دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

تمرین سوم درس پردازش تصویر

دکتر رحمتی

غلامرضا دار ۴۰۰۱۳۱۰۱۸

بهار ۱۴۰۱

فهرست مطالب

سوال ۱)

I) آشنایی با Discrete Cosine Transform

a) با توجه به رابطه و تصویر داده شده، $F(0.0)$ را محاسبه میکنیم. با جایگذاری ۰و۰ بجای $u.v$ خواهیم داشت:

$$F(0.0) = \frac{2 \times \frac{\sqrt{2}}{2} \times \frac{\sqrt{2}}{2}}{\sqrt{8 \times 8}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} 1 \times 1 \times f(i.j)$$

که برابر با جمع تمام پیکسل های تصویر می‌باشد.

$$F(0.0) = \frac{1}{8} \times 5440 = \mathbf{680}$$

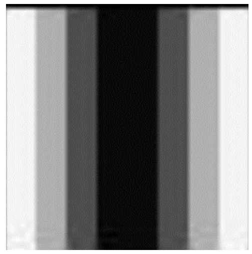
b) DC Coefficient در واقع ضریب سینوس و کسینوس‌های با فرکانس صفر است و به نوعی به عنوان یک Offset یا Bias عمل میکند. دلیل اینکه به این ضریب DC میگویند تاریخی است و به رشته برق برمی‌گردد که در آنجا DC به معنی جریان مستقیم است. ضرایب پشت سینوس و کسینوس ضرایب AC نام دارند که نشان می‌دهد این ضرایب Alternating Current یا جریان متناوب را کنترل می‌کنند.

c) طبق رابطه زیر داریم:

$$F(0.2) = \frac{2 \times \frac{\sqrt{2}}{2} \times \frac{\sqrt{2}}{2}}{\sqrt{8 \times 8}} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} 1 \times \frac{\cos(2j+1)\, v\pi}{2N} \times f(i.j)$$

همانطور که مشخص است، این عبارت، تعدادی Cos در جهت v (افقی) را نشان می‌دهد. با عدد گذاری در رابطه بالا متوجه می‌شویم که این عبارت دقیقا برابر با بخش زیر است:

0	1	2	3	4	5	6	7
0.92	0.38	-0.38	-0.92	-0.92	-0.38	0.38	0.92



این بخش از DCT وظیفه تولید لبه های عمودی با فرکانس نه چندان بالا را دارد.

II) در این قسمت از خواص تبدیل فوریه استفاده میکنیم.

d) $\mathscr{F}\{f(x-3)\} = e^{-3(2\pi)ju}F(u)$

e) $\mathscr{F}\{f(x)f(x)\} = F(u) \star F(u)$

f) $\mathscr{F}\{f(-x)\} = \frac{1}{|-1|}F\left(\frac{u}{-1}\right)$

g) $\mathscr{F}\{f(4x-3)\} = \mathscr{F}\left\{f\left(\frac{x-\frac{3}{4}}{\frac{1}{4}}\right)\right\} = \frac{1}{|4|}e^{-\frac{3}{4}ju}F\left(\frac{u}{4}\right)$

h) $\mathscr{F}\left\{\mathscr{F}\{f(x)\}\right\} = f(-x)$

i) بسم الله الرحمن الرحیم میریم که داشته باشیم جا گذاری در رابطه DFT: (هر رنگ نشان دهنده جملات مربوط به یک سطر است)

$$F(u.v) =$$

$$33 + 23\cos\left(-\frac{\pi v}{2}\right) + jsin\left(-\frac{\pi v}{2}\right) + 13\cos(-\pi v) + jsin(-\pi v)$$

$$+3\cos\left(-\frac{3\pi v}{2}\right) + jsin\left(-\frac{3\pi v}{2}\right)$$

$$+32\cos\left(-\frac{\pi u}{2}\right) + jsin\left(-\frac{\pi u}{2}\right) + 22\cos\left(-\frac{\pi u}{2} - \frac{\pi v}{2}\right) + jsin\left(-\frac{\pi u}{2} - \frac{\pi v}{2}\right) + 12\cos\left(-\frac{\pi u}{2} - \pi v\right)$$

$$+jsin\left(-\frac{\pi u}{2} - \pi v\right) + 2\cos\left(-\frac{\pi u}{2} - \frac{3\pi v}{2}\right) + jsin\left(-\frac{\pi u}{2} - \frac{3\pi v}{2}\right)$$

$$+31\cos(-\pi u) + jsin(-\pi u) + 21\cos\left(-\pi u - \frac{\pi v}{2}\right) + jsin\left(-\pi u - \frac{\pi v}{2}\right) + 11\cos(-\pi u - \pi v)$$

$$+jsin(-\pi u - \pi v) + 1\cos\left(-\pi u - \frac{3\pi v}{2}\right) + jsin\left(-\pi u - \frac{3\pi v}{2}\right)$$

$$+30\cos\left(-\frac{3\pi u}{2}\right) + jsin\left(-\frac{3\pi u}{2}\right) + 20\cos\left(-\frac{3\pi u}{2} - \frac{\pi v}{2}\right) + jsin\left(-\pi u - \frac{\pi v}{2}\right) + 10\cos\left(-\frac{3\pi u}{2} - \pi v\right)$$

$$+jsin\left(-\frac{3\pi u}{2} - \pi v\right) + 0$$

حال باید u,v های مختلف بین ۰ تا ۳ را در معادله بالا قرار دهیم و به ازای هر جفت u,v یک درایه از $F(u.v)$ به دست می‌آید.

$\frac{1}{16}$

264+0j	80-80j	80+0j	80+80j
8-8j	0	0	0
8+0j	0	0	0
8+8j	0	0	0

j) مانند بخش قبل عمل میکنیم منتهی اول ماتریس داده شده را در -1^{x+y} ضرب میکنیم.

$\frac{1}{16}$

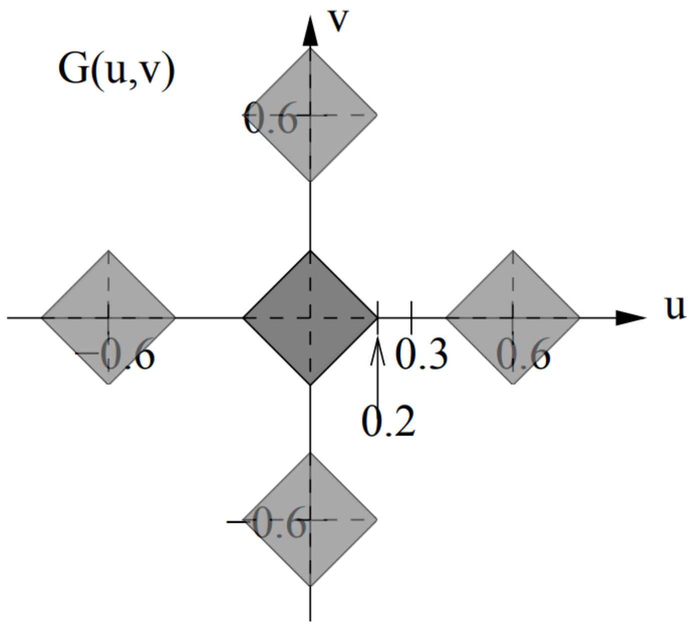
0	0	8+0j	0
0	0	8+8j	0
80+0j	80+80j	264+0j	80-80j
0	0	8-8j	0

$k)$ می‌دانیم ضرب کردن ورودی در -1^{x+y} معادل شیفت دادن تبدیل فوریه به نحوی است که فرکانس صفر به مرکز بیاید. اثبات این اتفاق در کتاب موجود است، کافی است این عبارت را در رابطه تبدیل فوریه جایگزین کنید و خواهید دید که تمام مقادیر به میزان نصف اندازه کل تصویر شیفت میخورند.

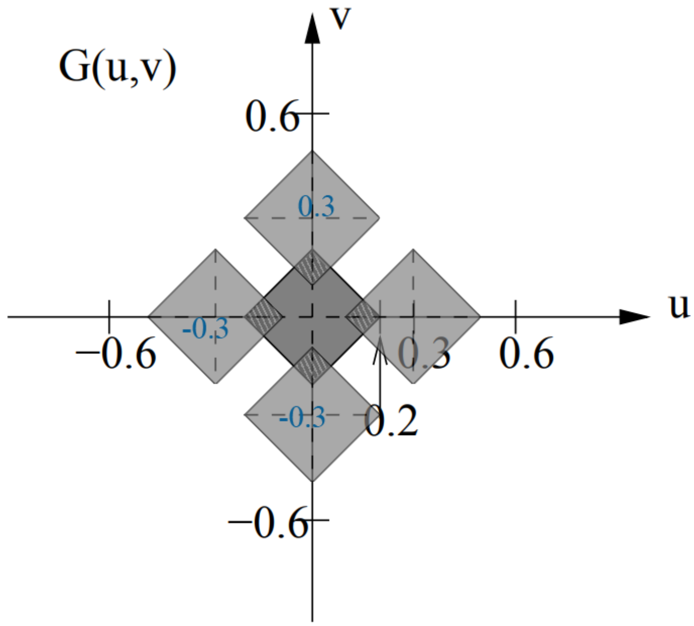
$l)$ مانند بخش *i* (با این تفاوت که فرمول *IFT* در *Cos* و *Sin* ها منفی ندارد تقسیم بر *MN* هم ندارد).

3+0j	*1+0j*	*3+0j*
2+3j	*-6+1j*	*2+3j*
-3+0j	*-1+0j*	*-3+0j*

$m)$ از آنجایی که *Sampling Distance* برابر با $\frac{1}{0.6}$ است، پس، فرکانس نمونه برداری برابر با *0.6* است.



n) مشابه بخش قبل داریم:



o) $h(x.y)$ دارای *Aliasing* است. $g(x.y)$ دارای *Aliasing* نیست. برای اینکه یک *Sampling* بدون *Aliasing* داشته باشیم باید فرکانس-های نمونه برداری ما از روابط زیر پیروی کنند. همچنین *Aliasing* در تصویر بالا با هاشور نشان داده شده است.

$$\frac{1}{\Delta_x} \geq 2 \times u_{max} = 0.4$$

$$\frac{1}{\Delta_y} \geq 2 \times v_{max} = 0.4$$

p) اگر $F(u.v)$ را ۴۵ درجه بچرخانیم تبدیل به یک مربع به ضلع $\frac{\sqrt{2}}{5}$ خواهد شد.. حال کافیست طبق رابطه معکوس تبدیل فوریه دو بعدی پیوسته، $v(x.y)$ را محاسبه کنیم.

$$V(u.v) = \begin{cases} 1 & \text{inside the square} \\ 0 & \text{otherwise} \end{cases}$$

$$v(x.y) = \int_{-\frac{\sqrt{2}}{10}}^{\frac{\sqrt{2}}{10}} e^{j2\pi ux}\, du \int_{-\frac{\sqrt{2}}{10}}^{\frac{\sqrt{2}}{10}} e^{j2\pi vy} dv$$

کافیست انتگرال ها را حساب کنیم. سپس، تابعی بر حسب $x.y$ بدست خواهد آمد.

q) مانند بخش o داریم:

$$\frac{1}{\Delta_x} \geq 2 \times u_{max} = \frac{2\sqrt{2}}{10}$$

$$\frac{1}{\Delta_y} \geq 2 \times v_{max} = \frac{2\sqrt{2}}{10}$$

سوال ۲)

a) با استفاده از رابطه تبدیل فوریه دو بعدی گسسته خواهیم داشت:

$$H_1(u.v) = 0 - e^{-j2\pi u} + 0 - e^{-j2\pi v} + 5e^0 - e^{j2\pi v} + 0 + e^{j2\pi u} + 0 =$$
$$5 - 2\cos(2\pi u) - 2\cos(2\pi v)$$

برای H2 میتوانیم ابتدا آن را جدا کنیم (چون جدایی‌پذیر است) و سپس فوریه یک بعدی را برای هر بعد محاسبه کنیم و سپس ضرب

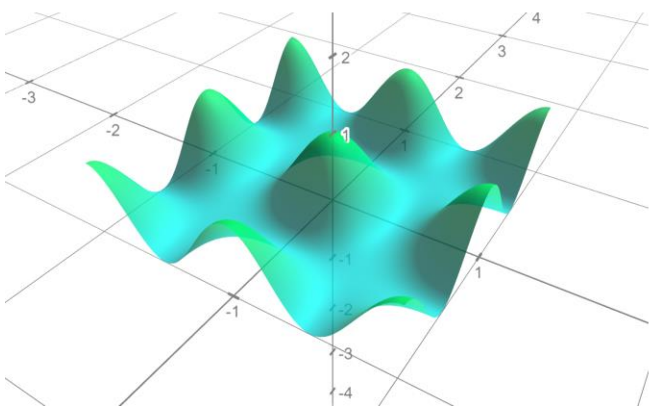
$$H_2(u) = \ 1 \times e^{j2\pi u} + 2 + 1 \times e^{-j2\pi u} = 2 + 2\cos(2\pi u)$$
$$H_2(v) = \ 1 \times e^{j2\pi v} + 2 + 1 \times e^{-j2\pi v} = 2 + 2\cos(2\pi v)$$
$$H_2(u.v) = H_2(u)H_2(v) = (2 + 2\cos(2\pi u)) \times (2 + 2\cos(2\pi v))$$

b) واضح است که فیلتر H1 یک فیلتر High-pass بعلاوه خود تصویر است.

$$H_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}$$

بنابراین یک فیلتر **بالاگذر** است.

فیلتر H2 نیز مشخصا یک تابع هموار کننده در نتیجه **پایین گذر** است. نمودار تابع بدست آمده در بخش قبل حاکی از این اتفاق است.



c) در این قسمت پس از گرفتن IDFT، u,v های مختلف در رنج ۰ تا ۲ را (جمعا ۹ حالت) درتابع بدست آمده می‌گذاریم و ماتریس های زیر را تولید میکنیم. IDFT برای H1 تابع ثابت ۰ را نتیجه می‌دهد.

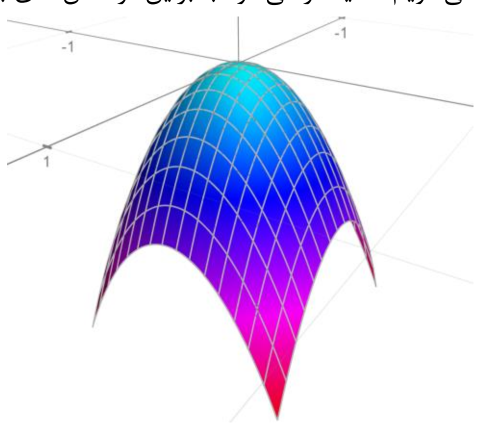
0	0	0
0	0	0
0	0	0

0	$-\frac{\sqrt{3}}{4}j$	$\frac{\sqrt{3}}{4}j$
$\frac{\sqrt{3}}{4}j$	0	$\frac{\sqrt{3}}{2}j$
$-\frac{\sqrt{3}}{4}j$	$-\frac{\sqrt{3}}{2}j$	0

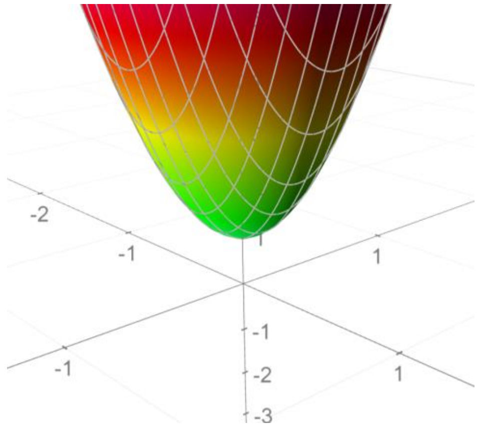
d) فیلتر اول که همواره تصویر را برابر با صفر می‌کند. مجموع درایه های فیلتر دوم صفر می‌شود که به این معنی است که این فیلتر بالاگذر است.

در این بخش با رسم کردن توابع مربوط به فیلترها درمورد Low-pass یا High-pass بودن آنها تصمیم میگیریم.

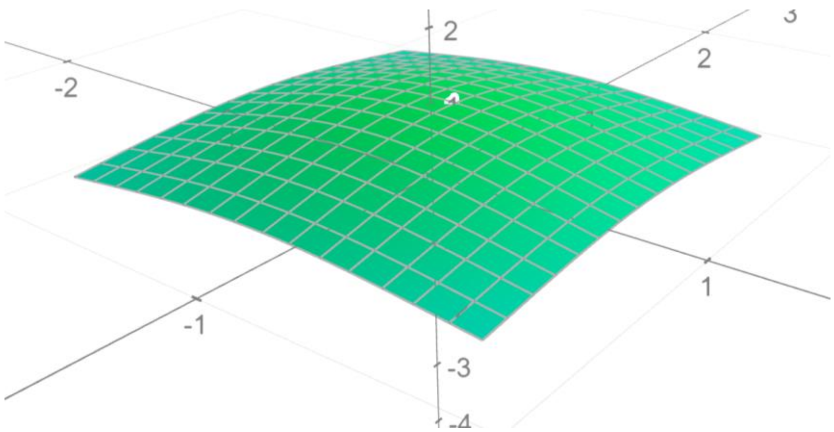
e) **Low-pass** است. همانطور که در شکل زیر مشاهده می‌کنید، این فیلتر، فرکانس های پایین را عبور می‌دهد(پس از شیفت به بالا) و هر چه از مرکز دور می‌شویم ضعیف‌تر می‌شود بنابراین فرکانس های بالا عبور داده نمی‌شوند.



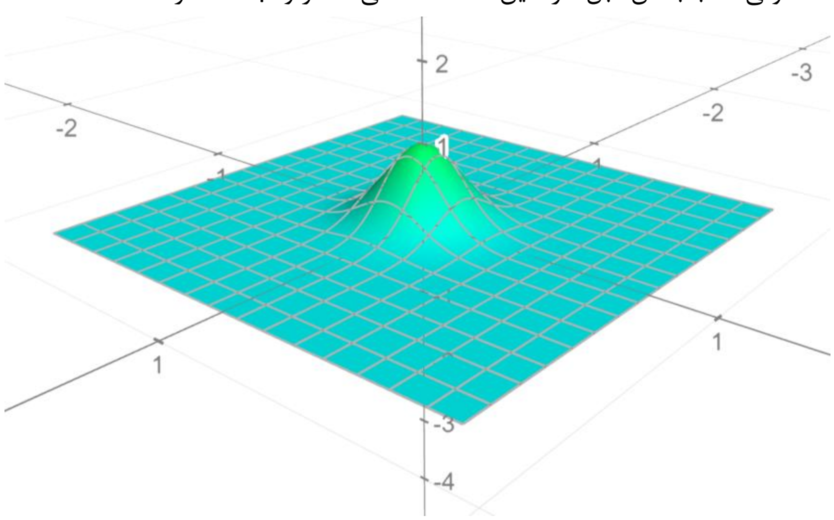
f)   **High-pass** است. بر خلاف بخش قبل فرکانس‌های پایین را عبور نمی‌دهد و فرکانس‌های بالا را عبور می‌دهد.



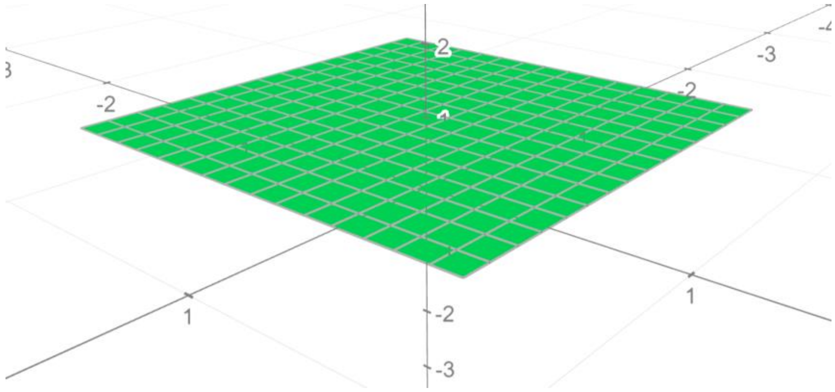
g)   **Low-pass** است. تفاوتی که با بخش قبل دارد این است که کمی ملایم‌تر و Smooth تر است.



h)   **Low-pass** است. تفاوتی که با بخش قبل دارد این است که کمی تندتر و Sharp تر است.

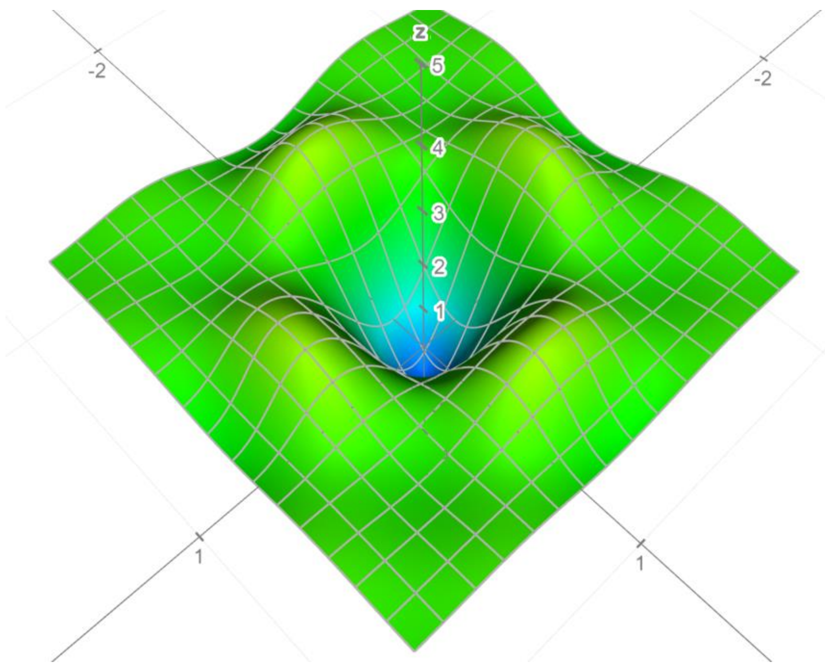


i) همه فرکانس ها را به یک اندازه فیلتر میکند.



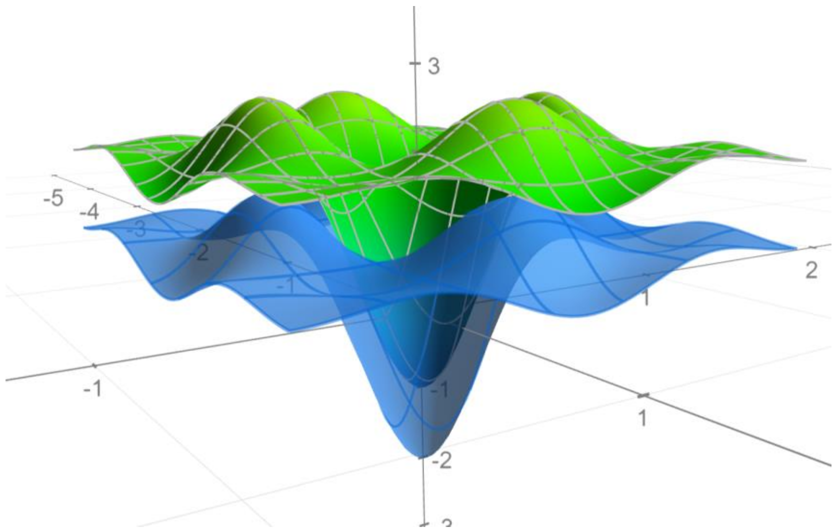
j) اگر دقت کنید متوجه می‌شوید که تابع F_2 در یک تابع دیگر در حوزه فرکانس ضرب شده است و F_1 را پدید آورده است. همچنین می‌دانیم که ضرب در حوزه فرکانس معادل کانولوشن در حوزه مکان است. بنابراین رابطه تصویر $f_1 . f_2$ این است که با کانوالو کردن یک فیلتر خاص(داده شده در سوال) در f_2، f_1 بدست می‌آید.

k) با رسم کردن نمودار فیلتر داده شده متوجه می‌شویم که یک فیلتر High-pass است. بنابراین با اعمال این فیلتر بر روی f_2، نتیجه شارپ‌تر می‌شود. که این به این معنی است که f_1 شارپ تر است.



ا) **High-pass Filtering**

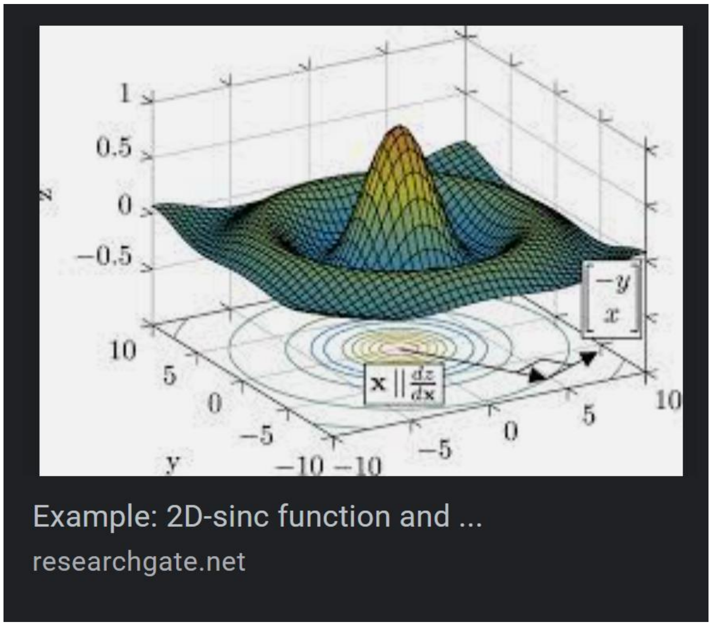
m) با رسم نمودار فیلتر جدید متوجه می‌شویم که این فیلتر نیز High-pass است اما همان فرکانس‌هایی که تابع قبل فیلتر میکرد را با قدرت متفاوتی فیلتر میکند. یعنی فرکانس های پایین را بیشتر کم میکند و فرکانس های بالا را کمتر زیاد میکند در نهایت این فیلتر هم بالاگذر است یعنی تصویر خروجی Sharp تر خواهد شد.



سوال ۳)

a) در این بخش می‌خواهیم خروجی تبدیل فوریه توابع مختلف را از نظر تعدادی خواص بررسی کنیم.

a. می‌دانیم تبدیل فوریه یک مربع، دو تابع $Sinc$ در جهات عمودی و افقی خواهد بود. می‌دانیم توابع $Sinc$ قسمت موهومی ندارند و کاملا حقیقی هستند. بنابراین قسمت موهومی آنها همواره صفر است. همچنین می‌دانیم در مرکز مختصات، این دو تابع $Sinc$ صفر نیستند و قرینگی دایره‌ای هم دارد. تمام مقادیر $Sinc$ نیز مثبت نیستند و شامل اعداد منفی نیز هست.



b. .....

c. این تابع در واقع دو تابع مربعی شیفت خورده است. فوریه هر کدام از آنها دو تابع $Sinc$ مانند بخش قبل خواهد بود. شیفت دادن تبدیل فوریه به میزان α در هر جهت، حاصل شیفت نخورده را در $e^{-j\pi\alpha u}$ ضرب می‌کند. همچنین بنابر خاصیت خطی بودن تبدیل فوریه خواهیم داشت:

$$sinc(u)sinc(v)e^{-j2\pi\alpha u} + sinc(u)sinc(v)e^{j2\pi\alpha u}$$
$$= sinc(u)sinc(v)2\cos(2\pi\alpha u)$$

که مشابه بخش a فقط دارای قسمت حقیقی است (Sin ها ساده می‌شوند). بنابراین قسمت موهومی در همه نقاط برابر صفر است و قرینگی دایره‌ای نیز نداریم و باز هم مانند بخش a تمام مقادیر مثبت نیستند.

d. ......

e. این تابع از ترکیب دو تابع مربعی تشکیل شده. تابع سمت چپ مشابه بخش c به مقدار $-\alpha$ در جهت x خورده و تابع سمت راست به میزان α شیفت خورده و همچنین در منفی ۱ نیز ضرب شده است. تبدیل فوریه دو تابع را می‌نویسیم:

۱) $sinc(u)sinc(v)e^{j2\pi\alpha u}$

۲) $sinc(u)sinc(v)e^{-j2\pi\alpha u} * -1$

می‌دانیم وقتی جمع یک تابع در تمام نقاط صفر شود، DC Term یا همان $F(0.0)$ برابر با صفر می‌شود. که در این شکل این اتفاق می‌افتد.

f. ....

g. تبدیل فوریه تابع ضربه شیفت خورده به میزان α در جهت u، برابر با $e^{-j2\pi\alpha u}$ است.

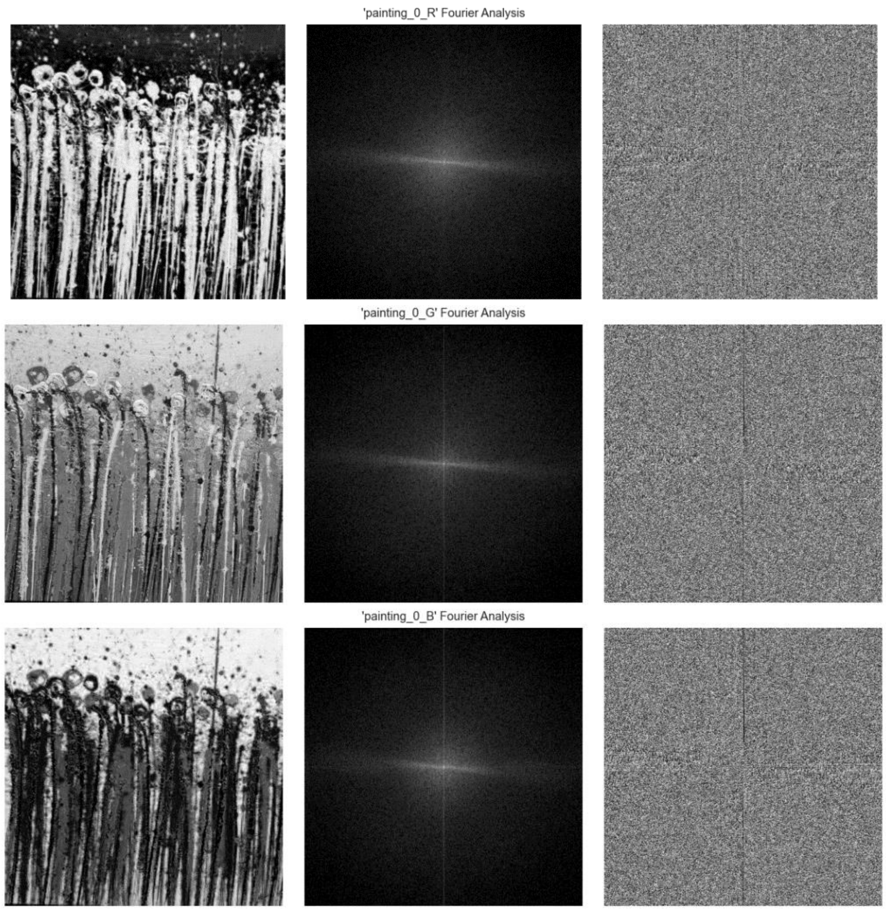
$$e^{-j2\pi\alpha u} = 2\cos(-2\pi\alpha u) + jsin(2\pi\alpha u)$$

بنابراین در نقطه صفر و صفر برابر با صفر نیست، دارای مولفه های حقیقی و موهومی است، مقادیر منفی نیز دارد، و قرینه دایره ای نیز دارد.

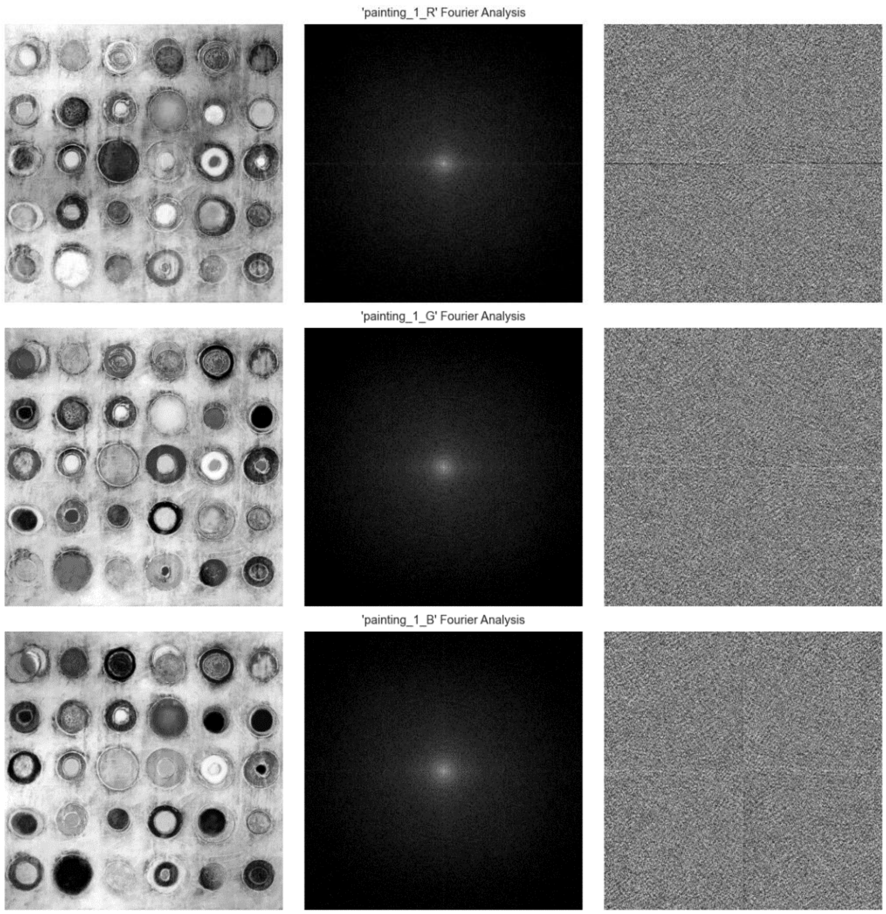
h. می‌دانیم وقتی جمع یک تابع در تمام نقاط صفر شود، DC Term یا همان F(۰,۰) برابر با صفر می‌شود. که در این شکل این اتفاق می¬افتد.

b) در این بخش از نقاشی های داده شده تبدیل فوریه میگیریم، Magnitude و Phase را برای هر کانال محاسبه میکنیم و نمایش میدهیم. همچنین دقت کنید برای بهتر دیده شدن Magnitude حتما از آن *log* بگیرید. همچنین در تمام سوالات این تمرین برای بهتر دیده شدن Magnitude فوریه، محدوده نمایش مقادیر سیاه و سفید دستکاری شده ان (به کمک *vmin, vmax* تابع *imshow*).

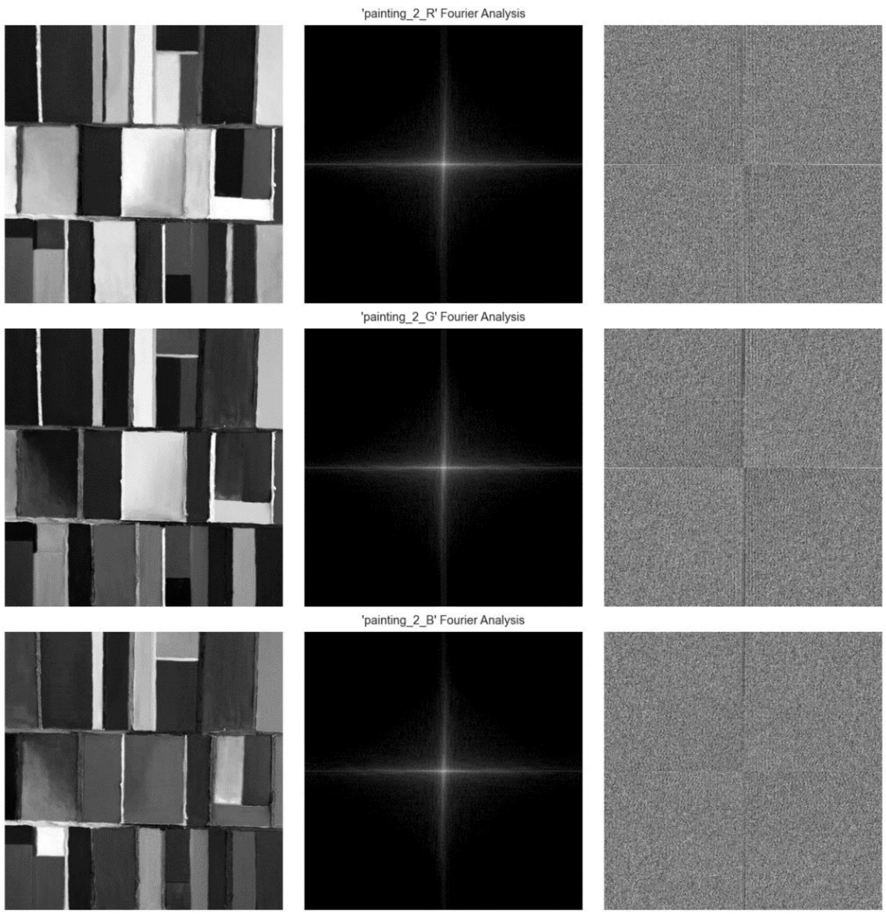
a. **نقاشی اول**: همانطور که در تصویر اصلی نیز واضح است، این نقاشی دارای تعدادی خط عمودی است. می‌دانیم برای تولید خطوط عمودی به موج های Sin, Cos در جهت افقی نیاز داریم. همچنین برای ایجاد لبه های Sharp به فرکانس های بیشتری نیاز داریم. و همانطور که مشاهده میکنید در Magnitude مربوط به کانال های مختلف این تصویر، خط افقی دیده می‌شود که در واقع موج‌هایی در جهت افقی با فرکانس های متعدد را نشان می‌دهد.



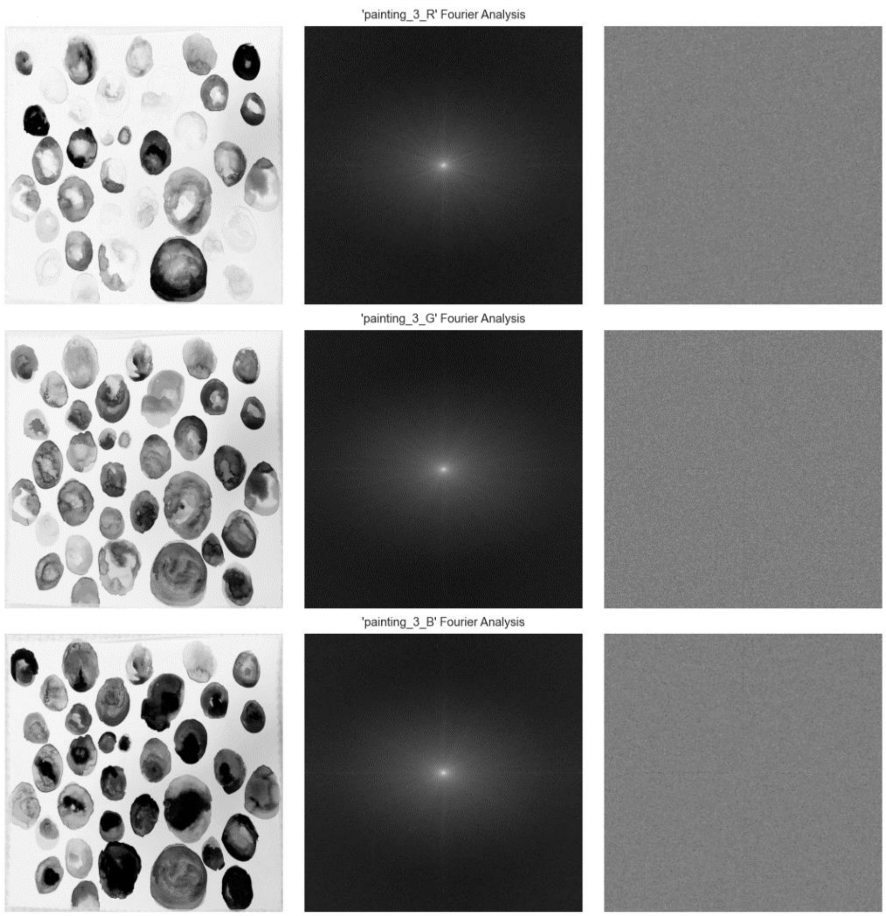
b. **نقاشی دوم**: همانطور که در تصویر اصلی نیز واضح است، این نقاشی دارای تعدادی دایره است. از آنجایی که دایره Circular Symmetry دارد، Magnitude تبدیل فوریه آن هم Circular Symmetry دارد. دلیل این اتفاق هم واضح است، یک دایره درواقع تعداد بینهایت لبه در تمام جهات ممکن است بنابراین به موج های در تمام جهات نیازمندیم.



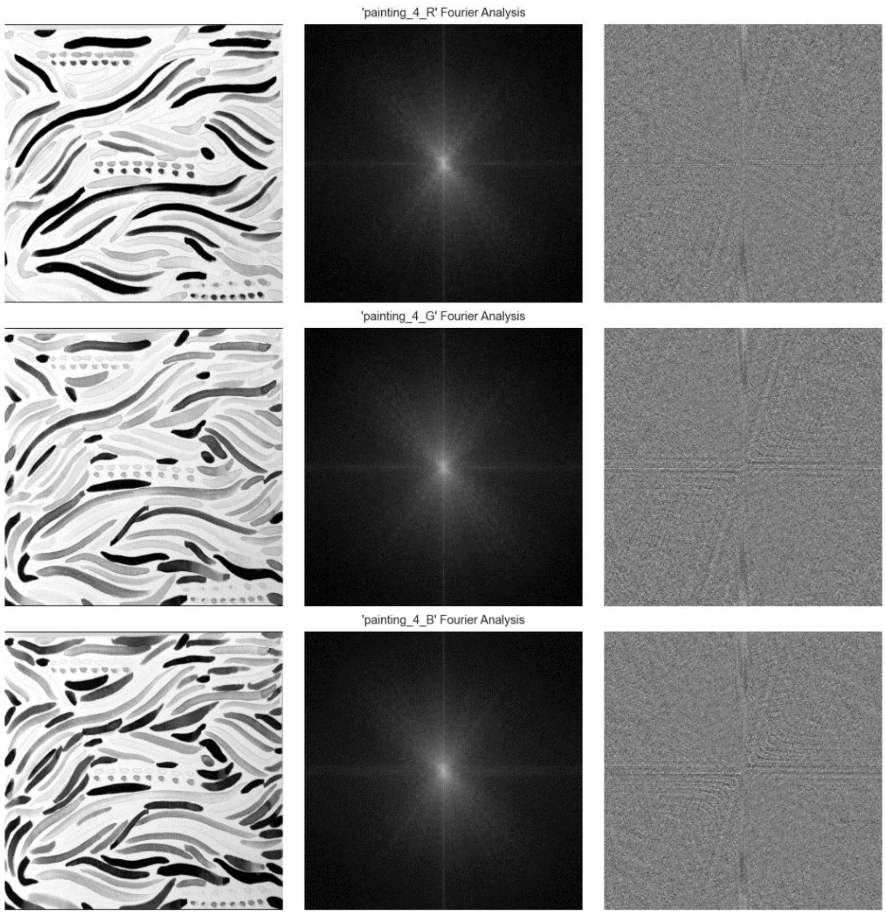
c. **نقاشی سوم**: همانطور که در تصویر اصلی نیز واضح است، این نقاشی دارای تعدادی خط افقی و عمودی است. در بخش a دیدیم که خطوط افقی در تصویر به خطوط عمودی در تبدیل فوریه تبدیل می‌شوند و برعکس. از آنجایی که در این تصویر هم خطوط افقی داریم و هم خطوط عمودی، واضح است که در تبدیل فوریه نیز هم موج های عمودی داشته باشیم و هم موج های افقی.



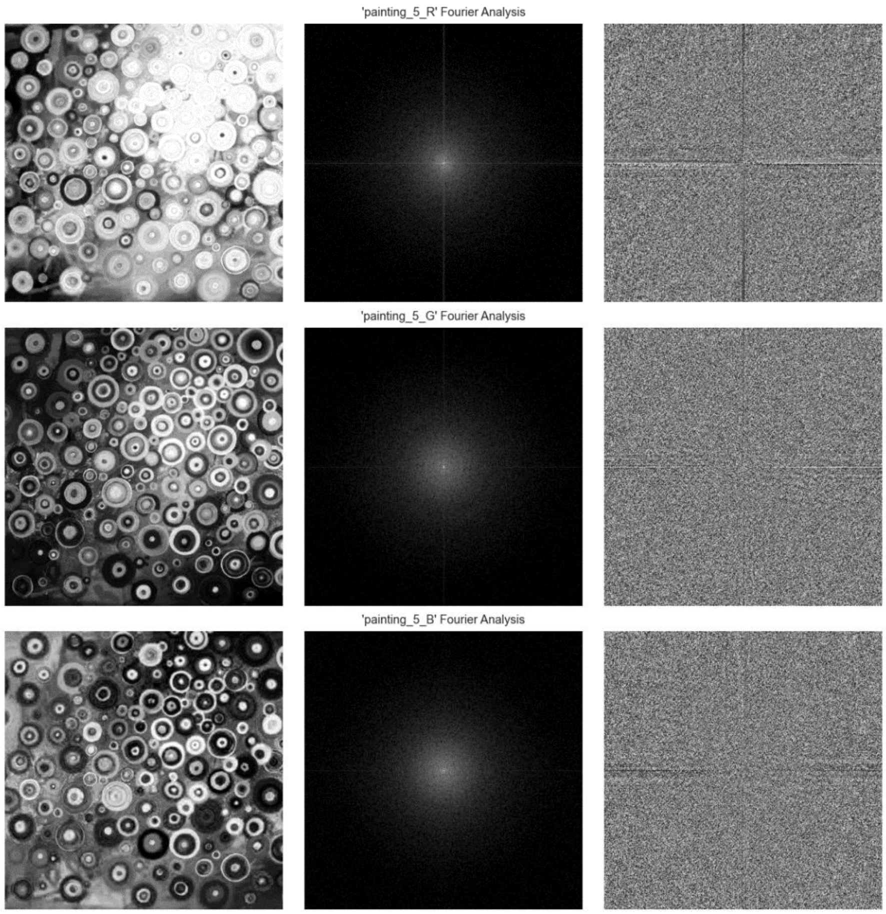
d. **نقاشی چهارم**: مشابه مورد b منتهی این تصویر نسبت به تصویر b قرینگی کمتری دارد و یکسری خطوط (لبه) در جهت های مختلف در تصویر دیده می‌شود. بنابراین تعدادی Streak در زوایای مختلف مشاهده می-شود. این Streak ها در همه زوایا نیستند زیرا این اشکال دایره کاملا هندسی نیستند.



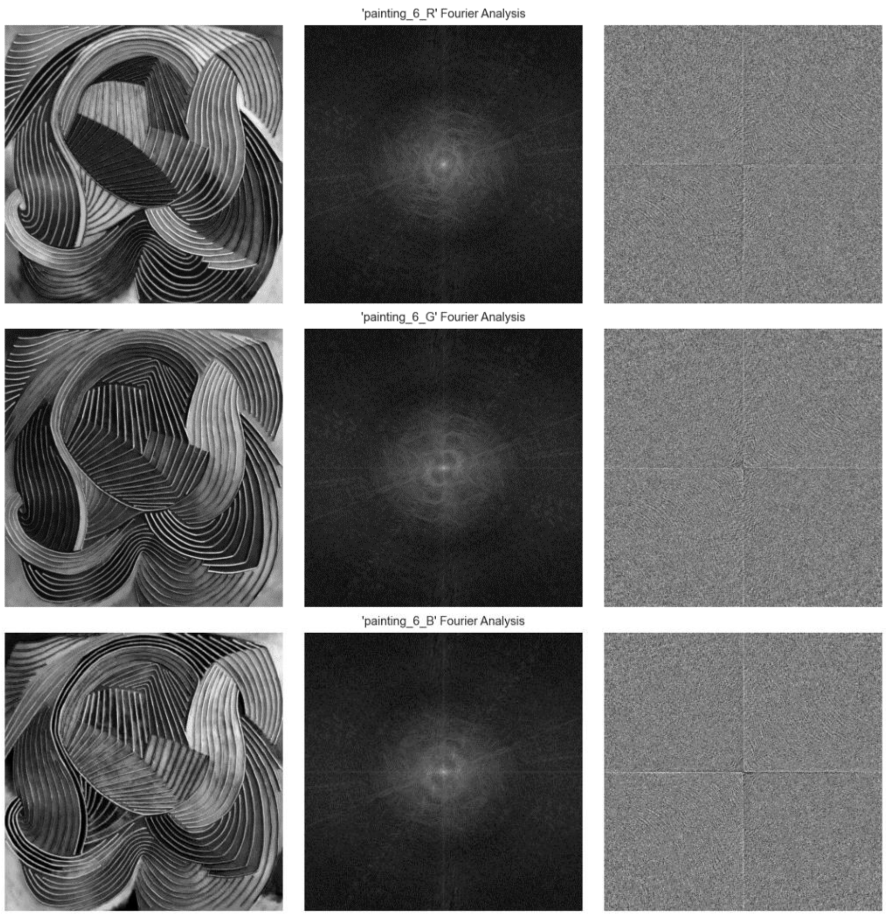
e. **نقاشی پنجم**: این تصویر تعدادی لبه در جهت های مختلف دارد اما از بین این خطوط تقریبا هیچ کدام آنها در جهت عمودی کامل قرار ندارند. بنابراین توقع داریم در تبدیل فوریه خطوط کاملا افقی نبینیم. و اگر به Magnitude فوریه توجه کنیم میبینیم که همینطور نیز است. تعدادی Streak در جهت های عموما عمودی، که نشان دهنده خطوط افقی و تقریبا افقی در تصویر اصلی است مشاهده می‌شود.



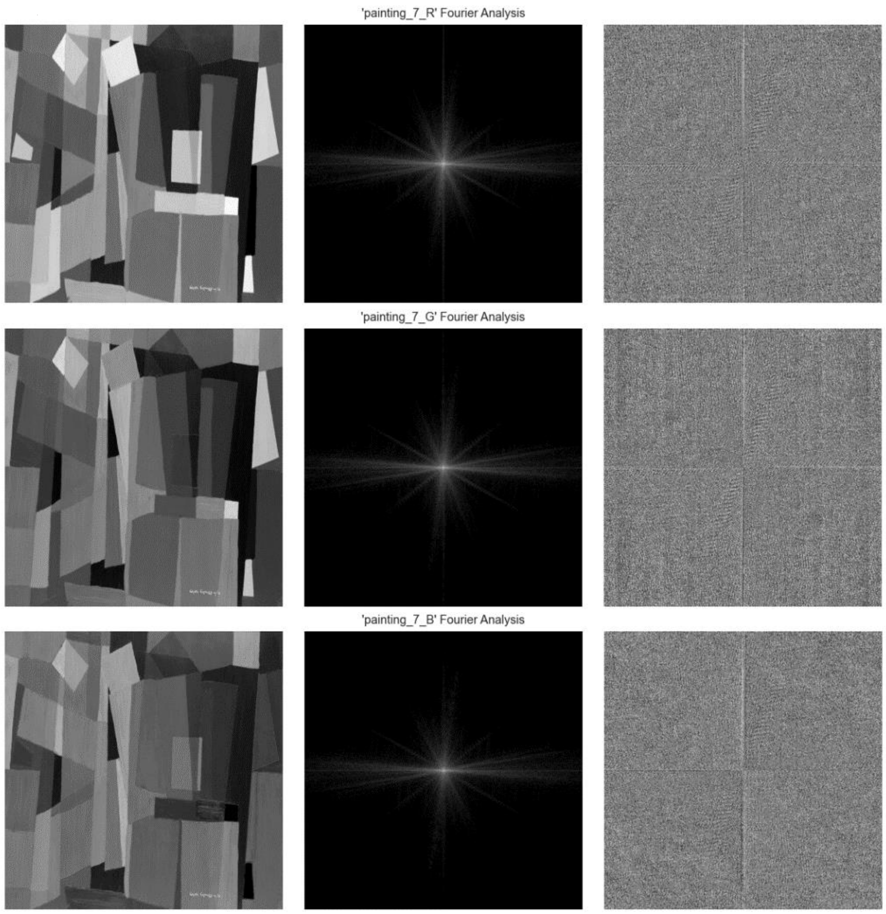
f. **نقاشی ششم**: مانند سایر تصاویری که دایره داشتند.



g. **نقاشی هفتم**: در این تصویر منحنی های پیچیده تری را مشاهده میکنیم که این پیچیدگی در تبدیل فوریه نیز مشاهده می‌شود. این منحنی ها کاملا خط نیستند، کاملا دایره هم نیستند. در تصویر مربوط به Magnitude فوریه نیز، خطوط، در اشکالی شبیه به دایره اما نه کاملا دایره قرار دارند که حاکی از وجود منحنی های مختلف در جهات مختلف است. تحلیل این تصاویر پیچیده تر به کمک تبدیل فوریه کار دشواری است.



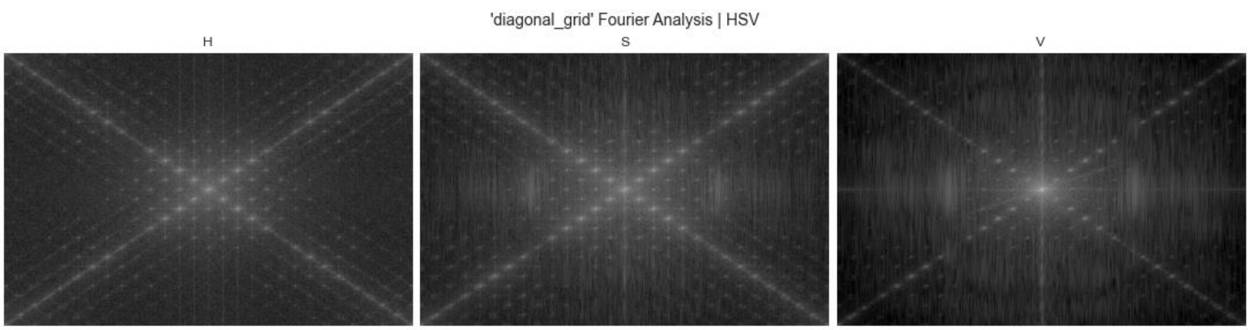
h. **نقاشی هشتم**: این تصویر شباهت زیادی به تصویر c دارد با این تفاوت که علاوه بر خطوط افقی و عمودی، تعداد زیادی خطوط مورب نیز داریم. این خطوط مورب در تصویر Magnitude تبدیل فوریه به شکل خطوط مورب و در جهت عمود خطوط تصویر اصلی ظاهر می‌شوند.

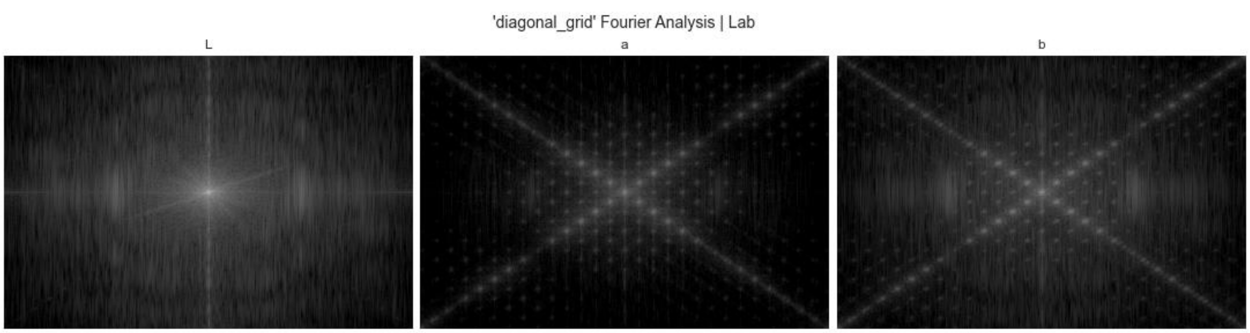


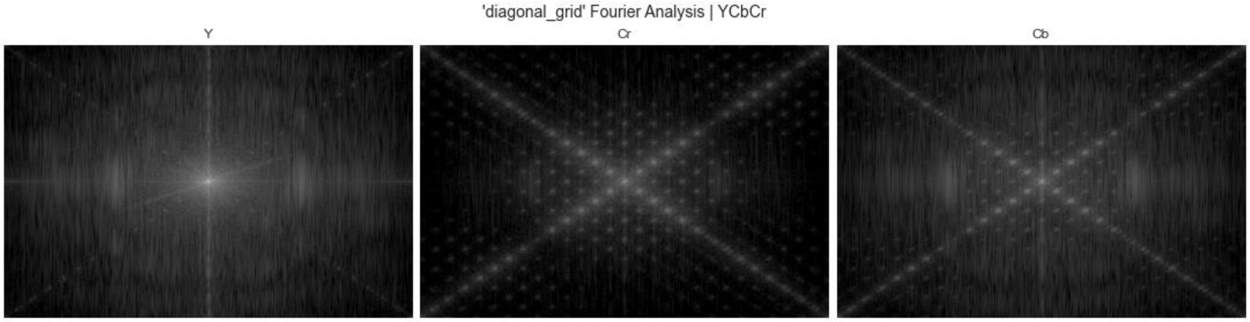
سوال ۴)

a) در این بخش تبدیل فوریه را بر روی ۳ تصویر داده شده در فضاهای رنگی مختلف اعمال میکنیم. خواهیم دید که فضای رنگی **YCbCr** بهتر از بقیه فضاهای رنگی در جداکردن پترن‌های موجود در تصویر عمل می‌کند.

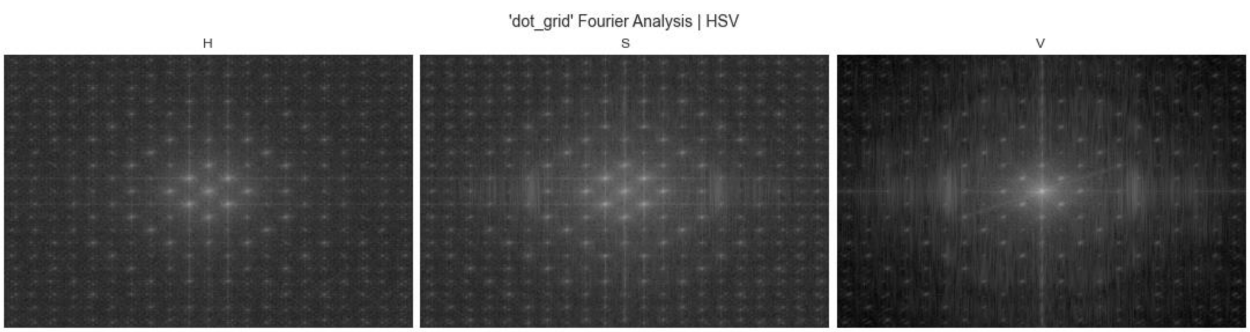
تصویر Diagonal_grid

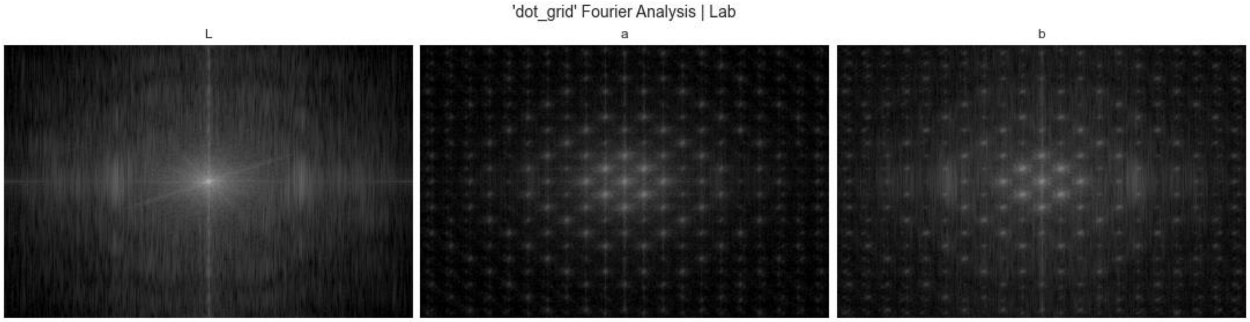


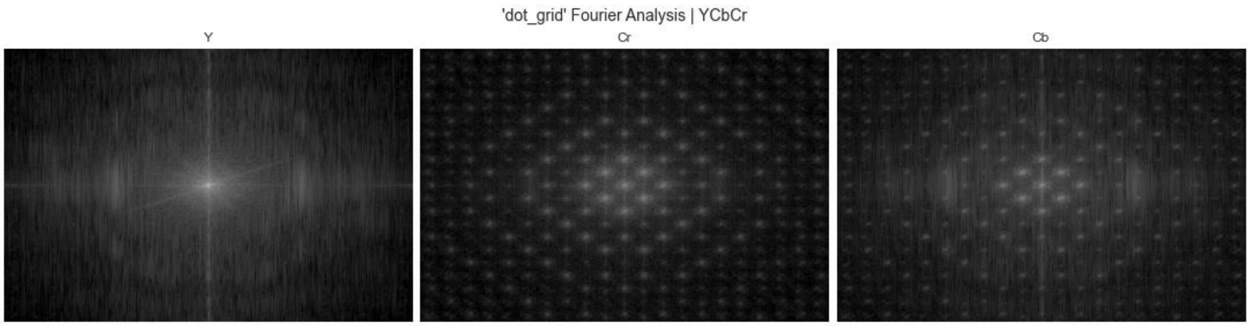




تصویر Dot_grid

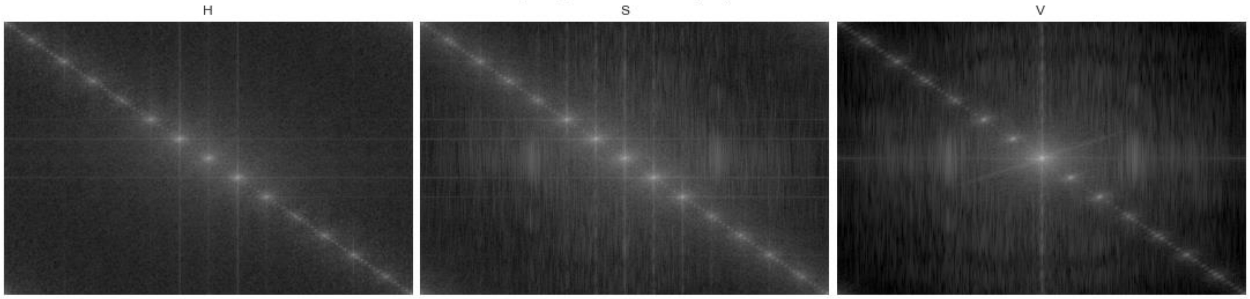




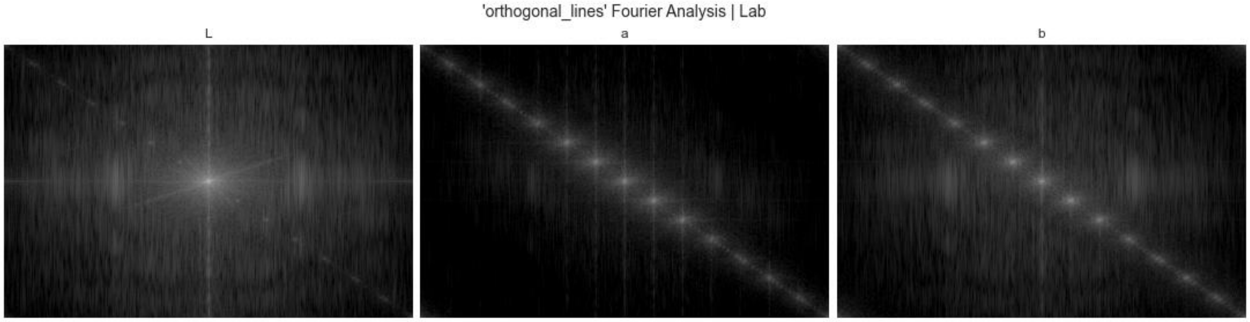


تصویر Orthogonal_lines

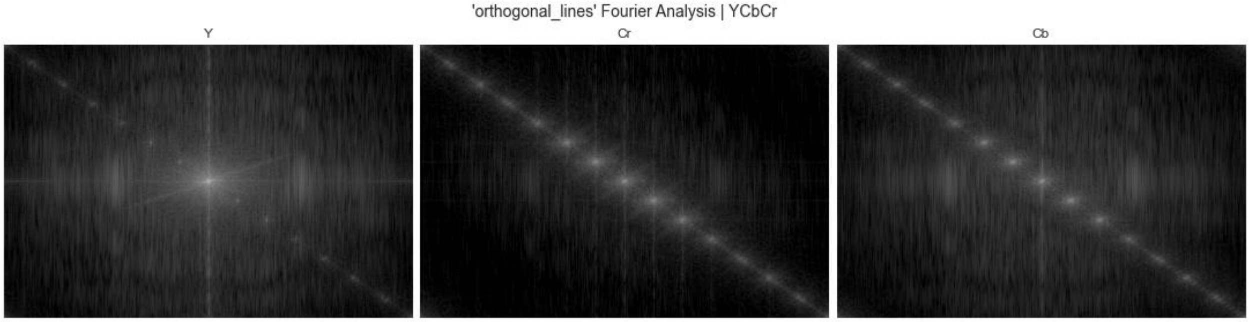
'orthogonal_lines' Fourier Analysis | HSV



'orthogonal_lines' Fourier Analysis | Lab



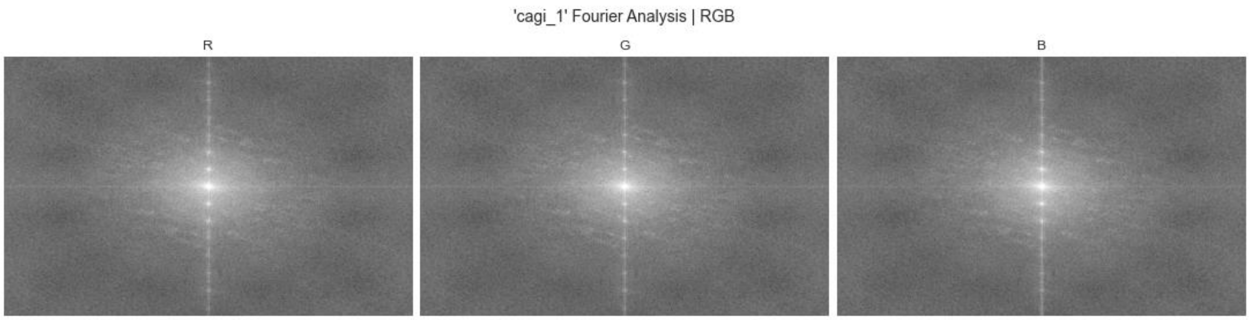
'orthogonal_lines' Fourier Analysis | YCbCr



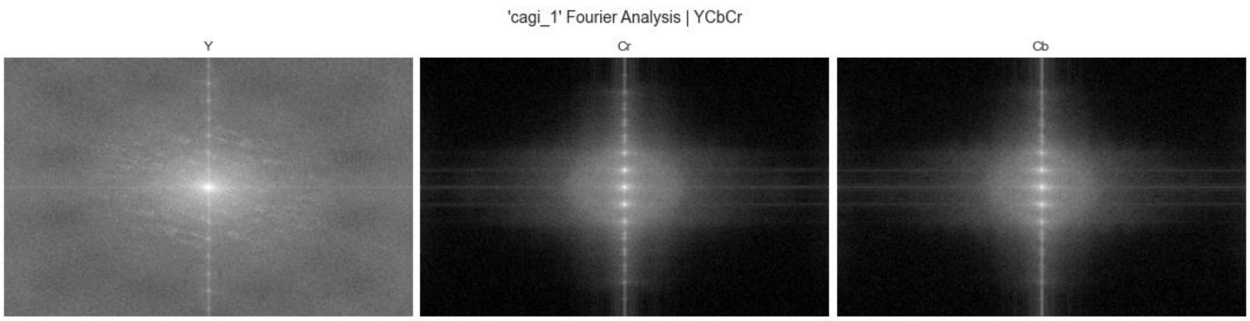
b) در این بخش سعی میکنیم خطوط Periodic موجود در تصاویر داده شده در این بخش را حذف کنیم. امید است که پس از حذف کردن این خطوط اثبات شود که تصاویر زیر این خطوط در واقع تصاویر سطح خاکستری ای بوده اند که چشم ما به اشتباه به شکلی رنگی تصور می‌کرده است.

در **مرحله اول** نیاز است مانند بخش قبل بهترین فضای رنگی را برای حذف کردن این خطوط پیدا کنیم. برای جلوگیری از شلوغ شدن بی مورد گزارش، این نتیجه، که بسیار شبیه به بخش قبل است، را فقط برای تصویر اول نشان می‌دهیم.

فضای **RGB** برای تصویر Cagi_1



فضای **YCbCr** برای تصویر Cagi_1

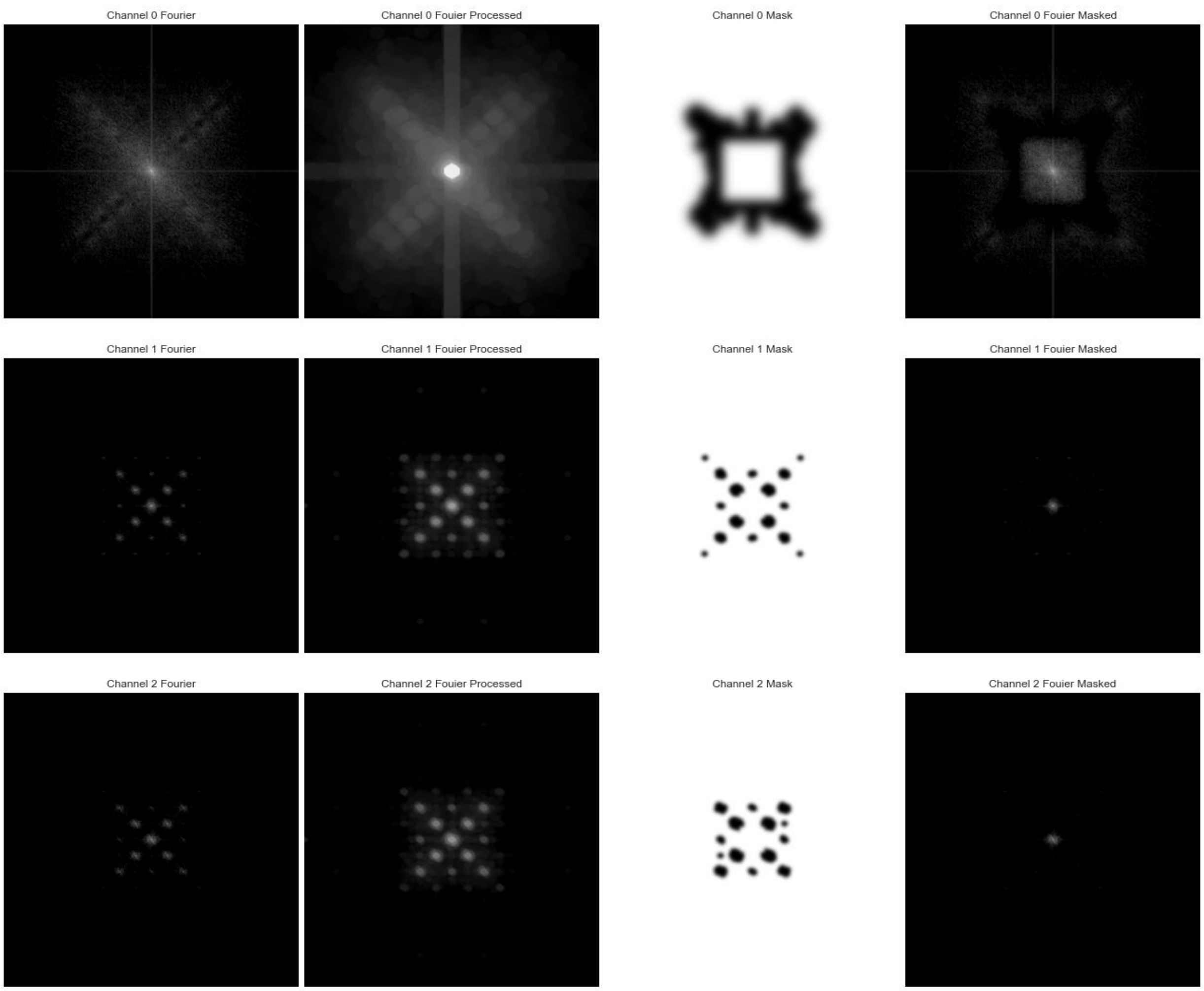


واضح است که فضای رنگی **YCbCr** نمایش بهتری برای جدا کردن خطوط دارد. تصاویر کامل مربوط مقایسه فضاهای رنگی هر ۳ تصویر در پوشه ارسالی موجود است.

در **مرحله دوم** میخواهیم مکانیزمی خودکار برای تولید ماسکی جهت حذف کردن نقاط بسیار روشن در Magnitude تبدیل فوریه ایجاد کنیم. این نقاط روشن در واقع همین پترن‌های تکراری موجود در تصویر هستند. لازم به ذکر است که نقطه روشنی که در مرکز تصویر مشاهده میکنید، نباید حذف شود. این ناحیه شامل تمام فرکانس های پایین سازنده تصویر است. با حذف کردن این ناحیه در واقع یک فیلتر بالاگذر ساخته ایم و باعث از بین رفتن اکثر اطلاعات اصلی تصویر می‌شویم.

برای تولید این ماسک چهار مرحله زیر را انجام می‌دهیم و سپس برای هر تصویر، ابرپارامتر های هر مرحله را تنظیم میکنیم.

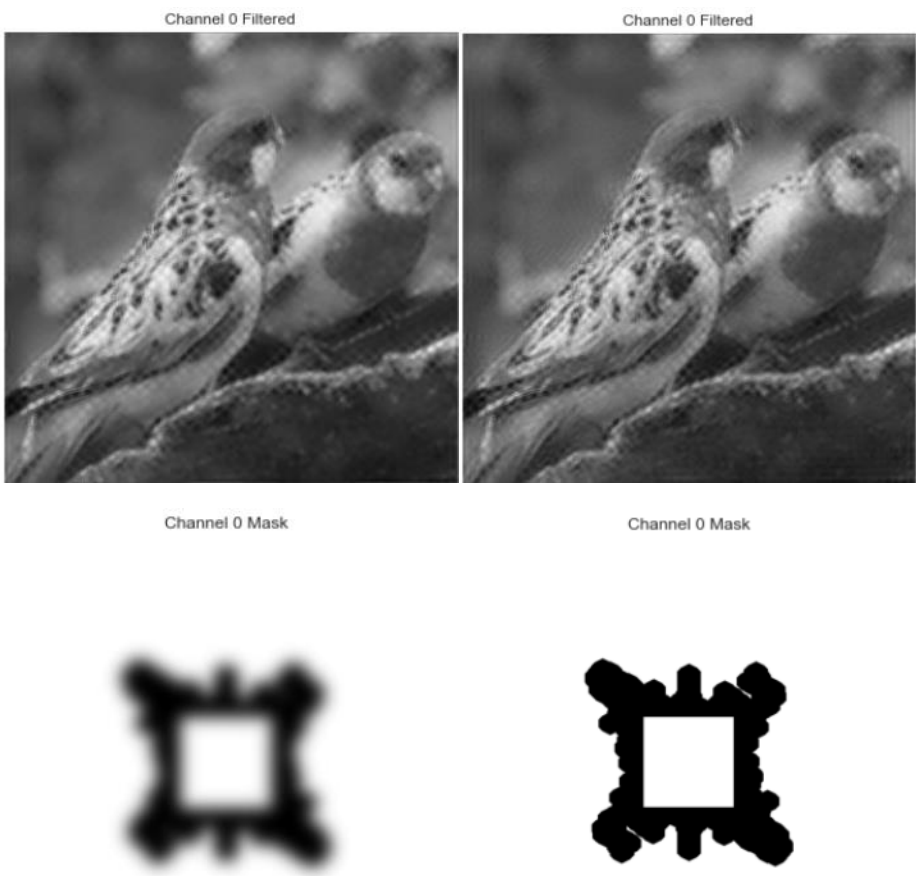
۱. پیش پردازش Magnitude برای واضح تر شدن نقاط روشن تصویر
۲. اعمال Thresholding جهت حذف کردن نواحی بسیار روشن
۳. بازگرداندن مرکز Magnitude
۴. اعمال Gaussian Blur بر روی ماسک تولید شده، جهت مقابله با پدیده **Ringing** ناشی از ماسک شارپ



همانطور که در تصویر صفحه قبل مشاهده می‌کنید، ابتدا با کمک اپراتور Dilate تصویر Magnitude را پردازش میکنیم تا نقاط روشن خود را بهتر نشان دهند. سپس به کمک یک آستانه گیری ساده، نقط بسیار روشن را شناسایی میکنیم و در ماسک مقدار ۰ را به آن پیکسل ها نسبت می‌دهیم. همچنین از ۰ شدن ناحیه وسط ماسک جلوگیری میکنیم. سپس با اعمال Gaussian Blur به ماسک تولید شده، لبه ها را نرم میکنیم تا از Ringing جلوگیری کنیم. در نهایت ستون سمت راست Magnitude جدید را نشان می‌دهد.

اشتباه رایج: دقت کنید از Magnitude تبدیل فوریه صرفا به عنوان یک راهنما استفاده میکنیم و در نهایت ماسک تولید شده را به خروجی اصلی DFT اعمال میکنیم. نه به Magnitude محاسبه شده از آن!

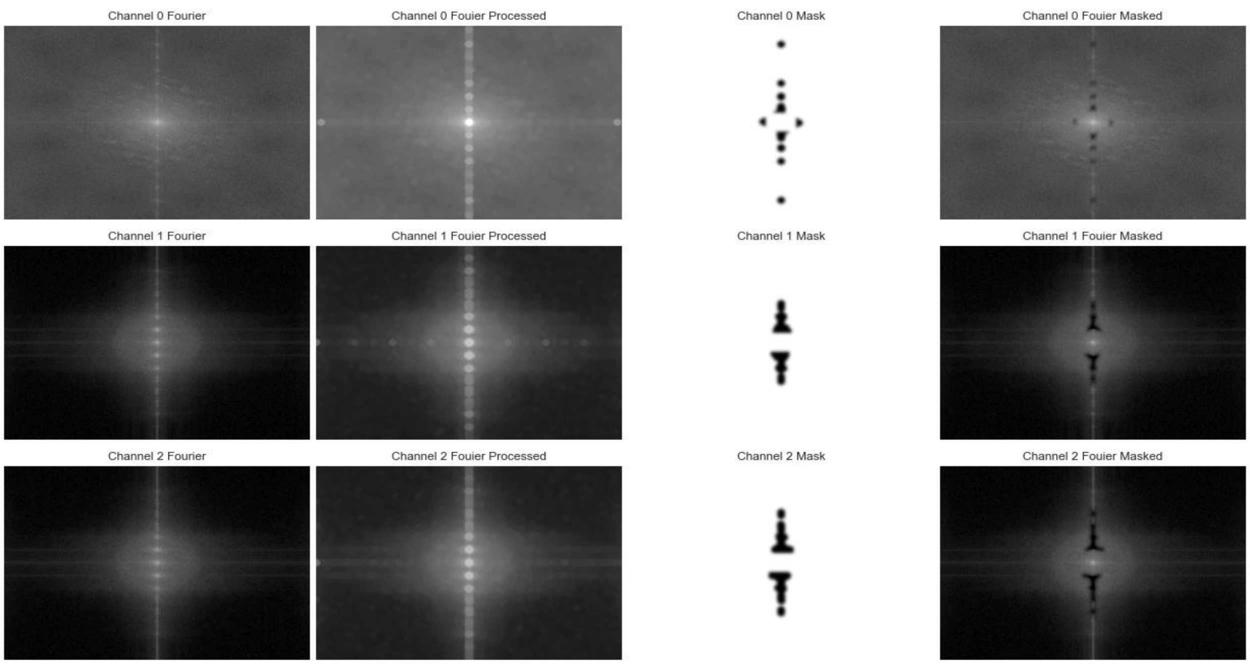
در تصویر زیر می‌توانید اثر اعمال Gaussian Blur به ماسک را مشاهده کنید. به تصویر سمت راست بلر اعمال نشده و پدیده رینگینگ مشاهده می‌شود.



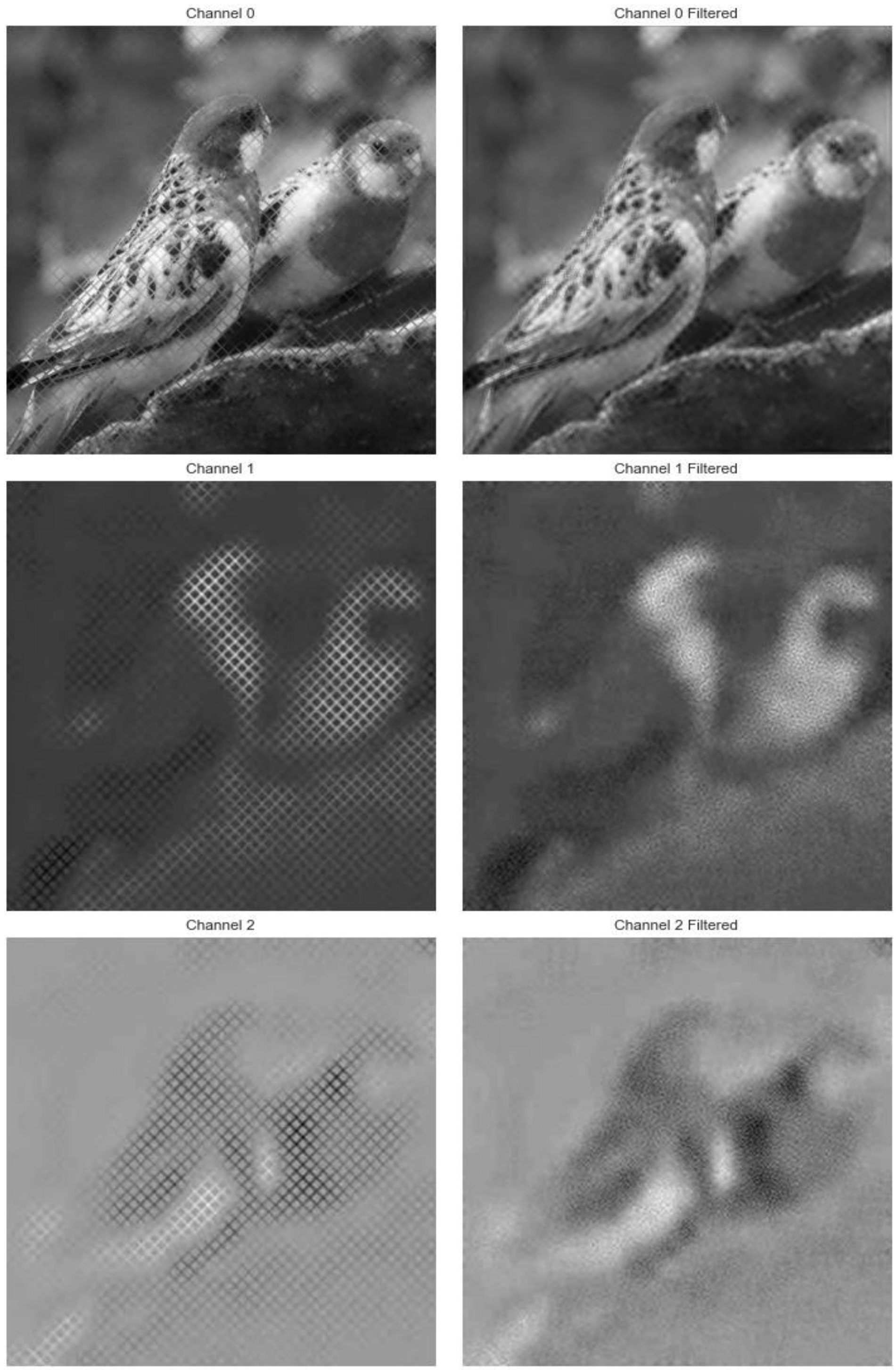
حالا که با روند کلی تولید ماسک و فیلترکردن تصویر کانال های مختلف آشنا شدیم نتایج این بخش بر روی تصاویر مختلف را مشاهده میکنیم. به ازای هر تصویر ورودی، سه خروجی مشاهده میکنید. **تصویر اول** نتیجه اعمال فیلترینگ به کانال های مختلف در فضای **YCbCr** (فهمیدیم که این فضا بهترین فضای رنگی برای این تسک است) است. **تصویر دوم** نتیجه ترکیب سه کانال فیلتر شده و تبدیل فضای رنگی به RGB است. **تصویر سوم** مراحل تولید و اعمال ماسک است که در مورد آن صحبت کردیم.

تصویر اول

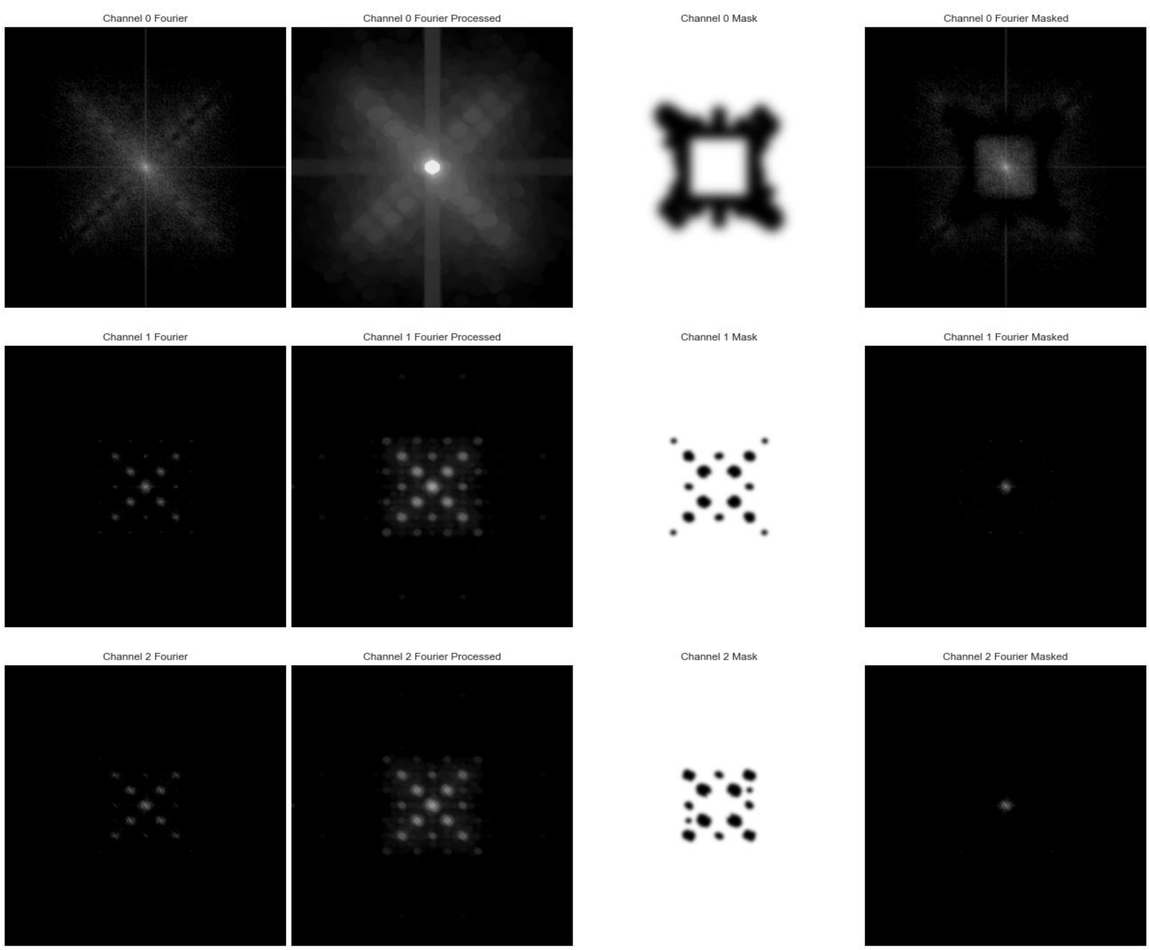

"cagi_1" Original
"cagi_1" Filtered


Channel 0 Fourier
Channel 0 Fouier Processed
Channel 0 Mask
Channel 0 Fouier Masked
Channel 1 Fourier
Channel 1 Fouier Processed
Channel 1 Mask
Channel 1 Fouier Masked
Channel 2 Fourier
Channel 2 Fouier Processed
Channel 2 Mask
Channel 2 Fouier Masked

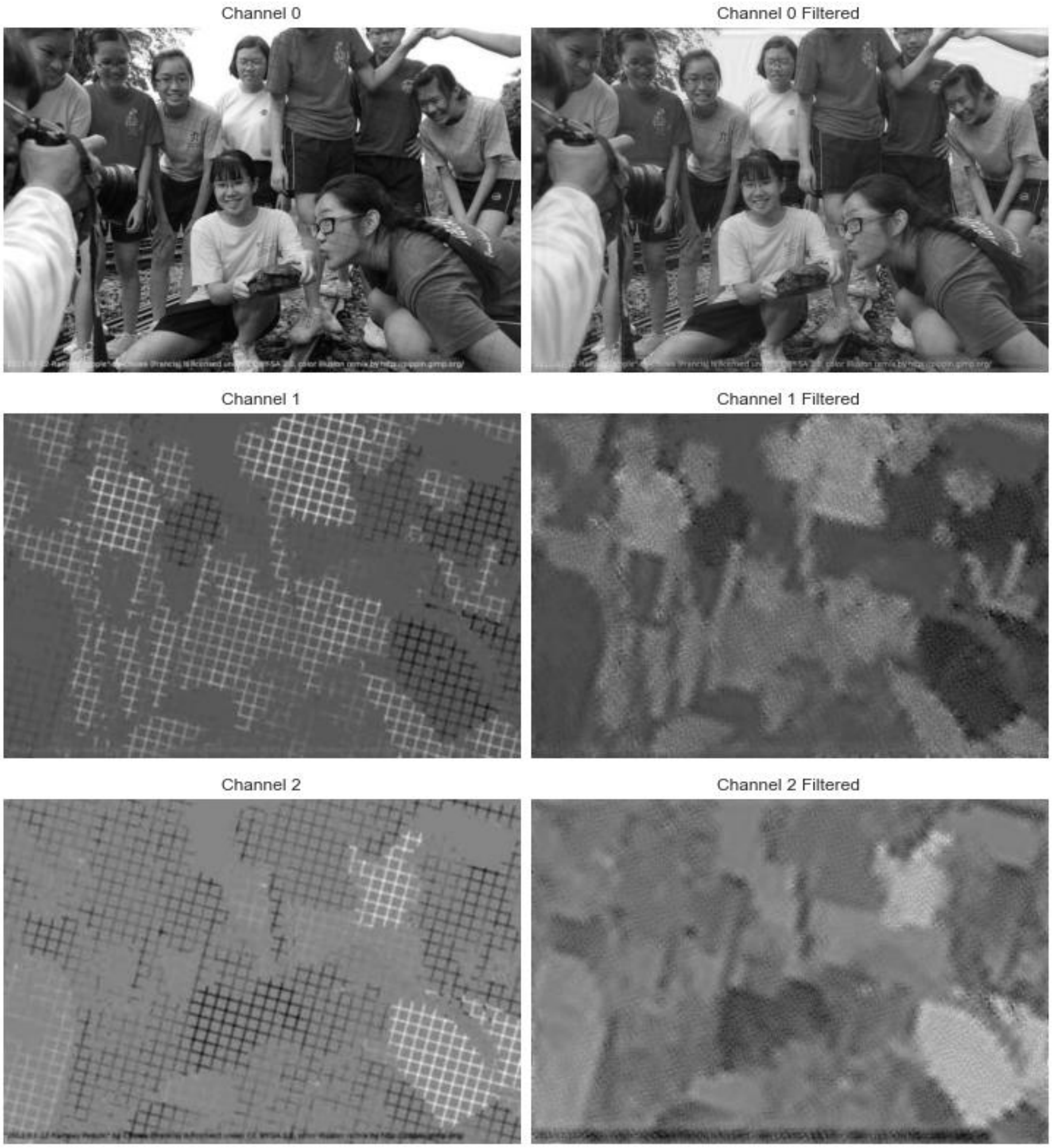
تصویر دوم


Channel 0
Channel 0 Filtered
Channel 1
Channel 1 Filtered
Channel 2
Channel 2 Filtered

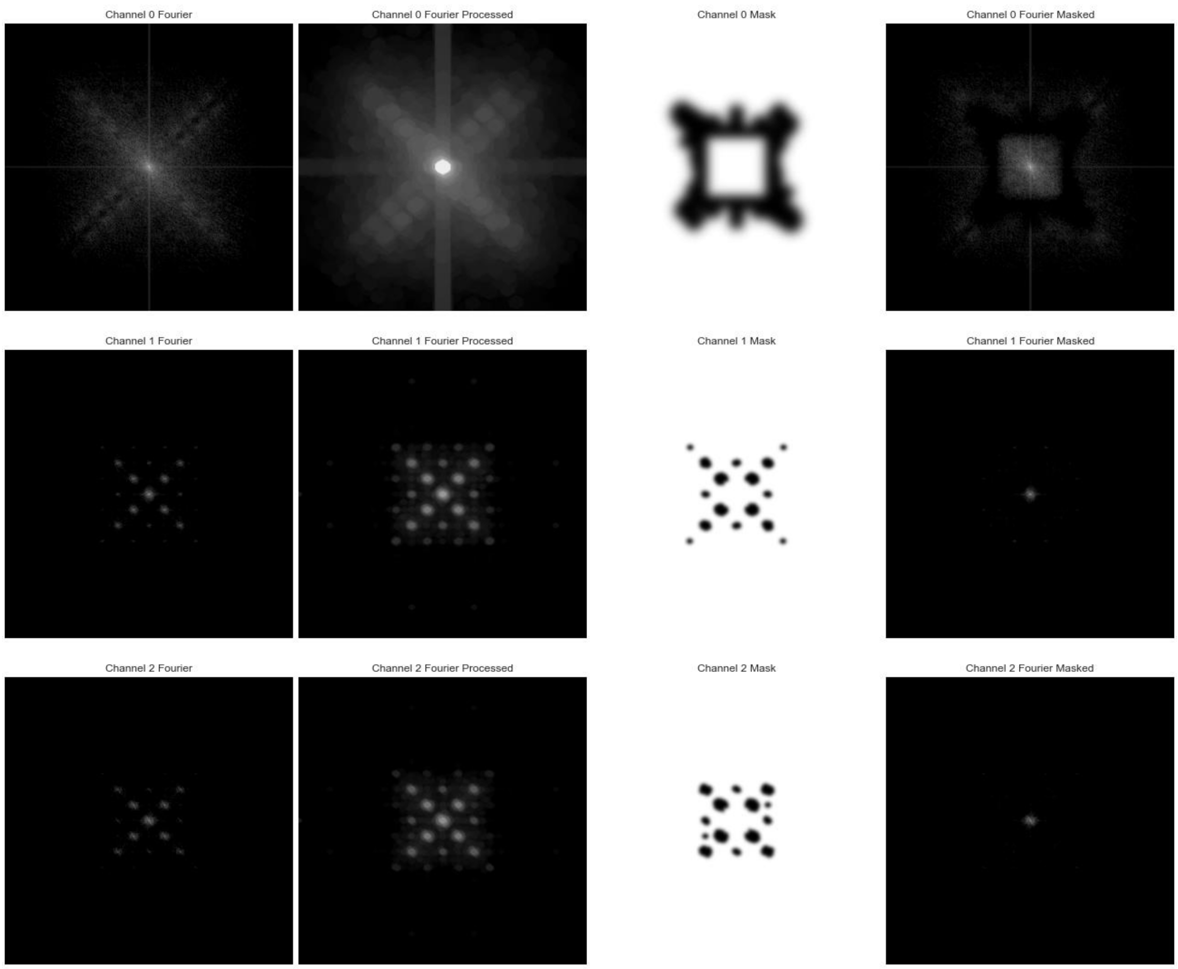

"cagi_2" Original
"cagi_2" Filtered


Channel 0 Fourier
Channel 0 Fourier Processed
Channel 0 Mask
Channel 0 Fourier Masked
Channel 1 Fourier
Channel 1 Fourier Processed
Channel 1 Mask
Channel 1 Fourier Masked
Channel 2 Fourier
Channel 2 Fourier Processed
Channel 2 Mask
Channel 2 Fourier Masked

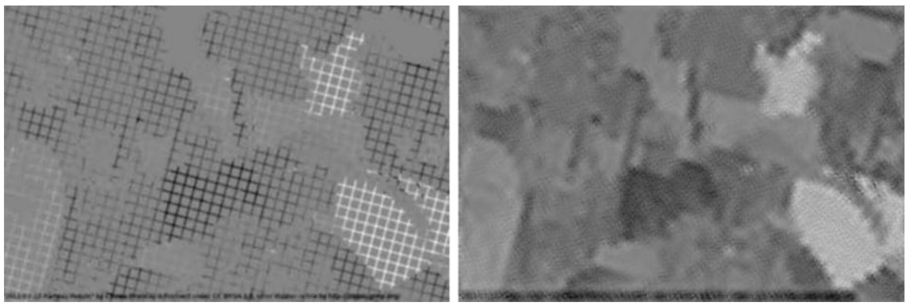
تصویر سوم


Channel 0
Channel 0 Filtered
Channel 1
Channel 1 Filtered
Channel 2
Channel 2 Filtered


"cagi_3" Original
"cagi_3" Filtered


Channel 0 Fourier
Channel 0 Fourier Processed
Channel 0 Mask
Channel 0 Fourier Masked
Channel 1 Fourier
Channel 1 Fourier Processed
Channel 1 Mask
Channel 1 Fourier Masked
Channel 2 Fourier
Channel 2 Fourier Processed
Channel 2 Mask
Channel 2 Fourier Masked

نتیجه گیری: همانطور که از نتایج مشخص بود نتیجه، با چیزی که انظار داشتیم کمی متفاوت بود. ما میخواستیم خطوط حذف شوند و سایر پیکسل ها دست نخورده باقی بمانند. اما در عمل با حذف شدن خطوط، مقدار رنگی این خطوط به سایر پیکسل ها انتشار داده شد. در قطعه تصویر زیر می‌توانید این اتفاق را به خوبی مشاهده کنید.

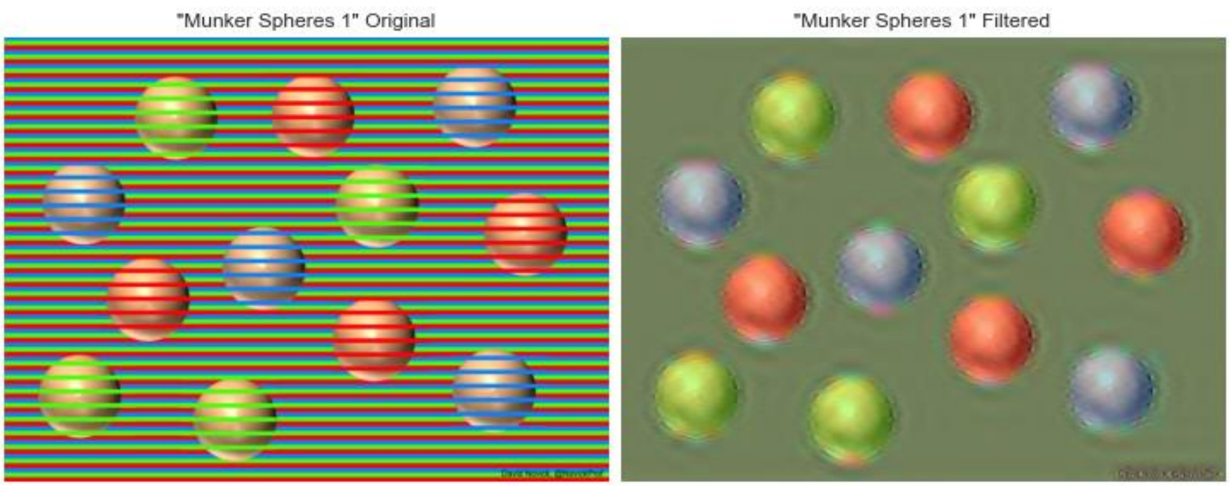


همانطور که مشاهده میکنید پیکسل های بین خطوط تقریبا خاکستری یک دست هستند اما پس از اعمال فیلترینگ و حذف این خطوط، مقدار رنگی نواحی بین این خطوط تحت تاثیر قرار گرفته و روشن و تاریک شده است. ممکن است این اتفاق به خاطر کم بودن رزولوشن تصویر(نسبت به قطر خطوط) باشد. به هر حال نتیجه از چیزی که انتظار داشتیم فاصله داشت.

c) در این بخش نیز به نتیجه مطلوب نرسیدیم و همان مشکل بخش قبل را داشتیم با این وجود تصاویر مربوط به اعمال فیلترینگ تصاویر این بخش را میتوانید مشاهده کنید. همچنین، برای دیدن سایر تصاویر(تصاویر مراحل ساخت ماسک و اعمال نتیجه بر روی تک تک کانال ها) به پوشه مربوط به این سوال مراجعه کنید.

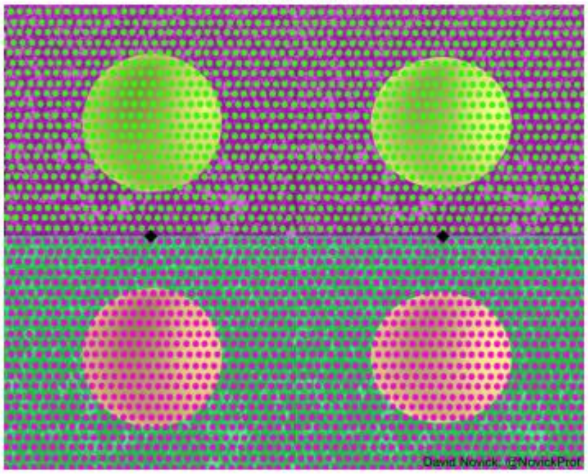
تصویر اول

تصویر دوم

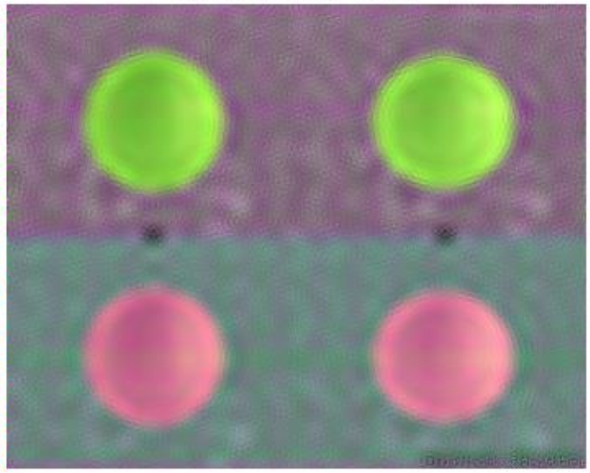


تصویر سوم

"Munker Spheres 2" Original



"Munker Spheres 2" Filtered



تصویر چهارم

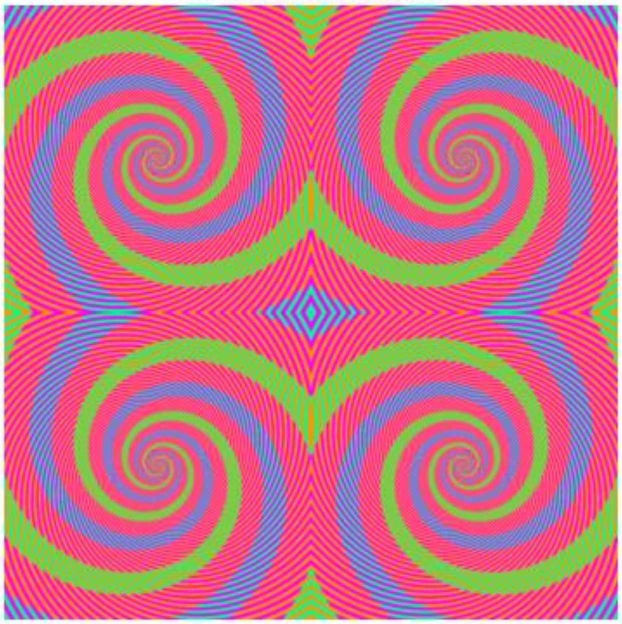
"Munker Zaigzag lines" Original

"Munker Zaigzag lines" Filtered

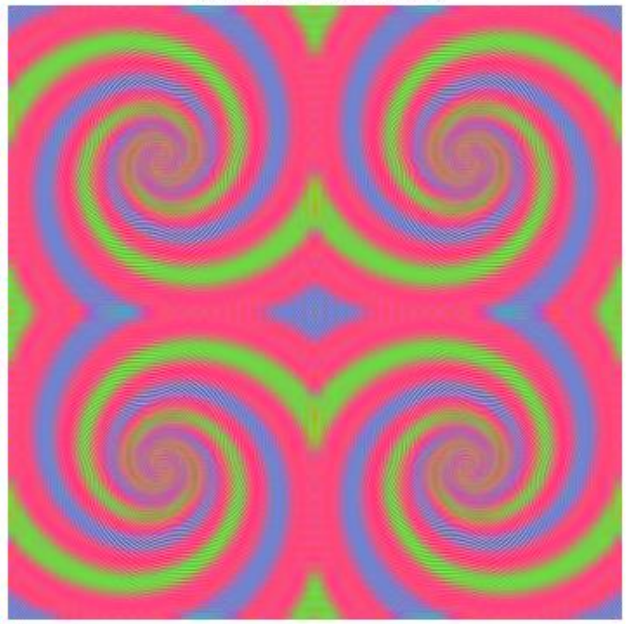
a) در این بخش نیز به نتیجه مطلوب نرسیدیم و همان مشکل بخش قبل را داشتیم با این وجود تصاویر مربوط به اعمال فیلترینگ تصاویر این بخش را میتوانید مشاهده کنید. همچنین، برای دیدن سایر تصاویر(تصاویر مراحل ساخت ماسک و اعمال نتیجه بر روی تک تک کانال ها) به پوشه مربوط به این سوال مراجعه کنید.

تصویر اول

"Munker Monspiral" Original

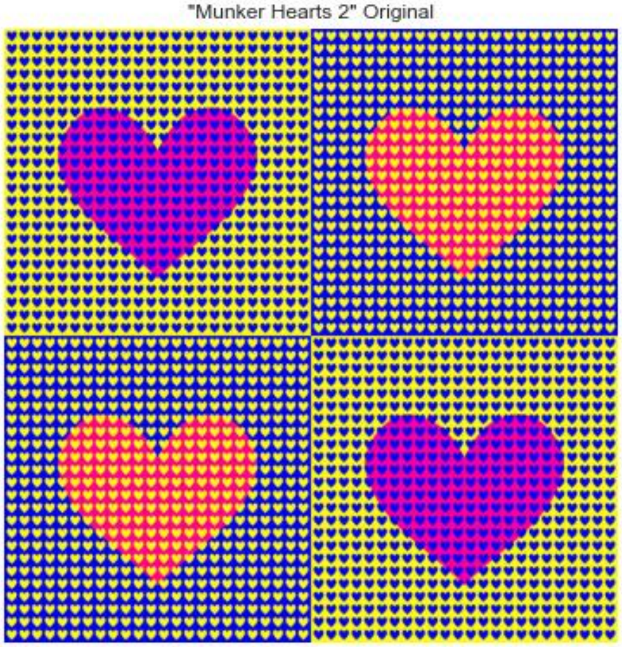


"Munker Monspiral" Filtered

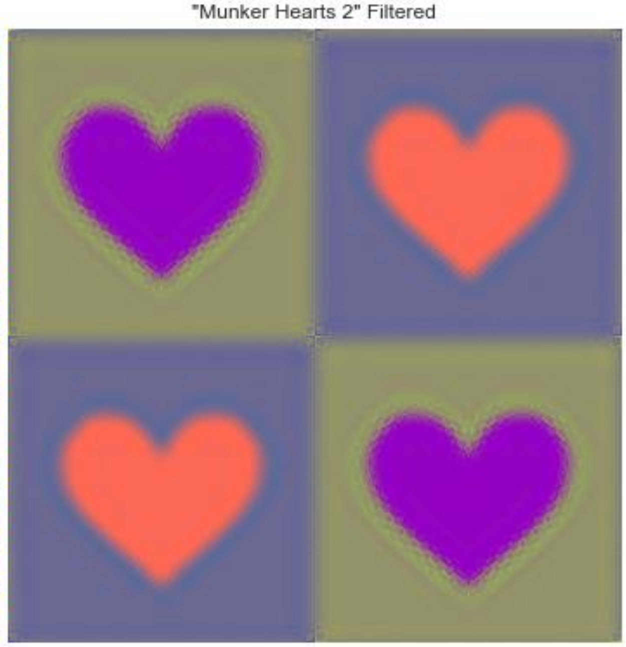


تصویر دوم

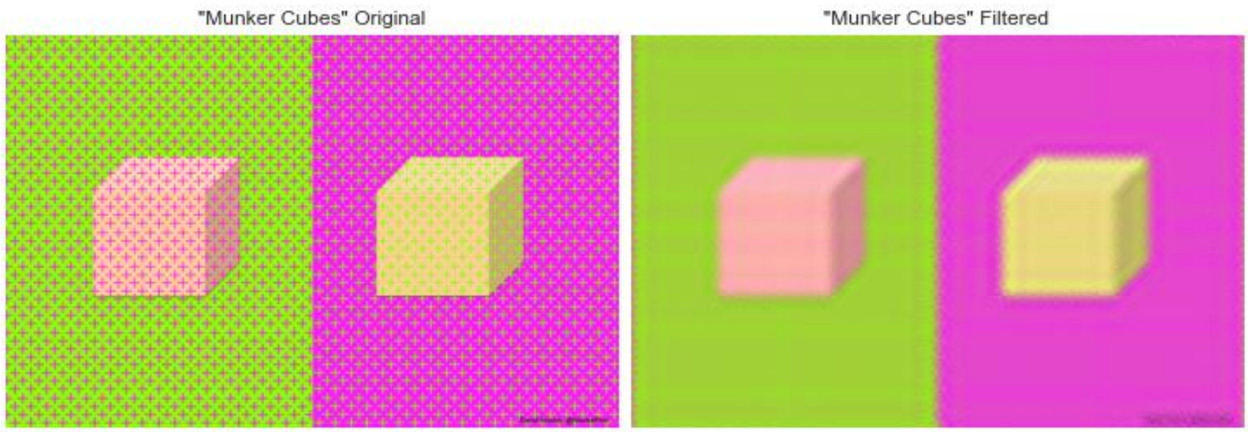
"Munker Hearts 2" Original



"Munker Hearts 2" Filtered

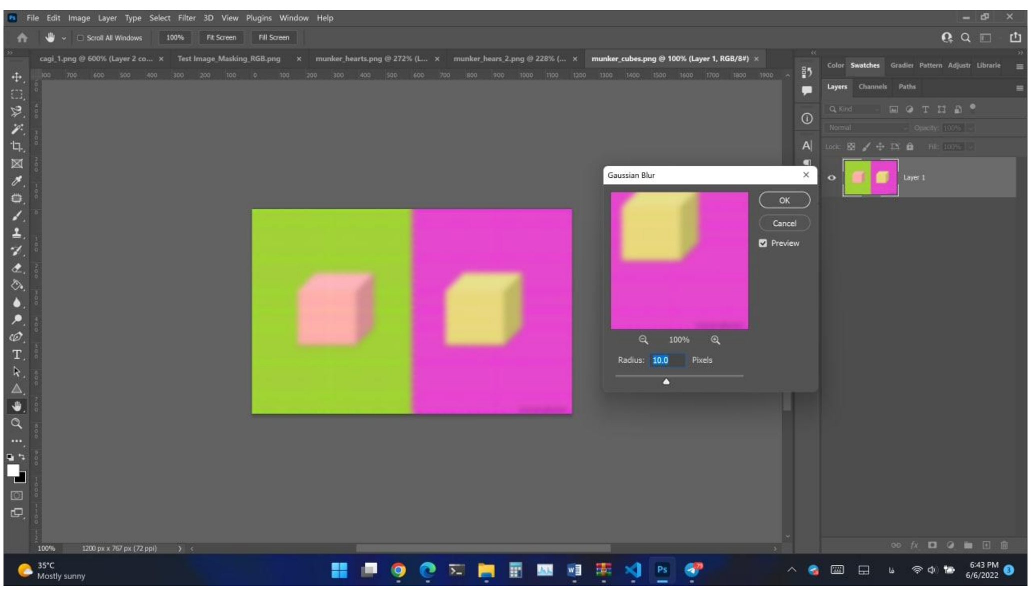


تصویر سوم



آپدیت بعد از سه روز:

با انجام آزمایش‌هایی متوجه شدیم که نتایج فیلتر کردن ما به کمک تبدیل فوریه، بسیار شبیه به اعمال یک فیلتر low-pass مانند Gaussian blur به تصاویر هستند. عملا کاری که ما انجام می‌دهیم اعمال یک فیلتر low-pass به تصاویر است به همین دلیل رنگ ها ترکیب می‌شوند و نتیجه مورد انتظار بدست نمی آید. **این مشکل حل نشد.**



اعمال فیلتر Gaussian Blur به تصویر که نتیجه ای مشابه با نتیجه ما دارد

سوال ۵)

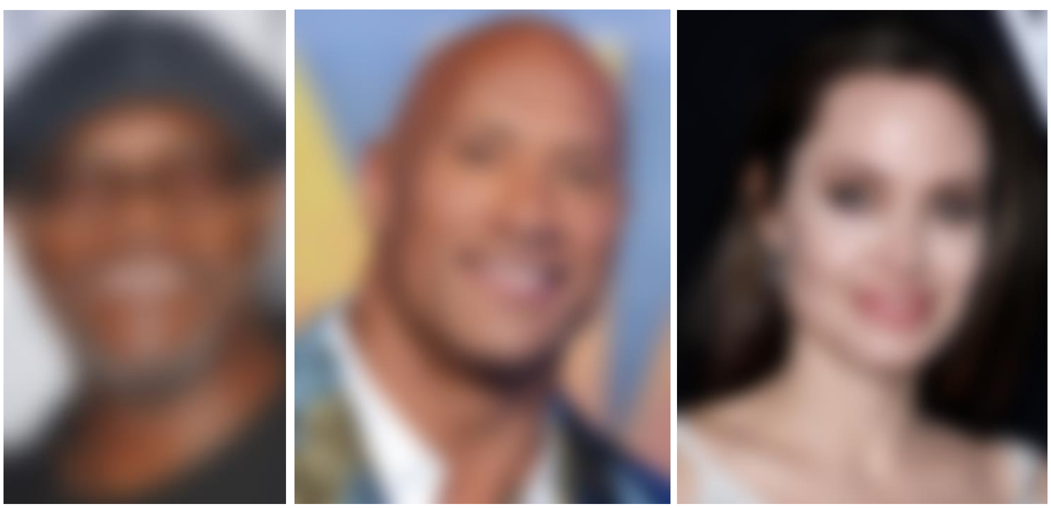
در این سوال می‌خواهیم تصاویری به نام Hybrid Image بسازیم. این تصاویر از ترکیب فرکانس‌های پایین یک تصویر و فرکانس-های بالای تصویر دیگر ساخته می‌شوند. در این سوال قرار است این تصاویر چهره افراد باشند اما لزومی ندارد حتما از چهره افراد استفاده کنیم بلکه هر نوع تصویری را می‌توان با این تکنیک ترکیب کرد. تصویر زیر یک نمونه از وبسایت Wikipedia است که متن های Northwest و Southwest را با هم ترکیب کرده است. اگر از صفحه دور شویم یا چشم های خود را ببندیم (نه کامل!) متن فرکانس پایین یعنی Northwest را میبنیم و در حالت عادی یا از نزدیک، تصویر فرکانس بالا یا Southwest مشاهده می‌شود.

همانطور که گفتیم، در این سوال قرار است این عمل را بر روی تصاویر چهره افراد انجام دهیم. برای اینکه بتوانیم بهترین نتیجه ممکن را از این روش بگیریم بهتر است چند نکته را رعایت کنیم.

i. بهتر است تصاویر افراد مختلف از زاویه تقریبا برابر گرفته باشند.

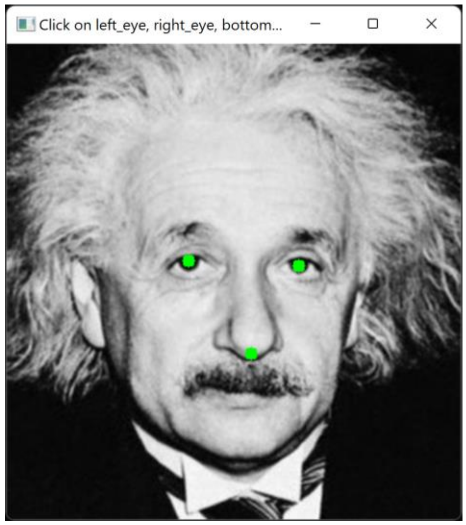
ii. بهتر است تصاویر افرادی را انتخاب کنیم که تقریبا به هم شباهت داشته باشند.

iii. بهتر است تصاویر افرادی را انتخاب کنیم که **معروف** باشند، تا بتوان از روی تصویر محوی نیز آنها را شناخت.



a) همانطور که در بخش قبل گفتیم بهتر است تصاویر انتخابی از چهره افراد تقریبا از یک زاویه گرفته شده باشند اما با این حال بسیار سخت است که بتوان دو تصویری پیدا کرد که دقیقا بر هم منطبق باشند. بنابراین در این مرحله سعی می‌کنیم با استفاده از یک سری تبدیلات هندسی، تصاویر دو فرد را بر هم منطبق کنیم.

برای راحت شدن روند پیدا کردن مختصات نقاط، یک Script جانبی با رابط گرافیکی نوشته شد تا بتوان مختصات دو چشم و دماغ را با کلیک کردن بر روی آنها پیدا کرد.

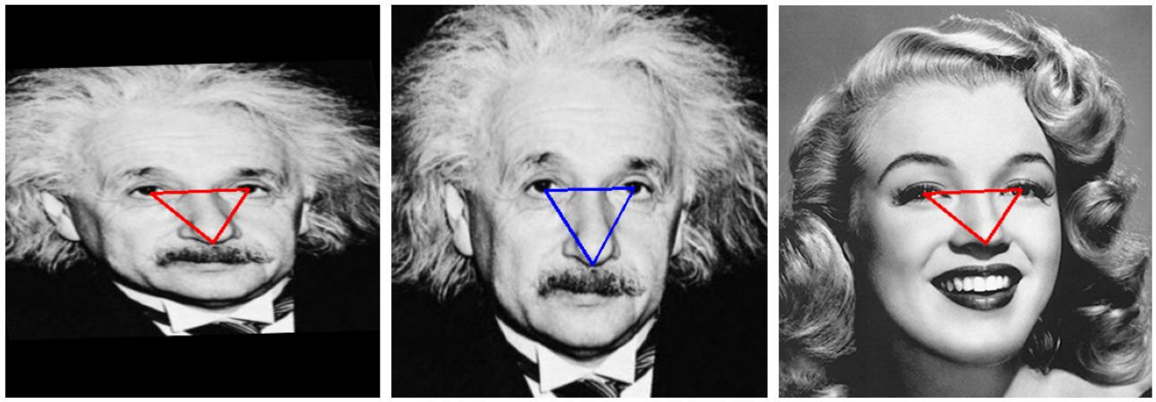



```
File name: inputs\P5\ein.png

points[1,0,:] = [145, 173] # l_eye a
points[1,1,:] = [233, 177] # r_eye a
points[1,2,:] = [195, 247] # nose_a

points[0,0,:] = [145, 173] # l_eye b
points[0,1,:] = [233, 177] # r_eye b
points[0,2,:] = [195, 247] # nose_b
----------------------------------------
```

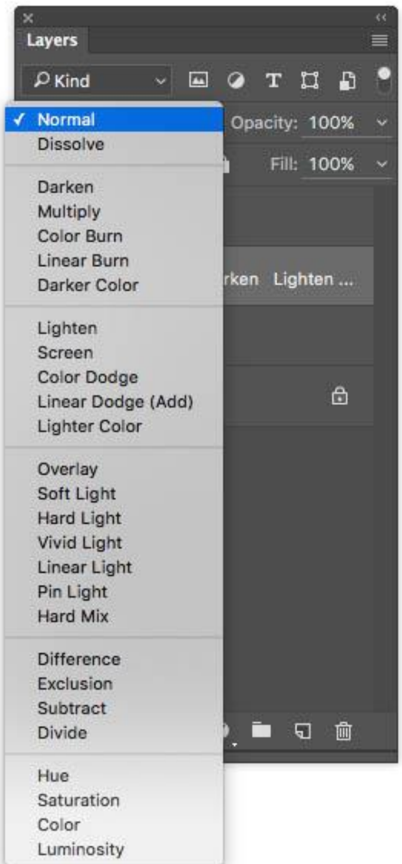
پس از کلیک کردن بر روی نقاط مورد نظر، برنامه این مختصات را به ساختاری که در کد اصلی استفاده میکنیم در می‌آورد و آماده انتقال به آن کد می‌باشد.

بعد از انجام دادن این کار برای تمام افراد، این نقاط نظیر را به تابع cv2.getAffineTransform می‌دهیم و ماتریس آفین مربوط به انتقال نقطه های سری ۲ به سری ۱ را دریافت میکنیم. سپس این ماتریس را به تابع cv2.warpAffine می‌دهیم و تبدیل شکل می‌گیرد. در این مرحله بجای استفاده از Affine Transform میتوانستیم از Perspective Transform نیز استفاده کنیم که طبق آزمایش های انجام شده، تصاویر را به شکل نا مطلوبی Distort می‌کرد.

در تصویر های زیر می‌توانید این تبدیل را در عمل مشاهده کنید.



b) پس از Align کردن تصاویر افراد به صورت دو به دو نوبت به آن رسیده که فیلتر های High-pass و Low-pass را به تصاویر اعمال کنیم. ابتدا یک تصویر را به عنوان پس زمینه انتخاب میکنیم و فیلتر Gaussian Blur را به آن اعمال میکنیم. با این کار اطلاعات High-frequency از تصویر حذف می‌شوند. سپس فیلتر Gaussian Blur را به تصویر دوم اعمال میکنیم و حاصل را از تصویر اصلی کم میکنیم. با این کار عملا یک فیلتر High-pass به تصویر اعمال کردیم که فقط اطلاعات High-frequency را حفظ می‌کند.



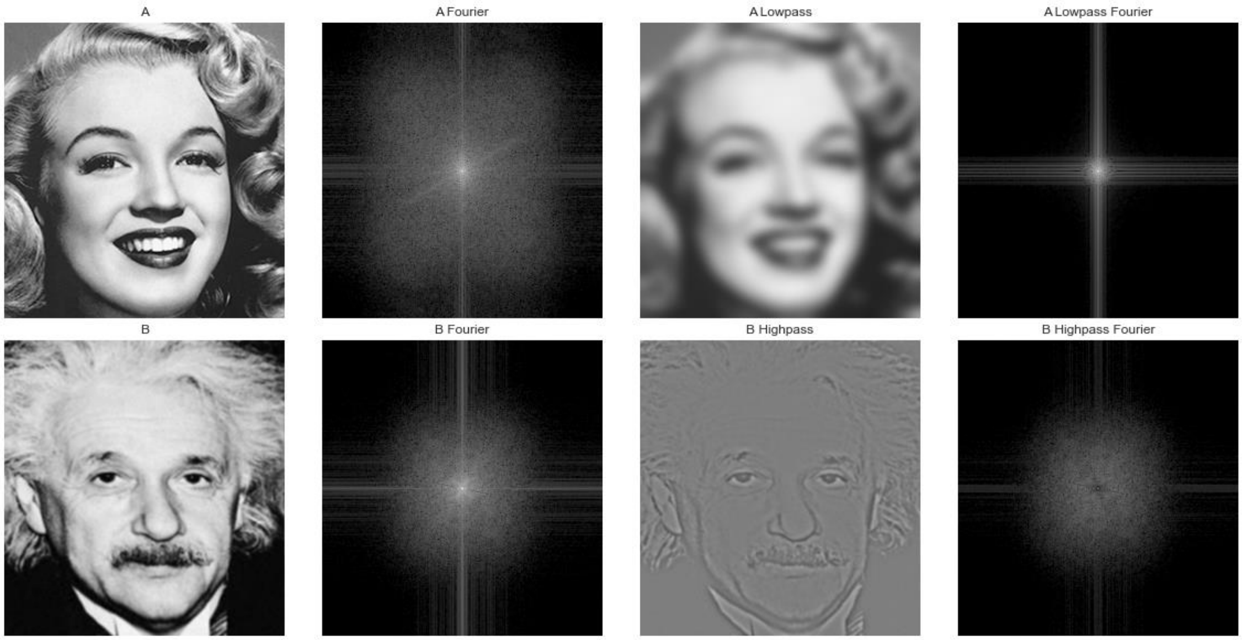
همانطور که می‌دانید برای ترکیب دو تصویر تعداد بی‌شمار روش وجود دارد(جمع، ضرب، میانگین، ...).

با توجه به آزمایش های انجام شده و تجربه قبلی در نرم افزار فوتوشاپ، حالت ترکیب **Linear Light** بهترین نتیجه را برای این کاربرد داشت. رابطه ریاضی مربوط به این حالت ترکیب را در معادله زیر میتوانید مشاهده کنید.

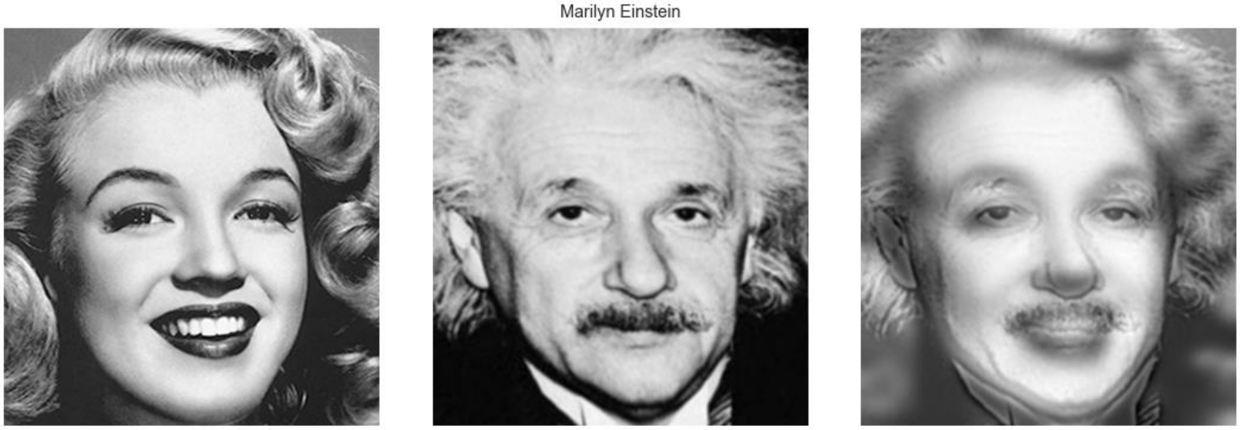
$$Linear\_Light(A.B) = 2 * A - 1 + B$$

تصاویر نتایج را در صفحات بعد مشاهده میکنید.

Einroe

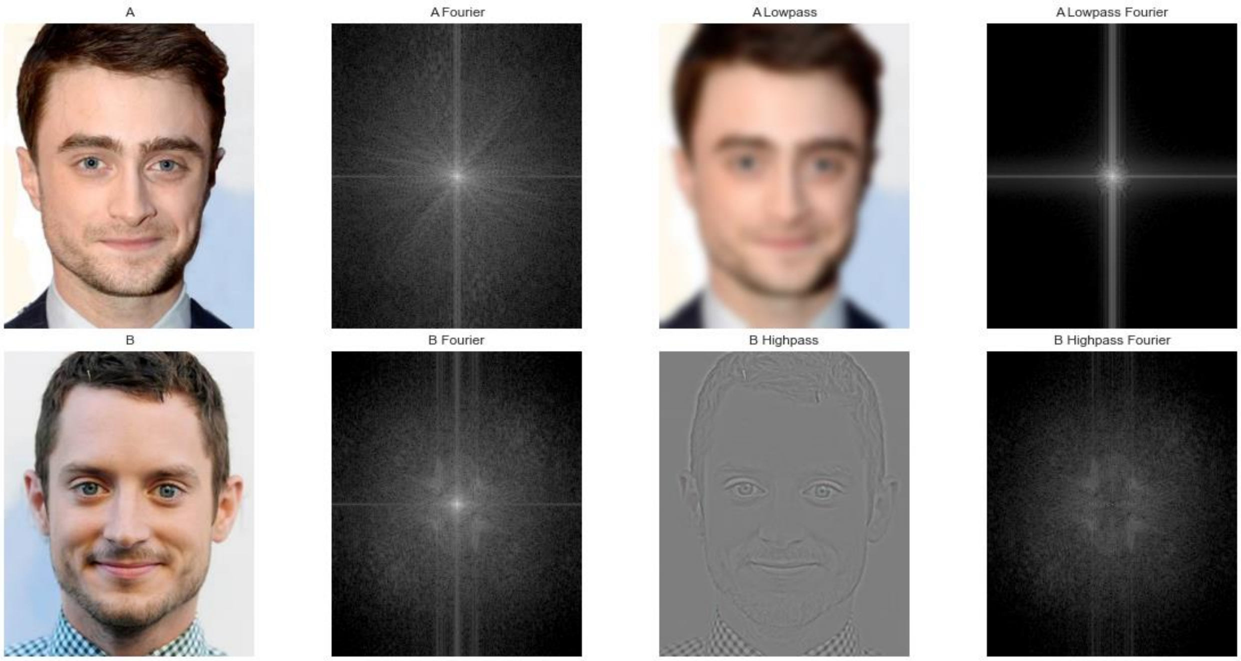


Marilyn Einstein



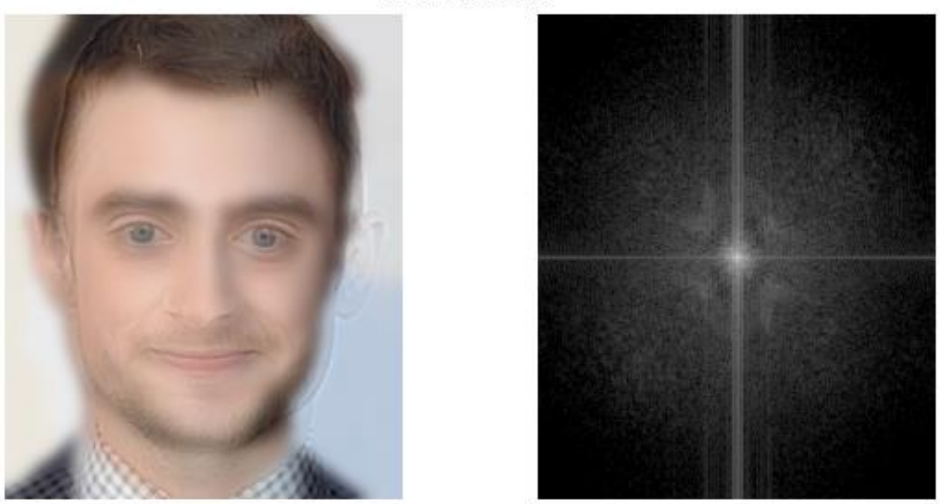
Marilyn Einstein

Danijah

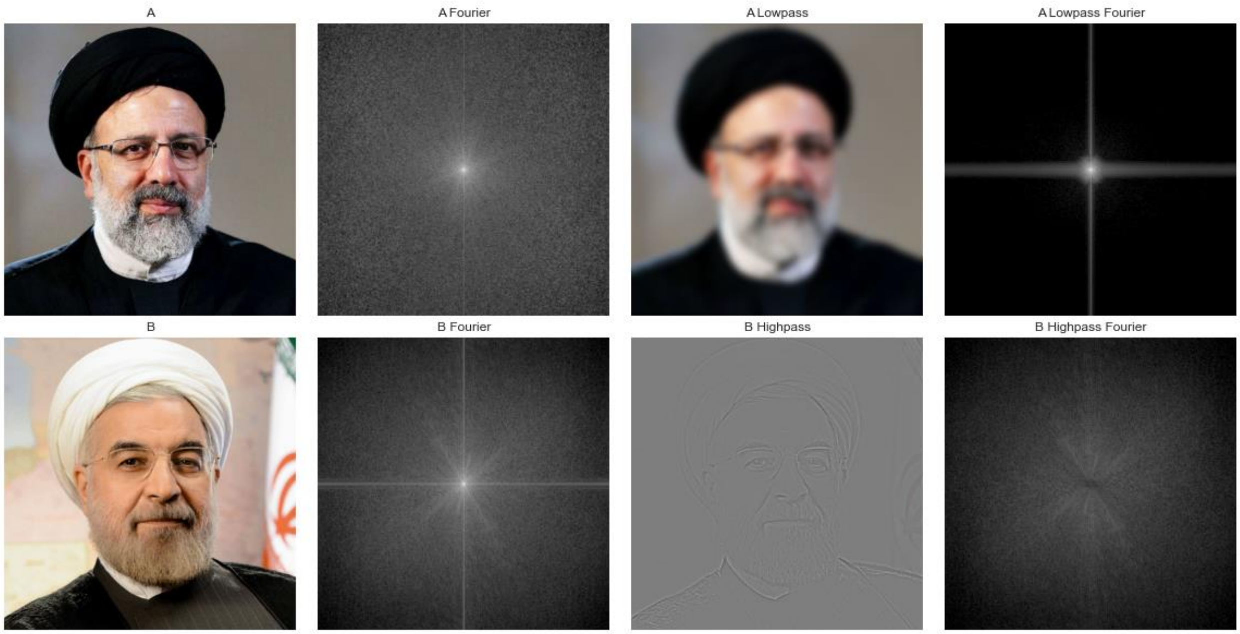


Daniel Woods

Daniel Woods



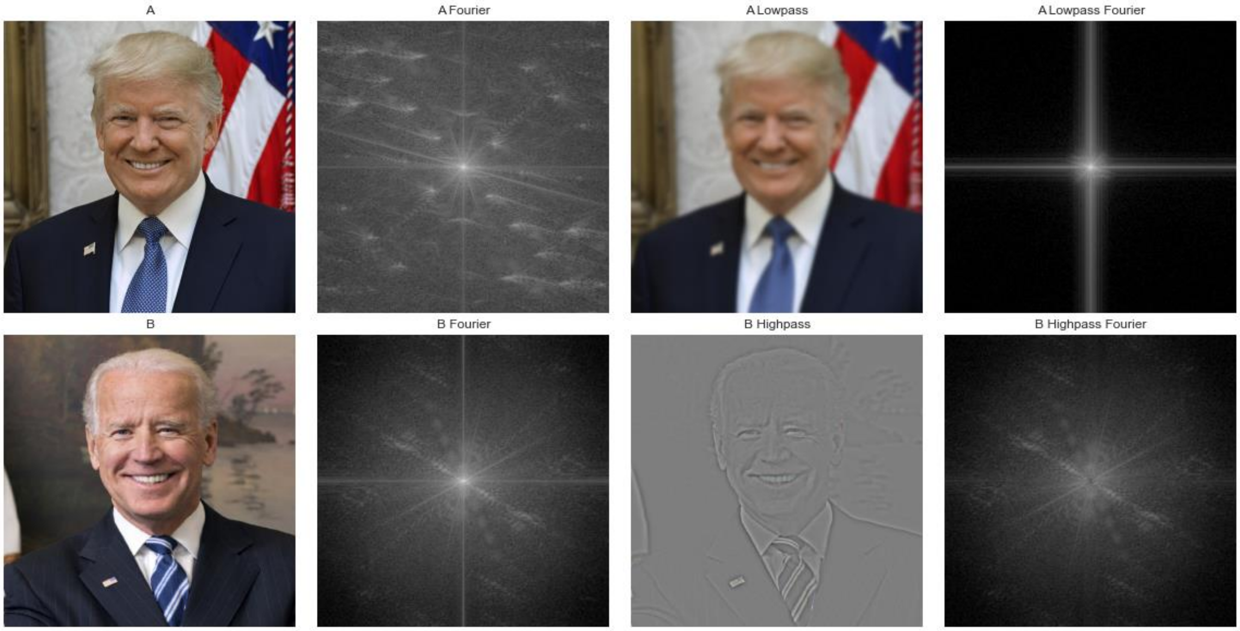
Raihani



Ebrahim Rouhani

Ebrahim Rouhani

Triden



Donald Biden

Donald Biden

نکته ای در مورد Align کردن: در تصویر Trump و Biden بهتر ا ست به‌جای Align کردن چشم ها و دماغ، چشم ها و دهن را Align کنیم. نتیجه این دو نوع Align کردن را در تصاویر زیر مشاهده می‌کنید. در این تصویر خاص، با توجه به اینکه دهان هر دو شخص باز ا ست و دندان های آنها م شخص ا ست و همچنین می‌دانیم دندان‌ها جزییات زیادی دارند بهتر ا ست اولویت با Align شدن دهان ها باشد تا دماغ ها.

Align کردن با دماغ

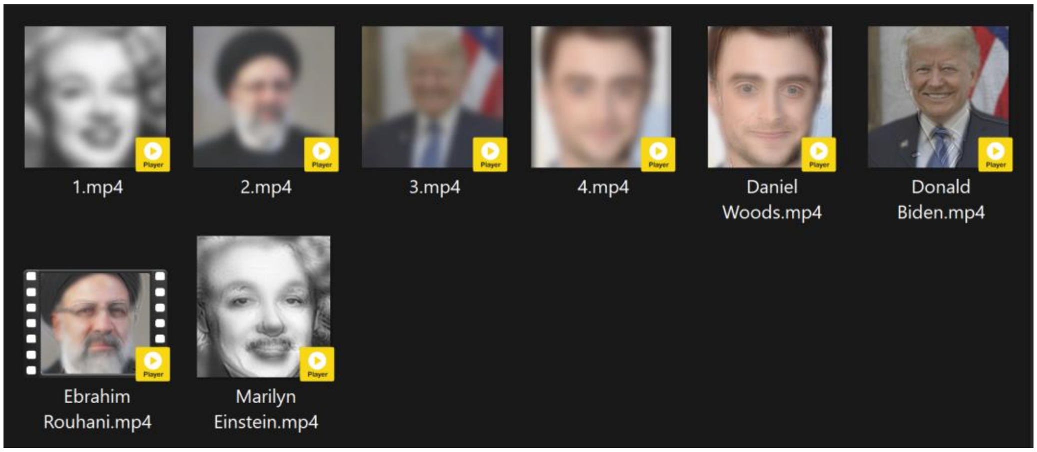
Align کردن با دهان

d) برای بخش Visualization این پدیده چندین راه وجود دارد. اگر در حال مطالعه PDF این گزارش هستید می‌توانید با زوم اوت کردن صفحه، تصاویر **Low-pass** را مشاهده کنید و با زوم این کردن، تصاویر **High-pass** را بیشتر مشاهده کنید. راه دیگر تنگ(!) کردن چشم‌هاست. به این صورت که با تنگ کردن چشم‌ها می‌توان تصویر **Low-pass** را مشاهده کرد.

راه سوم Blur کردن تصویر حاصل است. هر چه تصویر بیشتر Blur شود، تصویر Low-pass خود را بیشتر نشان می-دهد. نمایش این حالت در فایل PDF امکان پذیر نیست بنابراین برای دیدن اثر Blur شدن تصویر حاصل به صورت **ویدیویی**، به پوشه مربوط به این سوال مراجعه کنید.

نتیجه مربوط به ترکیب چهره Marilyn Monroe و Albert Einstein

ویدیوها در دو حالت Blur شدن تدریجی و زوم اوت شدن تدریجی ساخته شده اند.



سوال ۶)

a) می‌دانیم تبدیل فوریه یک تابع زوج، یک تابع زوج است و فوریه یک تابع فرد، فرد است. اثبات:

$$F(\omega) = \int e^{jx\omega}\, f(x)\, dx$$

تغییر متغیر $y = -x$ را اعمال میکنیم:

$$F(-\omega) = \int e^{-jx\omega}\, f(x)\, dx = \int e^{jy\omega}\, f(-y)\, dy$$

اگر تابع f زوج باشد:

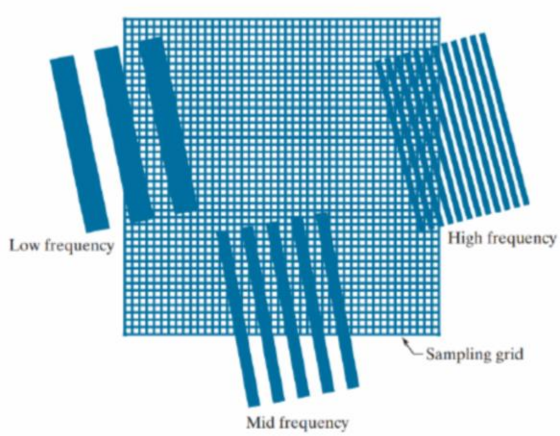
$$F(-\omega) = \int e^{jy\omega}\, f(-y)\, dy = \int e^{jy\omega}\, f(y)\, dy = F(\omega)$$

اگر تابع f فرد باشد:

$$F(-\omega) = \int e^{jy\omega}\, f(-y)\, dy = \int e^{jy\omega}(-f(y))\, dy = -F(\omega)$$

همچنین می‌دانیم، تبدیل کانولوشن دو تابع در حوزه مکان، برابر با ضرب فوریه آن دو تابع در حوزه فرکانس است. بنابراین اگر یک تابع زوج را با یک تابع فرد Convolve کنیم، مثل این است که یک تابع زوج را در یک تابع فرد در حوزه فوریه ضرب کنیم. واضح است که ضرب دو تابع زوج و فرد همواره فرد است. سپس نیاز است از این تابع فرد بدست آمده، معکوس فوریه بگیریم که اثبات کردیم اگر تابع در حوزه فرکانس فرد باشد، در حوزه مکان هم فرد است بنابراین حاصل کانولوشن دو تابع زوج و فرد، همواره یک **تابع فرد** است.

b) اگر تصویر دارای جزئیاتی با فرکانس بیشتر از نصف فرکانس نمونه گیری باشد Aliasing رخ می‌دهد. در مثال تصاویر دیجیتال، فرکانس نمونه برداری برابر با $1\,\frac{\text{cycle}}{\text{pixel}}$ است. بنابراین اگر پترن متناوبی با فرکانس0.5 یا بیشتر داشته باشیم Aliasing رخ می‌دهد. با توجه به تصویری که در کتاب موجود است، خطوط ایجاد شده توسط Aliasing در جهت متفاوتی ایجاد می‌شوند و فاصله خطوط کمتر به نظر می‌رسد.



c) وقتی یک تصویر را به روش اول Pad میکنیم، پس از شیفت و گرفتن تبدیل فوریه، فرکانس های پایین به مرکز تصویر می‌آیند. اما با روش دوم، پس از اعمال شیفت و گرفتن تبدیل فوریه، در مرکز تصویر ۰ خواهد بود و فرکانس های پایین در جای دیگری قرار دارند. بنابراین باید مراقب باشیم این دو نوع تصویر را به یک شکل فیلتر نکنیم. فیلتر های یکسان، بر روی این دو تصویر نتایج متفاوتی دارند.

d) طبق آزمایش‌های انجام شده، ضرب کردن فاز در ۱- باعث فلیپ شدن تصویر در دو جهت افقی و عمودی می‌شود. برای اثبات این پدیده می‌توان فاز را در منفی یک ضرب کرد و در رابطه فوریه معکوس قرار داد. خواهیم دید که تابع بدست آمده در اثر فوریه معکوس، در پشت x,y علامت منفی دارد.

```python
fourier_img = np.fft.fft2(img.astype(np.float32))
fourier_img_shift = np.fft.fftshift(fourier_img)

real = fourier_img_shift.real
imag = fourier_img_shift.imag

magnitude = np.abs(fourier_img_shift)
phase = np.angle(fourier_img_shift) * -1

combined = magnitude*np.exp(1j*phase)
combined_shift_back = np.fft.ifftshift(combined)
img_back = np.fft.ifft2(combined_shift_back)

disp(img_back.real)
```

[31] 0.3s