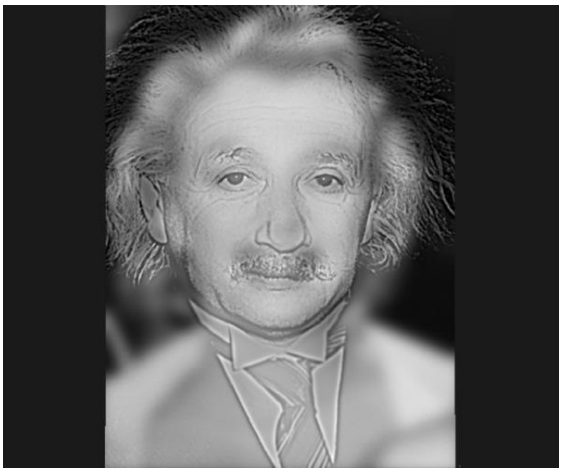


Figure 34 - پهنای باند 300

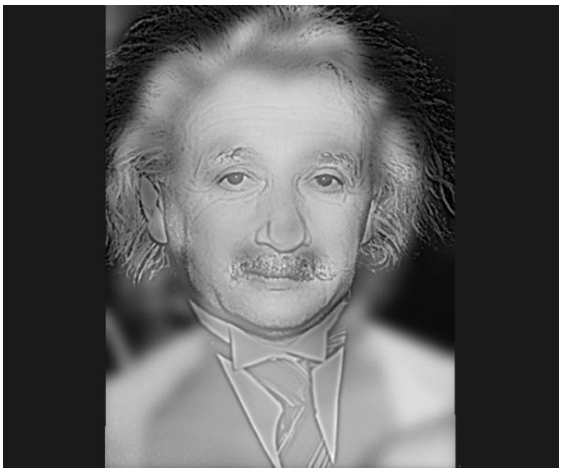


Figure 35 - پهنای باند 500

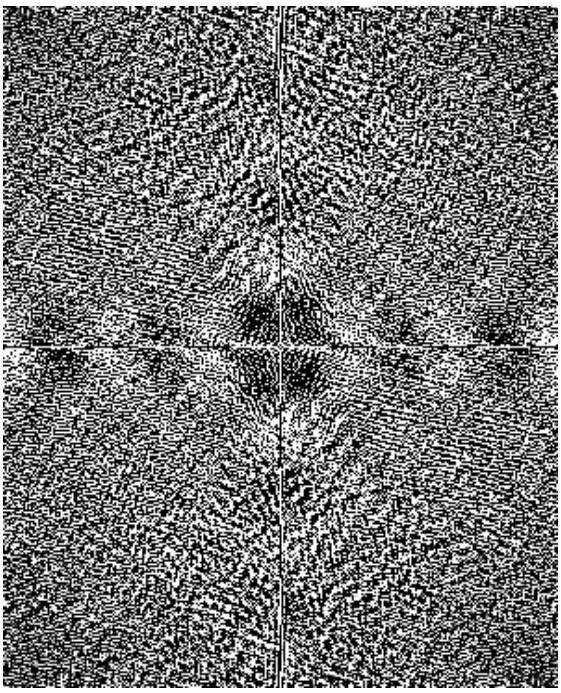


Figure 36 - تفاضل فیلتر پایین گذراعمال شده بر تصویر Im421 در حوزه فرکانس

از این تصاویر نتیجه میگیریم در فرکانس های پایین تر تصویر مرلین مونرو قابل مشاهده است اما هرچه بیشتر اجازه بدهیم فرکانس های بالا رد بشوند آن تصویر محو در تصویر انیشتین میشود .

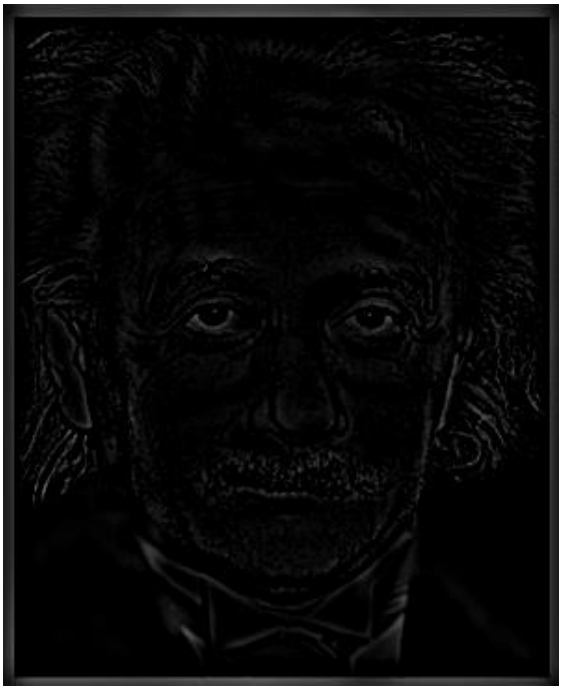


Figure 37 - تفاضل فیتر پایین گذر اعمال شده بر تصویر Im421 در حوزه مکان

همانطور که انتظار میرفت این تفاضل به ما فرکانس های بالا (لبه های تصویر) را میدهد که در اینجا لبه های تصویر انیشتین دیده میشود.

برای تصویر Im423 نیز همین کار را میکنیم :

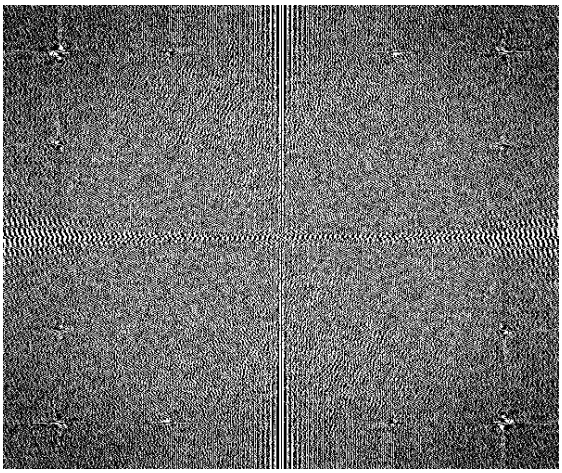


Figure 38- تفاضل اعمال فیلتر پایین گذر بر im423 در حوزه فرکانس

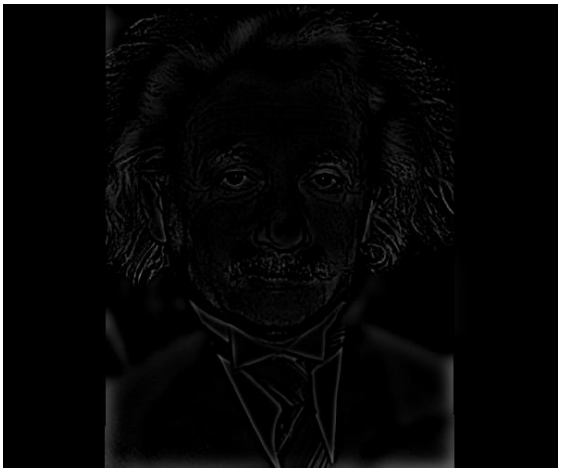


Figure 39- تفاضل اعمال فیلتر پایین گذر بر تصویر Im423 در حوزه مکان

از این دو تصویر نیز نتیجه میگیریم که این تفاضل در واقع فقط فرکانس های بالا یا لبه ها حضور دارند و وقتی آن را به حوزه مکان برمیگردانیم تصویر انیشتین را میبینیم .

نتایج قسمت 5.4 :

تصویر اصلی face1 :

Figure 40- تصویر اصلی face1

تصویر اصلی face2 :

Figure 41- تصویر اصلی face2

بر روی face1 فیلتر پایین گذرگوسی با پهنای باند 50 اعمال میکنیم و تصویر زیر بدست میاید .

Figure 42- face1 پس از اعمال فیلتر پایین گذر

سپس بر روی face2 فیلتر بالاگذر گوسی با پهنای باند 50 اعمال میکنیم و تصویر زیر بدست میاید .

سپس 0.7 تصویر فیلتر شده پایین گذر را با 1.2 تصویر فیلتر شده بالاگذر جمع میکنیم و تصویر زیر بدست میاید .

Figure 43- تصویر ترکیب شده Face1 وface2

اگرچه تصاویر دارای scale های متفاوت بودند و دقیق روی هم نیافتادند اما باز هم این تصویر ترکیب شده به خوبی قابل مشاهده است و وقتی که تصویر را از نزدیک نگاه میکنیم فرکانس های بالا یعنی face2 را میبینیم ولی وقتی که از دور نگاه میکنیم فرکانس های پایین یعنی face1 را میبینیم .

4- پیوست (کد برنامه)

```
[ ]  # importing images
     import cv2 as cv
     import numpy as np
     from matplotlib import pyplot as plt
     img_im184 = cv.imread('Im184.jpg',0)
     img_im183= cv.imread('Im183.jpg',0)
```

Figure 44-imports

```
def get_fft(image):
  return  np.fft.fft2(image)
```

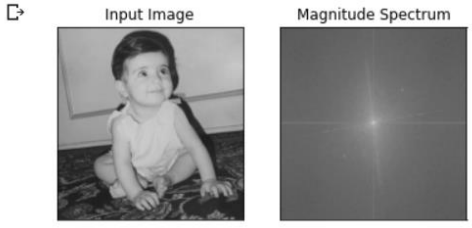
```
def convert_to_magnitude(image):
  f = np.fft.fft2(image)
  fshift = np.fft.fftshift(f)
  magnitude_spectrum = 20*np.log(np.abs(fshift))
  plt.subplot(121),plt.imshow(image, cmap = 'gray')
  plt.title('Input Image'), plt.xticks([]), plt.yticks([])
  plt.subplot(122),plt.imshow(magnitude_spectrum, cmap = 'gray')
  plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])
  plt.show()
  return magnitude_spectrum
```

```
def convert_to_phase(image):

  dft = np.fft.fft2(image)
  dft_shift = np.fft.fftshift(dft)
  phase_spectrum = np.angle(dft_shift)
  ax1 = plt.subplot(1,2,1)
  ax1.imshow(image, cmap='gray')
  ax2 = plt.subplot(1,2,2)
  ax2.imshow(phase_spectrum, cmap='gray')
  plt.show()
```

Figure45 -convert to magnitude and phase functions

```
im183_magnitude=convert_to_magnitude(img_im183)
```




```
[ ]  im184_phase=convert_to_phase(img_im183)
```

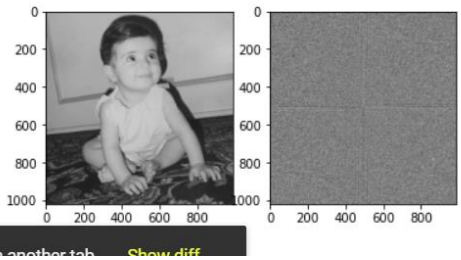


Figure46 - converting child image to phase and magnitude

```
[ ]  def combine(magnitude_spectrum,phase_spectrum):
       combined = np.multiply(np.abs(magnitude_spectrum), np.exp(1j*np.angle(phase_spectrum)))
       imgCombined = np.real(np.fft.ifft2(combined))
       return imgCombined
```

```
get_fft(img_im183).shape
```

```
combine_im183phase_im184magnitude=combine(get_fft(img_im184),get_fft(img_im183))
plt.imshow(combine_im183phase_im184magnitude, cmap = 'gray')
```

+ Code + Text

```
combine_im183magnitude_im184phase=combine(get_fft(img_im183),get_fft(img_im184))
plt.imshow(combine_im183magnitude_im184phase, cmap = 'gray')
```

```
[ ]  combine_im183magnitude_im183phase=combine(get_fft(img_im184),get_fft(img_im183))
     plt.imshow(combine_im183magnitude_im183phase, cmap = 'gray')
```

Figure47 -function for combining phase and magnitude and its usage

```
[ ]  def distance(p1,p2):
       return np.sqrt((p1[0]-p2[0])**2 +(p1[1]-p2[1])**2)
```

```
[ ]  def guassianLP(D0,image_shape):
       base=np.zeros(image_shape[:2])
       rows,cols=image_shape[:2]
       center=(rows/2,cols/2)
       for x in range(cols):
         for y in range(rows):
           base[y,x]=np.exp(((-distance((y,x),center)**2)/(2*(D0**2))))

       return base
```

```
def apply_GLPF_to_image(image,D0):
  orginal = np.fft.fft2(image)
  center=np.fft.fftshift(orginal)
  low_pass_center=center* guassianLP(D0,image.shape)
  low_pass=np.fft.ifftshift(low_pass_center)
  inverse_low_pass=np.fft.ifft2(low_pass)
  #cv2_imshow(inverse_low_pass)
  return inverse_low_pass
```

```
[ ]  img_im421=cv.imread("Im421.jpg",0)
     img_im423=cv.imread("Im423.jpg",0)
```

Figure48 gussian low pass filter and applying it on image functions

```
[ ]  from google.colab.patches import cv2_imshow
```

```
[ ]  im421_bandwidth10=apply_GLPF_to_image(img_im421,10)
     im421_bandwidth20=apply_GLPF_to_image(img_im421,20)
     im421_bandwidth40=apply_GLPF_to_image(img_im421,40)
     im421_bandwidth50=apply_GLPF_to_image(img_im421,50)
     im421_bandwidth60=apply_GLPF_to_image(img_im421,60)
     im421_bandwidth80=apply_GLPF_to_image(img_im421,80)
     im421_bandwidth100=apply_GLPF_to_image(img_im421,100)
     im421_bandwidth200=apply_GLPF_to_image(img_im421,200)
     im421_bandwidth300=apply_GLPF_to_image(img_im421,300)
     im421_bandwidth500=apply_GLPF_to_image(img_im421,500)
```

```
     cv2_imshow(im421_bandwidth500)
```

Figure 49 using 10 differenct bandwidth for glp on im421

```
]  im423_bandwidth10=apply_GLPF_to_image(img_im423,10)
   im423_bandwidth20=apply_GLPF_to_image(img_im423,20)
   im423_bandwidth40=apply_GLPF_to_image(img_im423,40)
   im423_bandwidth50=apply_GLPF_to_image(img_im423,50)
   im423_bandwidth60=apply_GLPF_to_image(img_im423,60)
   im423_bandwidth80=apply_GLPF_to_image(img_im423,80)
   im423_bandwidth100=apply_GLPF_to_image(img_im423,100)
   im423_bandwidth200=apply_GLPF_to_image(img_im423,200)
   im423_bandwidth300=apply_GLPF_to_image(img_im423,300)
   im423_bandwidth500=apply_GLPF_to_image(img_im423,500)
```

```
   cv2_imshow(im423_bandwidth500)
```

```
/usr/local/lib/python3.7/dist-packages/google/colab/patche
  a = a.clip(0, 255).astype('uint8')
```

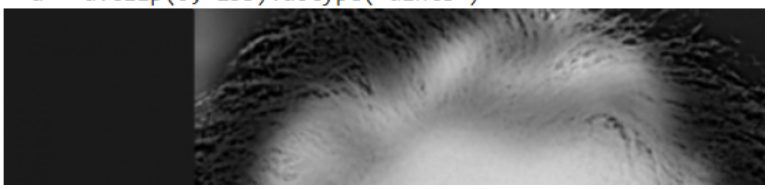


Figure50 using 10different bandwidth and applying them on im423

```python
def part3ofquestion5(image,D0):
  orginal = np.fft.fft2(image)
  center=np.fft.fftshift(orginal)
  low_pass_center=center* guassianLP(10,image.shape)
  remaining=center-low_pass_center # baghimoonde dar freq domain
  low_pass_remaining=np.fft.ifftshift(remaining)
  inverse_low_pass_remaining=np.fft.ifft2(low_pass_remaining) # baghimoonde dar spatial domain
  low_pass=np.fft.ifftshift(low_pass_center)
  inverse_low_pass=np.fft.ifft2(low_pass) #pas az emal filter too spatial domain

  cv2_imshow(remaining)
  cv2_imshow(inverse_low_pass_remaining)
```

```python
[ ] part3ofquestion5(img_im423,100)
```

Figure51 A function to solve problem in part 3 of question 5

```python
[ ] def gaussianHP(D0,image_shape):
      base=np.zeros(image_shape[:2])
      rows,cols=image_shape[:2]
      center=(rows/2,cols/2)
      for x in range(cols):
        for y in range(rows):
          base[y,x]=1-np.exp(((-distance((y,x),center)**2)/(2*(D0**2))))

      return base
```

```python
[ ] def apply_GHPF_to_image(image,D0):
      orginal = np.fft.fft2(image)
      center=np.fft.fftshift(orginal)
      low_pass_center=center* gaussianHP(D0,image.shape)
      low_pass=np.fft.ifftshift(low_pass_center)
      inverse_low_pass=np.fft.ifft2(low_pass)
      #cv2_imshow(inverse_low_pass)
      return inverse_low_pass
```

```python
face1=cv.imread("face1.jpg",0)
face2=cv.imread("face2.jpg",0)
image_with_low_freq=apply_GLPF_to_image(face1,50)
image_with_high_freq=apply_GHPF_to_image(face2,50)
```

Figure52 guassian high pass filter and applying it on image functions

```
[ ]  mixed_image=image_with_low_freq*.7 + image_with_high_freq*1.3
```

```
cv2_imshow(mixed_image)
```

```
/usr/local/lib/python3.7/dist-packages/google/colab/patches/__init__.py:22: ComplexWarni
  a = a.clip(0, 255).astype('uint8')
```

Figure53 mixing two filtered images together with some exprimental coefs

مراجع

R. Boyle and R. Thomas *Computer Vision: A First Course*, Blackwell Scientific Publications, 1988, pp 32 - 34.

E. Davies *Machine Vision: Theory, Algorithms and Practicalities*, Academic Press, 1990, Chap. 3.

D. Vernon *Machine Vision*, Prentice-Hall, 1991, Chap. 4.

گزارش تمارین

سروش باقرنژاد

اطلاعات گزارش	چکیده
تاریخ:	مدل های رنگی میتوانند بر پایه فیزیک یا ادراک انسان ساخته شوند . تعریف های فیزیکی به دو صورت جمع شونده (ترکیبی از نور ها مانند RGB) یا تفریق شونده (حذف برخی از نور ها مانند CMYK) و مدل های ساخته شده از روی ادراک انسان بر پایه یک سری ازمایشات تجربی روی انسان ها است .
واژگان کلیدی: رنگ فضا های رنگی RGB sRGB CIE CIELAB	

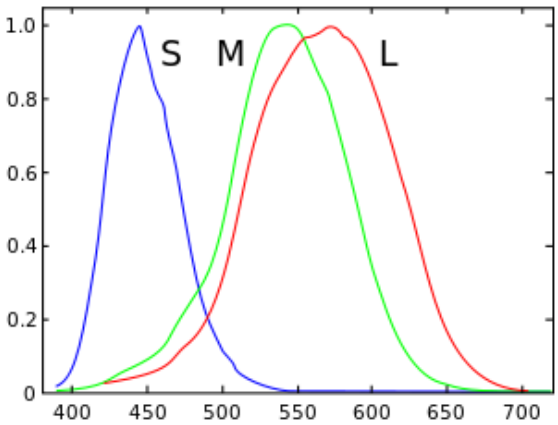
1-مقدمه

در سوال 7 از ما خواسته شده تا درباره ی سه فضای رنگ به غیر از فضاهایی که در کتاب آورده شده تحقیق کنیم و کاربرد های آن ها و ارتباط آن ها با RGB و یا HIS را بنویسیم .

2-شرح

فضای رنگی CIE 1931 :

این فضای رنگی اولین تلاش برای ربط دادن عددی بسط امواج الکترو مغناطیس در طیف مریی بود . این سیستم در سال 1931 توسط *"Commission Internationale de l'éclairage"* *پس از انجام آزمایشات مختلف روی ادراک بینایی انسان طراحی شد.*
چشم یک انسان نرمال، سه نوع سلول مخروطی شکل برای حس کردن نور دارد . هر کدام از این نوع سلول ها در طول موج خاصی به بالاترین میزان حساسیت خود میرسند که در نمودار زیر میتوان آن را دید .



The normalized spectral sensitivity of -1 Figure human cone cells of short-, middle- and long-wavelength types.

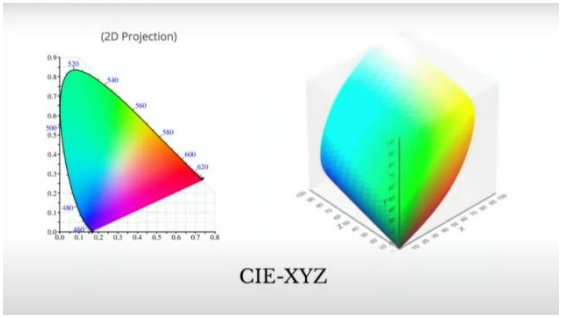
این سلول ها درک مارا نسبت به رنگ ها در روشنایی بالا معلوم میکنند . در نتیجه با کمک سه عدد که سلول های

مخروطی شکل به ما میدهند میتوانیم امواج مریی الکترومغناطیس را درک کنیم .

در سال 1931 پس از نیاز شدید به استانداردی جهت نمایش رنگ ها john guild مدل RGB و در همان زمان Irwin priest مدل خود را که بر پایه HIS بود را ارایه داد و پس از بحث پیرامون مدل استاندارد برای رنگ ها به مدل CIE رسیدند .

فضای رنگی CIE XYZ همه ی رنگ هایی که برای یک انسان معمولی وجود دارد را در برمیگیرد .

در این مدل سه بعدی Y اشاره به روشنایی ، Z برابر با ابی در مدل CIE RGB و X نمایانگر ترکیبی از سه رنگ CIE RGB هست که مقدار نامنفی دارند .



-CIE XYZ2 Figure

شکل بالا فضای رنگی CIE XYZ است که برای راحت تر نشان دادن آن را به نموداری 2 بعدی تصویر میکنیم.

$$x = \frac{X}{X+Y+Z}$$
$$y = \frac{Y}{X+Y+Z}$$
$$z = \frac{Z}{X+Y+Z} = 1-x-y$$

x,y,z -3 Figure روابط برقرار برای

در هنگام تصویر کردن بر روی دو بعد از محور Y که روشنایی را مشخص میکرد صرف نظر میکنیم .

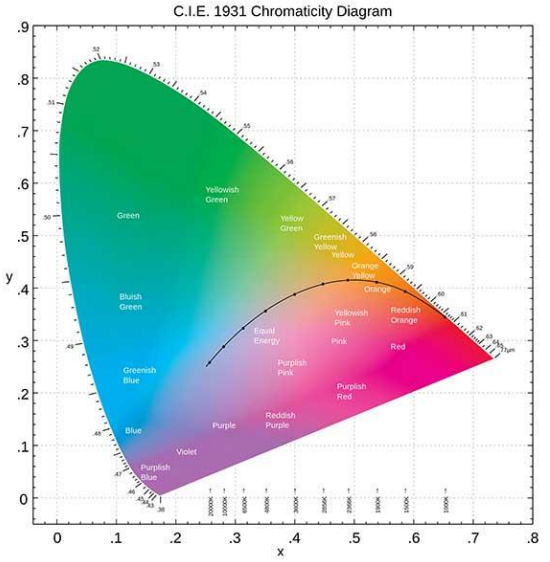
همچنین با روابط زیر میتوان به مقادیر سه گانه قبلی برگشت :

$$X = \frac{Y}{y}x,$$
$$Z = \frac{Y}{y}(1-x-y).$$

4 Figure

فضای رنگی CIE XYZ در واقع مهم ترین فضای رنگی است و اگر بخواهیم فضای رنگی جدیدی بسازیم زیر مجموعه ای از این فضا را جدا میکنیم . اگر بتوانیم از فضای رنگی ساخته شده به فضای CIE XYZ برسیم و برعکس اگر از CIE XYZ به آن فضا در طی تبدیلاتی برسیم میفهمیم که فضای رنگی ساخته شده معتبر است .

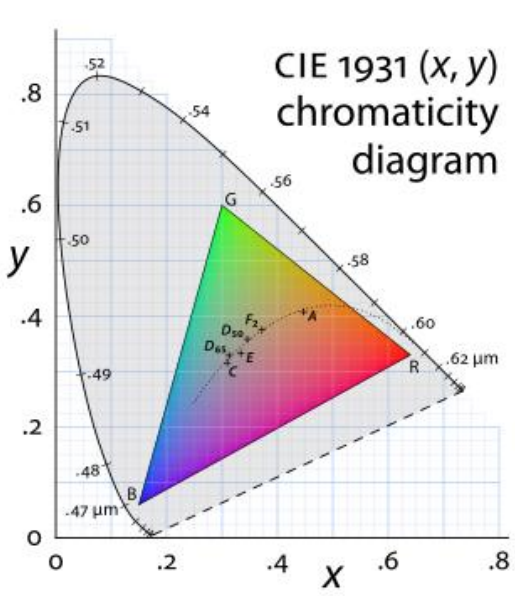
ویژگی این فضای رنگی این است که همه ی رنگ های قابل مشاهده انسان را در بر دارد .



CIE نمودار رنگی -5 Figure

همانطور که بالاتر گفته شد ، قبل این مدل، زمانی که تکنولوژی نبود تا طول موج ها را اندازه بگیرند john guild با ایده ترکیب سه رنگ اصلی که ما در دنیای واقعی داریم و با ترکیب این سه رنگ رنگ های دیگر را تولید میکنیم ، مدل RGB را ارایه داد .

مدل RGB زیر مجموعه ای از مدل CIE بود .در شکل صفحه بعد تصویر فضای رنگی RGB را در تصویر شده ی فضای رنگی CIE مشاهده میکنید .



در مورد این فضا به تفصیل در کتاب بحث شده و در اینجا بحثی در مورد آن نمیکنیم.

با گذشت زمان و فراگیر شدن مانیتور های crt ، به فکر این افتادند که فضای رنگی استانداردی برای تولید رنگ با سه رنگ اصلی داشته باشند و اینگونه بود که فضای رنگی sRGB که مخفف Standard RGB است را معرفی کردند.

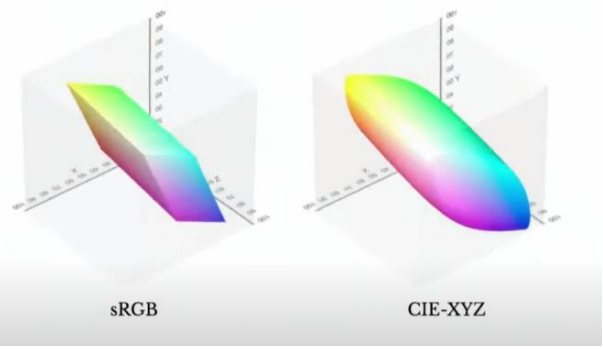


Figure 6- فضای sRGB در مقابلCIE XYZ

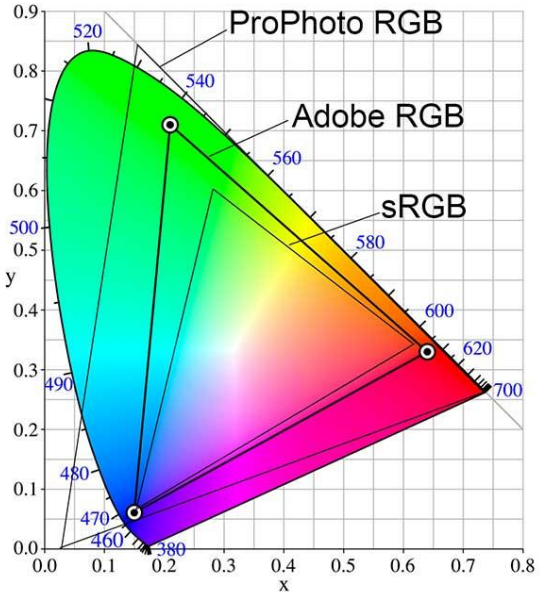


Figure 7-مقایسه فضاهای رنگی مختلف

در این مانیتور ها رنگ ها به وسیله فسفر هایی که دارای سه رنگ اصلی سبز و ابی و قرمز بودند به وجود می آمدند. مشکلی که در این مدل وجود داشت این بود که برای چشم انسان تفکیک شدت های تیره رنگ بهتر صورت میگیرد تا تمییز دادن شدت های روشن رنگ، برای حل این مشکل تابعی غیر خطی پیشنهاد شد تا اصطلاحا رنگ ها را در فضای رنگی پخش کند اما چون تابع ما غیر خطی بود باعث میشد تا دیگر ترکیب دو رنگ در دنیا دیجیتال بر خلاف انتظار ما در دنیای واقعی عمل کند .یعنی ما اگر دو رنگ قرمز و ابی را در دنیای واقعی ترکیب کنیم و بنفش را بدست بیاوریم دیگر چنین چیزی پس از اعمال تابع تبدیل gamma curve بدست نمیاوریم . هر چند رنگ بدست امده بسیار نزدیک به رنگ دنیای واقعی است .

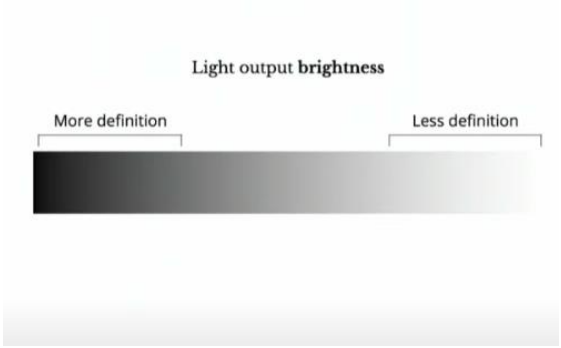
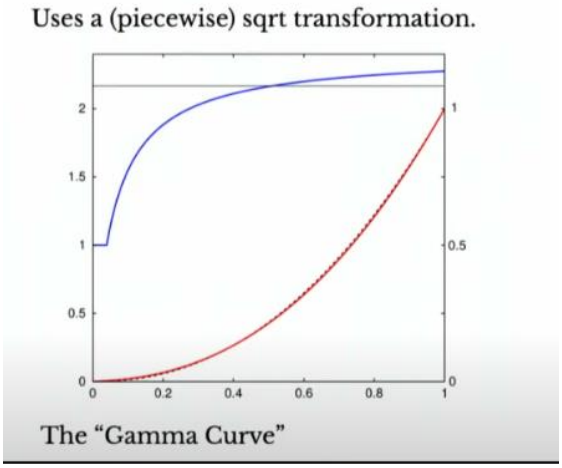


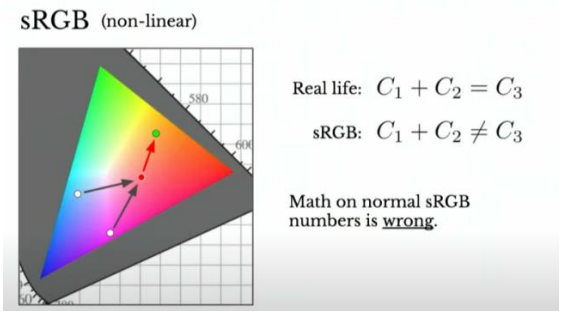
Figure 8- درک جزییات در شدت نور های مختلف

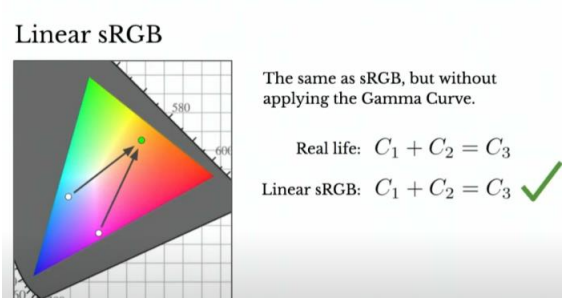
همانطور که در تصویر بالا مشاهده میشود، ما در ناحیه های روشن توانایی تشخیص کمتری داریم پس پژوهشگران به این فکر افتادند که رنگ را به کمک تابع زیر (sqrt transformation) پخش کنند .



gamma curve -9 Figure

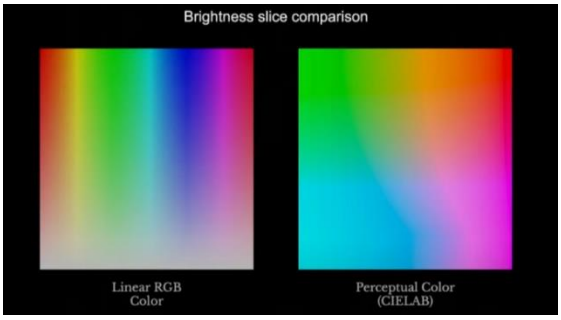
برای حل این مشکل دیگر از gamma curve استفاده نکردند و تبدیل را خطی انجام دادند تا به مدل sRGB linear رسیدند.





linear sRGB-10 Figure

اما اینکه فضای رنگی ما همیشه از نظر ریاضی مطابق با دنیای واقعی باشد برای ما ملاک نیست و گاهی لازم است تا از لحاظ ادراکی فضای رنگی یک دست و یکنواختی داشته باشیم. این نیاز باعث شد تا مدلCIELAB بوجود بیاید .



CIELAB-11 Figure در مقابلlinear sRGB

فضای رنگیCIELAB فضایی یکنواخت است که در آن به طور عمد از عملگر غیرخطی استفاده شده تا آن را شبیه به چیزی کند که انسان انتظار دیدنش را دارد .

یکی از کاربرد های مهم CIELAB این است که در هنگام استفاده از گرادیان برای رنگ ها ، رنگ های smooth تری بوجود میاورد در صورتی که در فضای RGB ما هنگام استفاده از گرادیان شاهد حضور رنگ مشکی هستیم .

کاربرد دیگر این فضای رنگ عملیات desaturating برای تصاویر است . همانطور که در شکل زیر مشاهده میشود این عملیات در فضای رنگی CIELAB بسیار بهتر از sRGB انجام میشود .

CIELAB VS sRGB in desaturating-12 Figure

3-بحث و نتایج

در این مقاله سه فضای رنگی sRGB و linear sRGB و CIRELAB معرفی شد .در این بخش به جمع بندی و کاربرد های هر یک میپردازیم و سپس فضاهای رنگی دیگر را فقط برای اشنایی بیشتر نام میبریم .

ما زمانی از مدل linear sRGB استفاده میکنیم که بخواهیم رفتار دنیای واقعی را شبیه سازی کنیم . کاربرد این مدل در CG rendering میباشد.

اگر بخواهیم رفتار و ادراک انسانی را در زمینه ی رنگ ها بازسازی کنیم از مدل CIELAB استفاده میکنیم که در حوزه های data visualization و color interpolation و ... کاربرد دارد . همچنین اگر بخواهیم مدلی استاندارد و همگانی برای انتقال و تبدیل در دستگاه های مختلف و یا FILE I/O از sRGB استفاده میکنیم.

به غیر از مدل های اشاره شده در بالا فضاهای رنگی بسیار بیشتری نیز وجود دارند که میتوان به واریانت های CIE مثل CIEUVW و یا CIELUV و یا واریانت های RGB مثل Adobe RGB و Rec .2000 و ... اشاره کرد .

همچنین فضاهای رنگی ویژه و تجاری ای نیز وجود دارند که از معروف ترین آن ها میتوان به **Pantone LLC** اشاره کرد که در این فضا هر رنگ اسم مخصوص به خود را دارد و این شرکت هر ساله رنگ سال را تعیین میکند که در صنعت فشن و دیجیتال بسیار کاربرد دارد .

مراجع

John Austin *A Stranger Gravity: Strange loop conference,2019*

R. Boyle and R. Thomas *Computer Vision: A First Course*, Blackwell Scientific Publications, 1988, pp 32 - 34.

E. Davies *Machine Vision: Theory, Algorithms and Practicalities*, Academic Press, 1990, Chap. 3.

D. Vernon *Machine Vision*, Prentice-Hall, 1991, Chap. 4.