

به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

تمرین درس پردازش تصویر-سری سوم

فردين آيار

شماره دانشجویی: ۹۹۱۳۱۰۴۰

استاد: دکتر رحمتی

دانشکده کامپیوتر- زمستان ۹۹

(l)

: (a) می دانیم $g(t-t_0) \Leftrightarrow e^{-jut_0}G(u)$ بنابراین:

$$g(t) = \sin(t-2) \Leftrightarrow G(u) = e^{-2ju} \times \frac{1}{2j} [2\pi\delta(u-1) - 2\pi\delta(u+1)]$$

(b)

$$g(t) = \frac{2t^2 + 1}{t^2 + 1} = 2 - \frac{1}{t^2 + 1} \Leftrightarrow G(u) = 2\delta(u) - \pi e^{-|u|}$$

(c)

$$g(t) = \delta(t-1) + \delta(t+2) = e^{-ju} + e^{2ju}$$

(d)

(e)

(f)

$$g(t) = \cos(4\pi t)\cos(6\pi t) \Leftrightarrow \frac{1}{2} \sqrt{\frac{\pi}{2}} [\delta(u-10\pi) + \delta(u-2\pi) + \delta(u+2\pi) + \delta(u+10\pi)]$$

(g) تبدیل فوریه این سیگنال $e^{-j2\pi u}$ می باشد که بخش حقیقی آن صفر نیست (شرط g برقرار نیست).

(h) این سیگنال حقیقی و فرد است بنابراین تبدیل فوریه آن موهومی و فرد خواهد بود (شرط h برقرار نیست).

(i) تبدیل فوریه این سیگنال فرد خواهد بود بنابراین شرط i را دارد (اما شرط h را ندارد).

(j)

(k) تبدیل فوریه یک سیگنال متناوب، متناوب نخواهد بود؛ بنابراین این سیگنال شرط k را ارضاء نمی کند.

(l) فقط به ازای آلفا صفر، این سیگنال شرط l را دارد.

(m)

$$f = [1, 3, -1, 2, 0, -3, 0, 0, 0, 0, 0, 0]$$

$$h = [-1, 3, -2, 0, 0, 0, 0, 0, 0, 0, 0]$$

$$f^*h = [-1, 0, 8, -11, 8, -1, -9, 6, 0, 0, 0, 0]$$

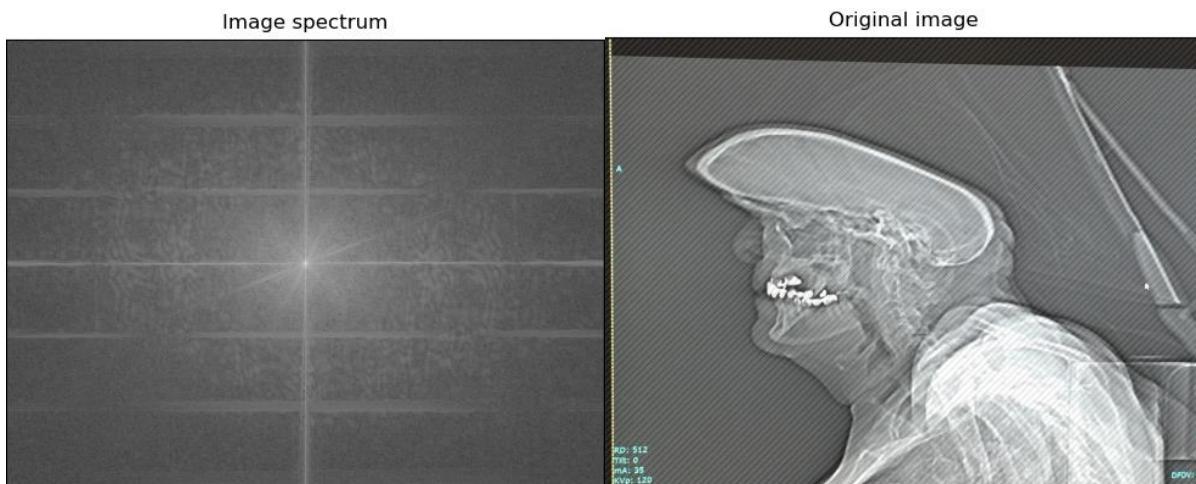
(n)

(o)

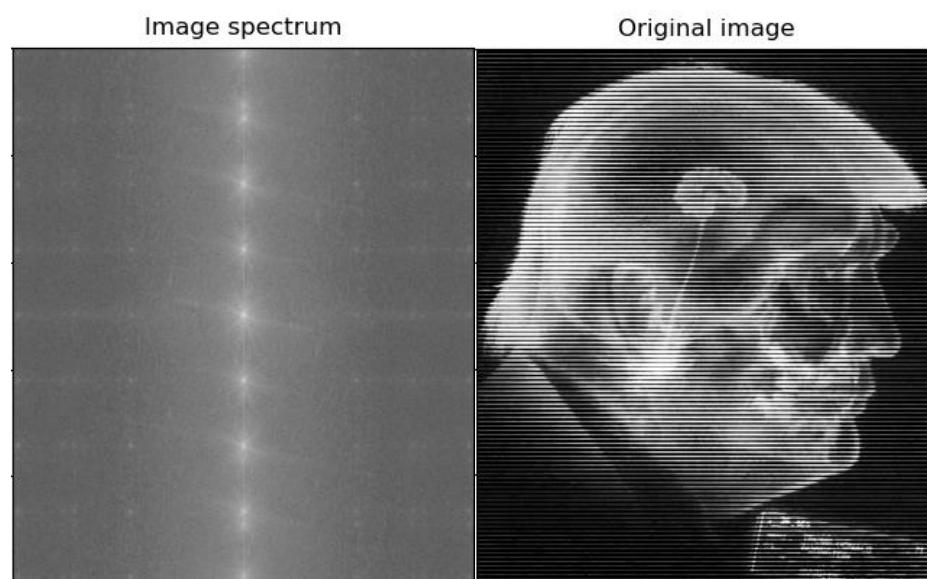
(p)

(a) کد مربوط به این قسمت در فایل `a.py` قرار دارد. تصاویر و طیف تبدیل فوریه آن‌ها به ترتیب در تصاویر ۱-۲ و ۲-۲ نمایش داده شده است. در تبدیل فوریه تصویر اول، نقاط مربوط به طرح مویر دیده نمی‌شود. اگرچه می‌دانیم تبدیل فوریه یک سری خطوط موازی، یک خط عمود بر آن هاست؛ بنابراین احتمالاً تعریف یک فیلتر قطری که محدوده فرکانس‌های بالا را حذف می‌کند مناسب است.

در تبدیل فوریه تصویر دوم، نقاط مربوط به طرح مویر کاملاً مشهود است. بنابراین کافی است یک فیلتر که فرکانس‌های جهت عمودی را حذف می‌کند، انتخاب کنیم.

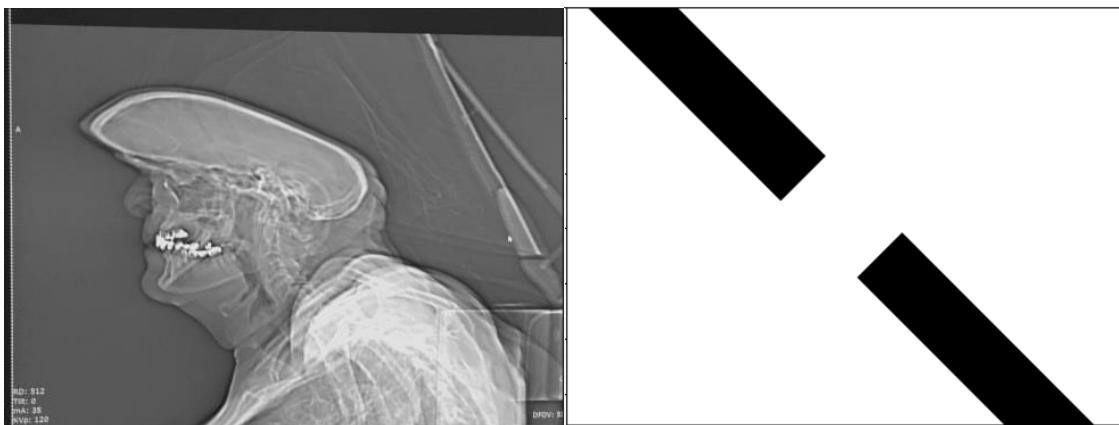


شکل ۱-۲

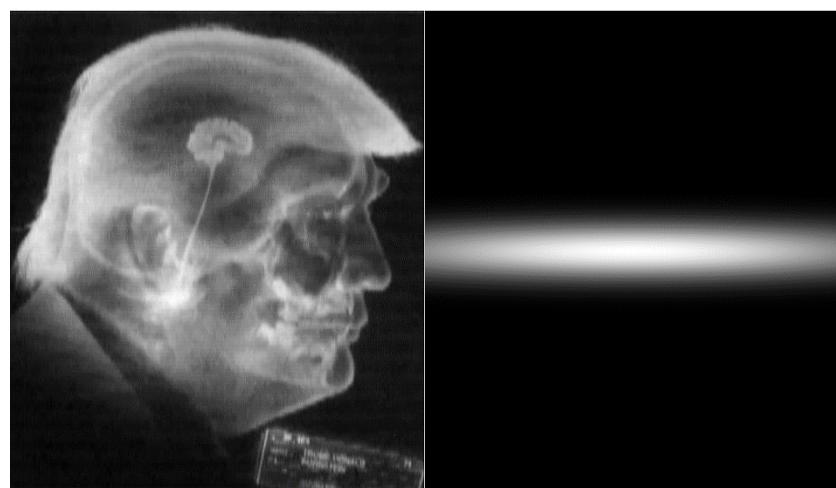


شکل ۲-۲

(b) کد مربوط به این قسمت در فایل `b.py` قرار دارد. با توجه به توضیحات قسمت قبل، فیلتر انتخاب شده و تصویر خروجی در شکل‌های ۳-۲ و ۴-۲ نمایش داده شده است.

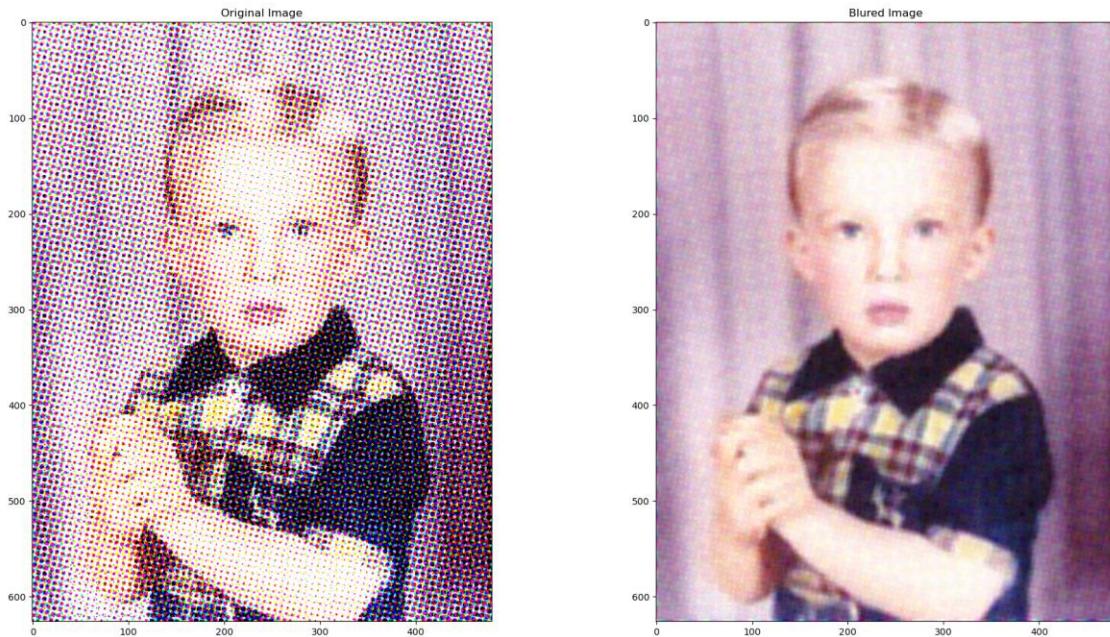


شکل ۳-۲

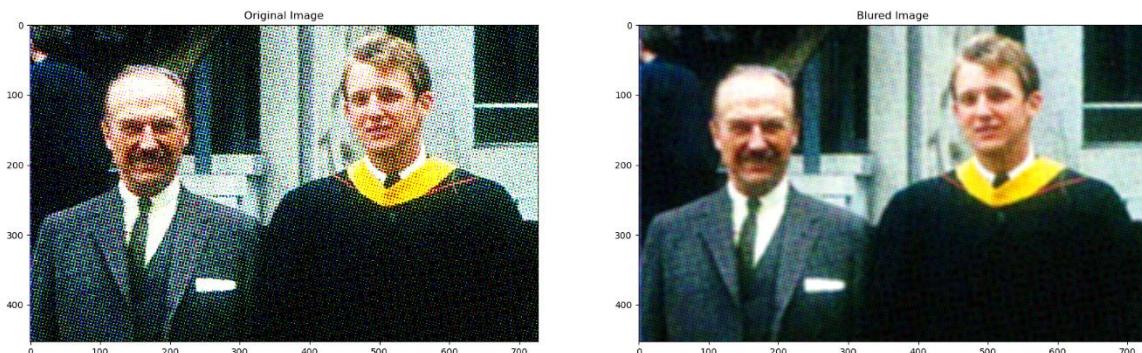


شکل ۴-۲

۴) کد مربوط به این قسمت در فایل `C.py` قرار دارد. از فیلتر گاووسی با سایز ۷ و سیگما ۲۰ استفاده می‌کنیم. نتایج در شکل‌های ۵-۲ و ۶-۲ نمایش داده شده است.

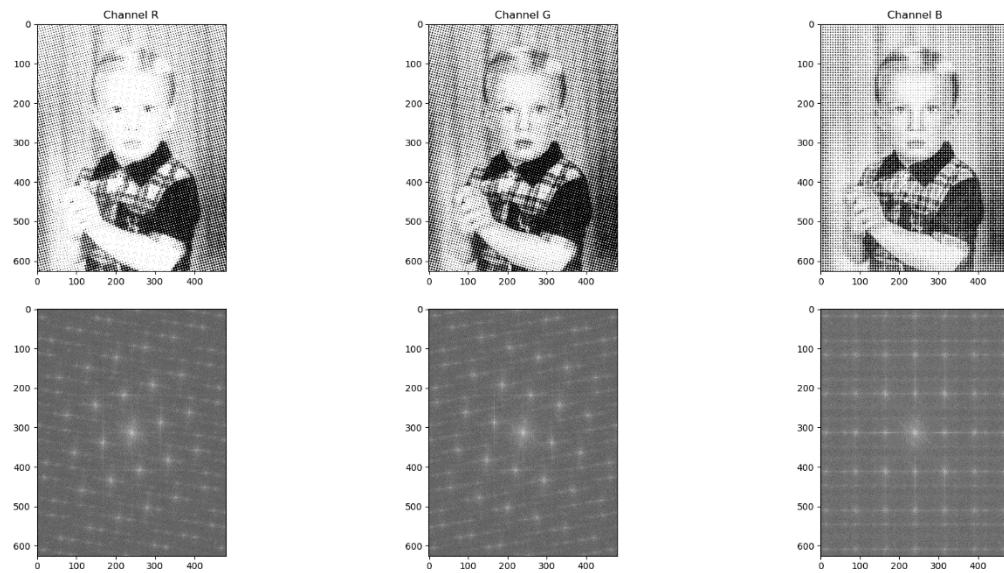


شکل ۲

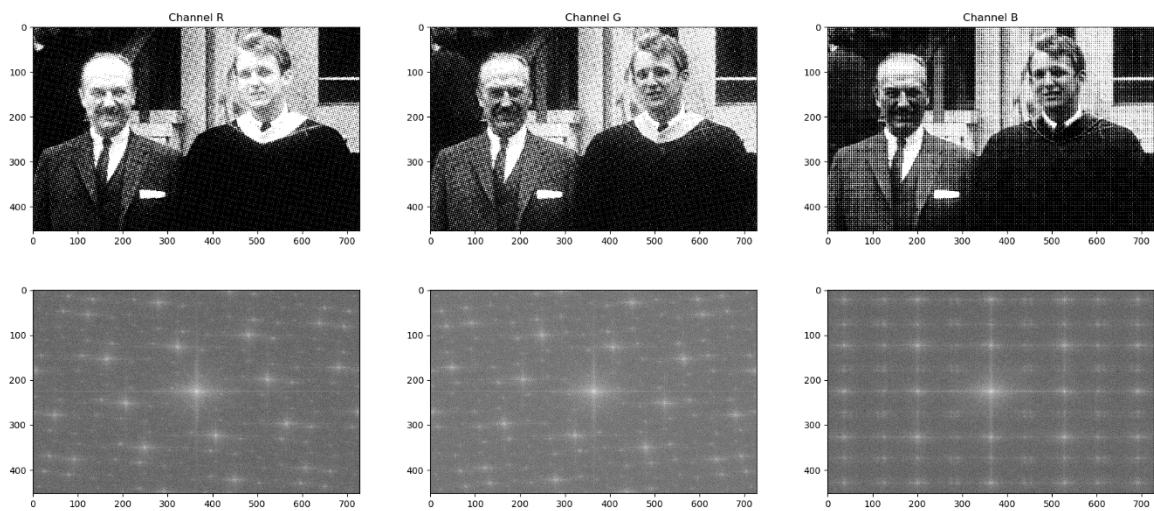


شکل ۳

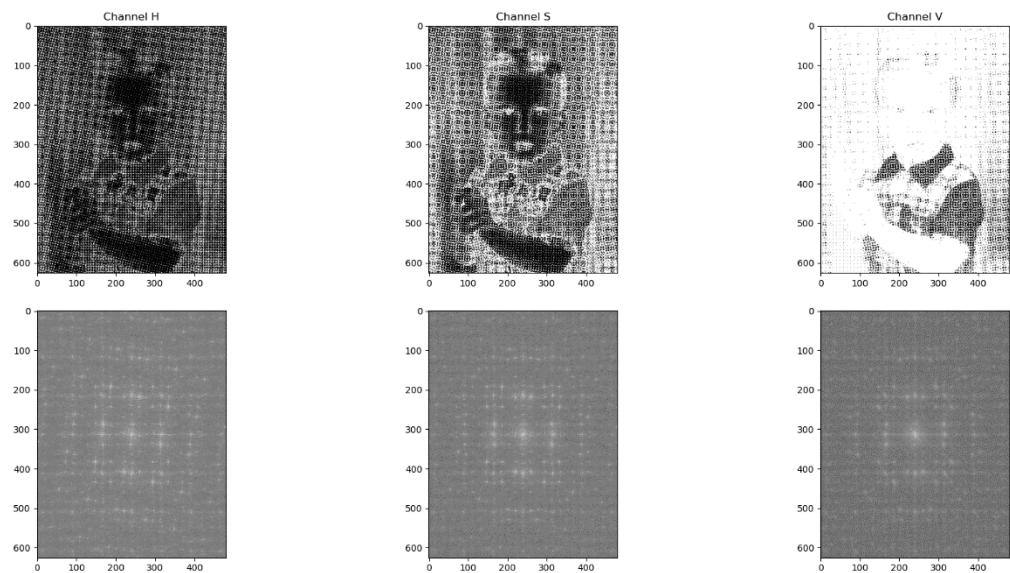
(d) کد مربوط به این قسمت در فایل `d.py` قرار دارد. خروجی‌ها در شکل‌های ۷-۲ تا ۱۴-۲ نمایش داده شده‌است. با توجه به این تصاویر، به نظر می‌رسد فضای رنگ RGB به علت اینکه نقاط روشن فرکانس‌های بالا در آن از هم فاصله بیشتری دارند، بهتر است.



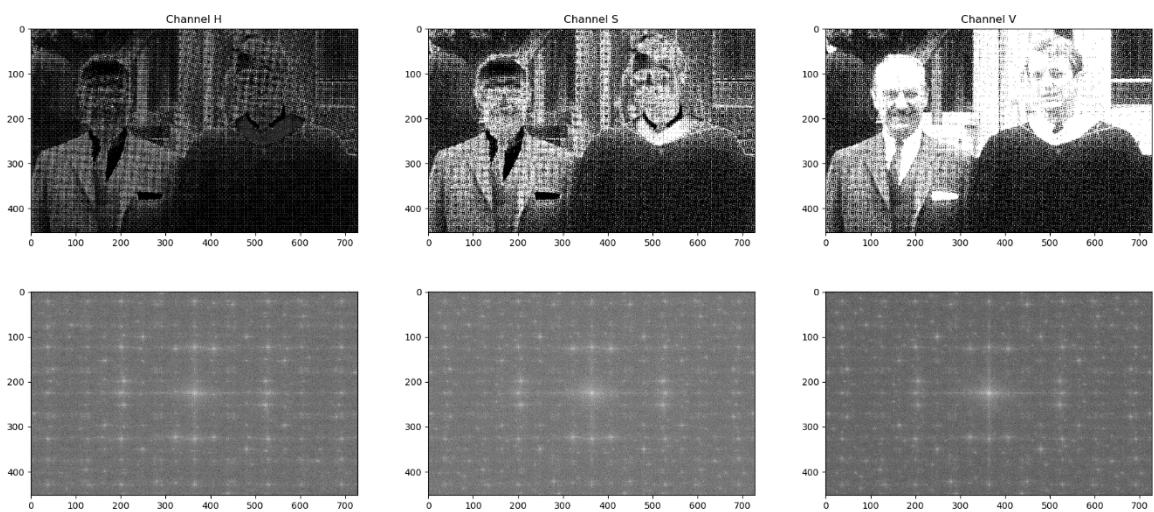
٤-٣ مُكَبَّل



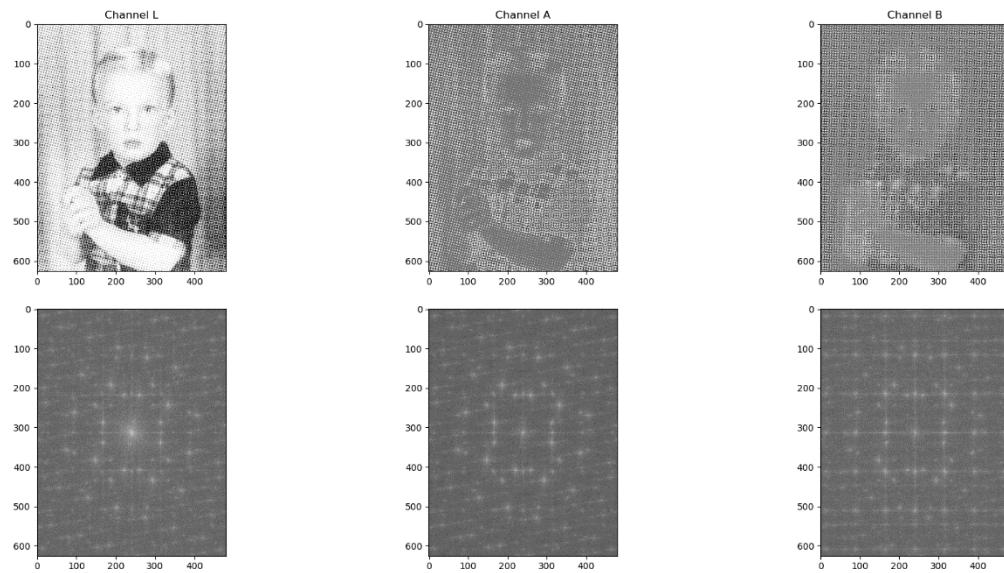
٤-٤ مُكَبَّل



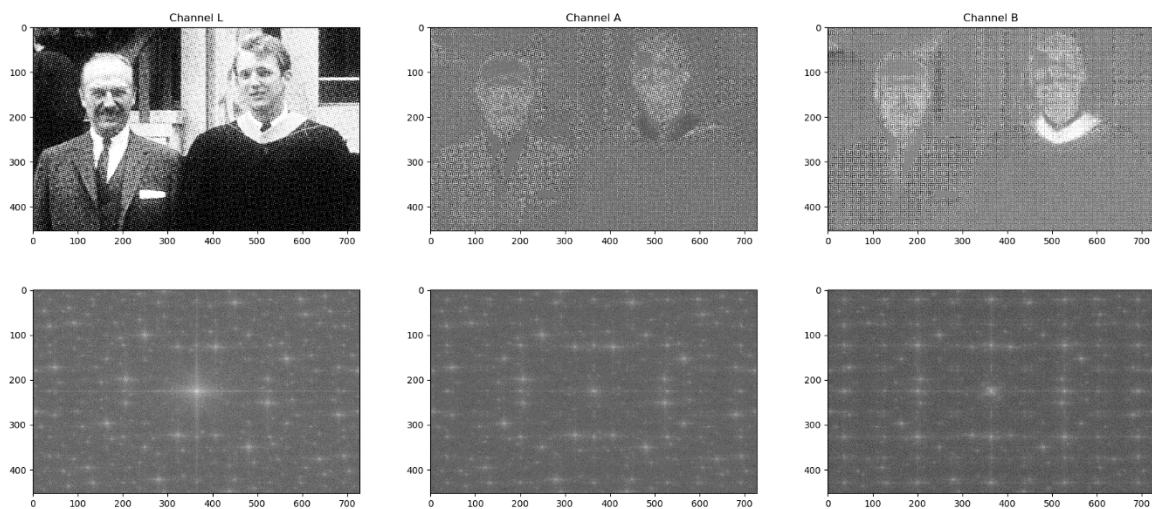
شکل ۹-۲



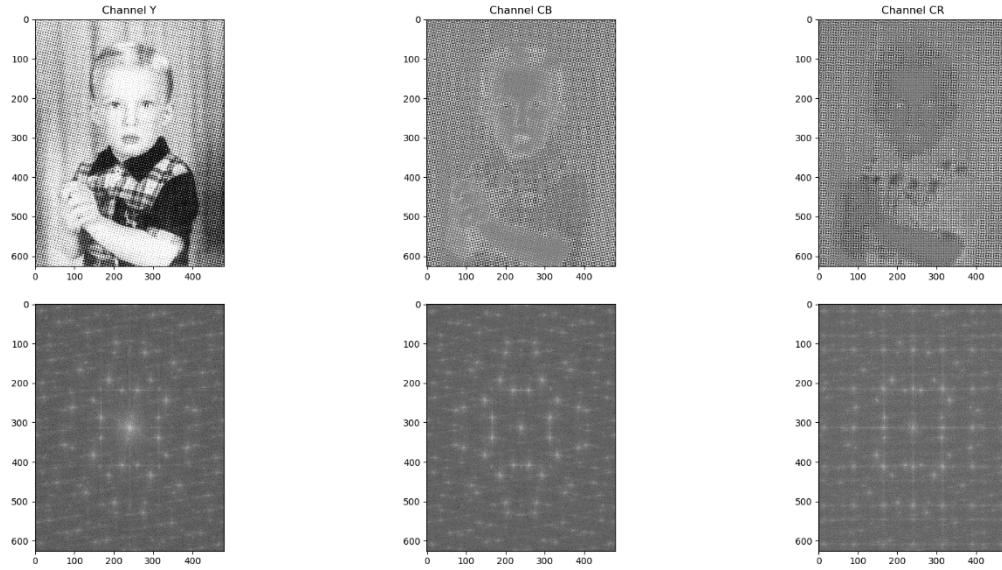
شکل ۱۰-۲



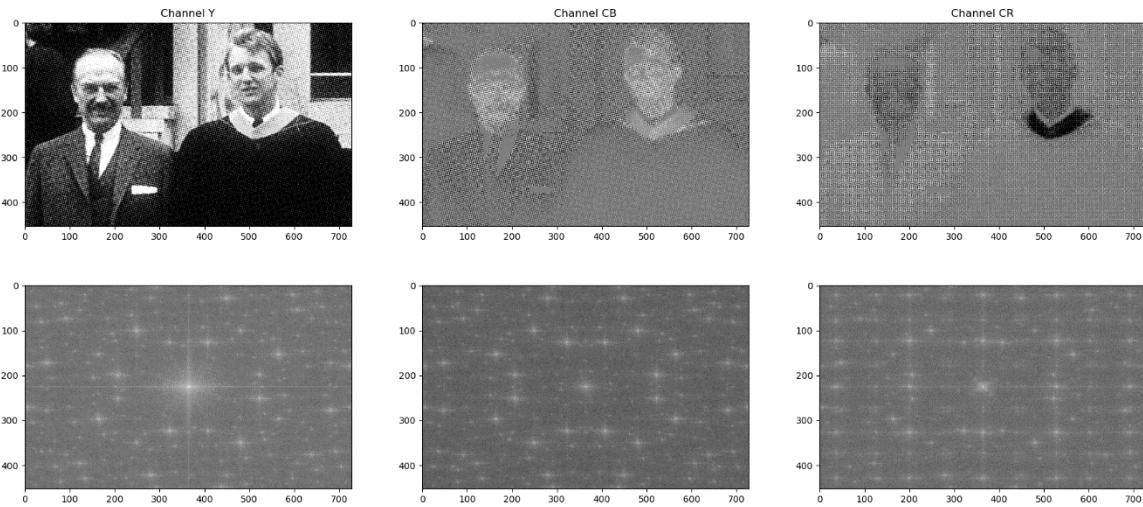
شكل ١١-٣



شكل ١٢-٣



شکل ۱۳-۲

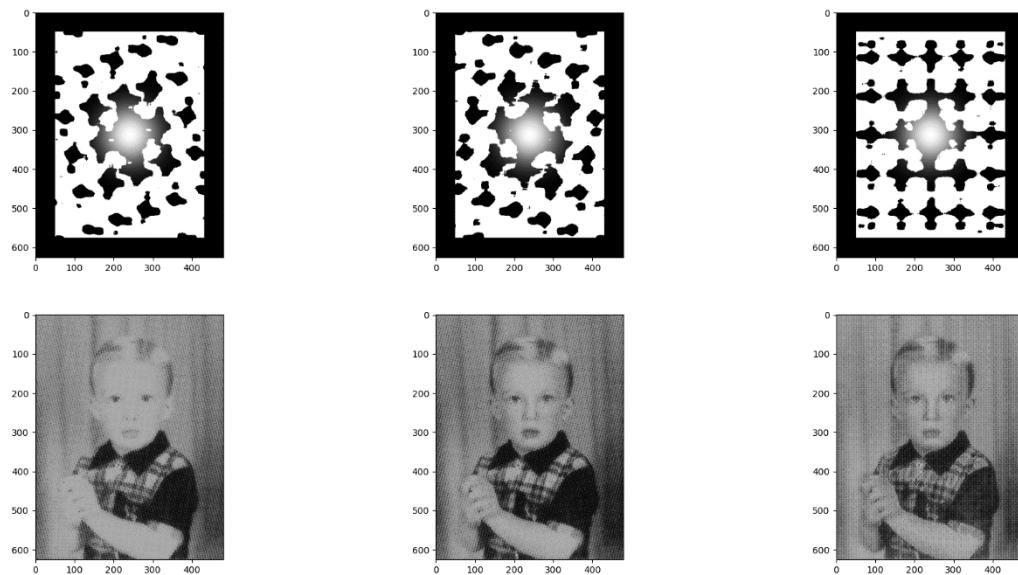


شکل ۱۴-۲

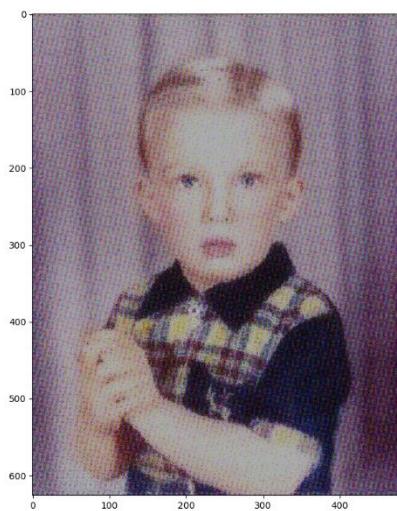
(e) کد مربوط به این قسمت در فایل `e.pv` قرار دارد. سعی می‌کنیم به جای تعریف دستی فیلتر، با یک روش خودکار فیلتر را ایجاد کنیم. روش کار به این صورت است:

- (۱) بر روی تصویر طیف فوریه، یک فیلتر بلور با ابعاد 20×20 در اعمال می‌کنیم.
- (۲) در تصویر حاصل از مرحله قبل، پیکسل‌های با مقدار بیشتر از 40 را صفر و بقیه را یک می‌کنیم.
- (۳) مرحله قبل باعث حذف فرکانس‌های پایین نیز می‌شود، بنابراین یک فیلتر گاوسی با سیگما 5 به نتیجه مرحله قبل اضافه می‌کنیم.
- (۴) در تصویر حاصل از مرحله قبل، لبه‌ها (فرکانس‌های بسیار بالا) را صفر می‌کنیم.

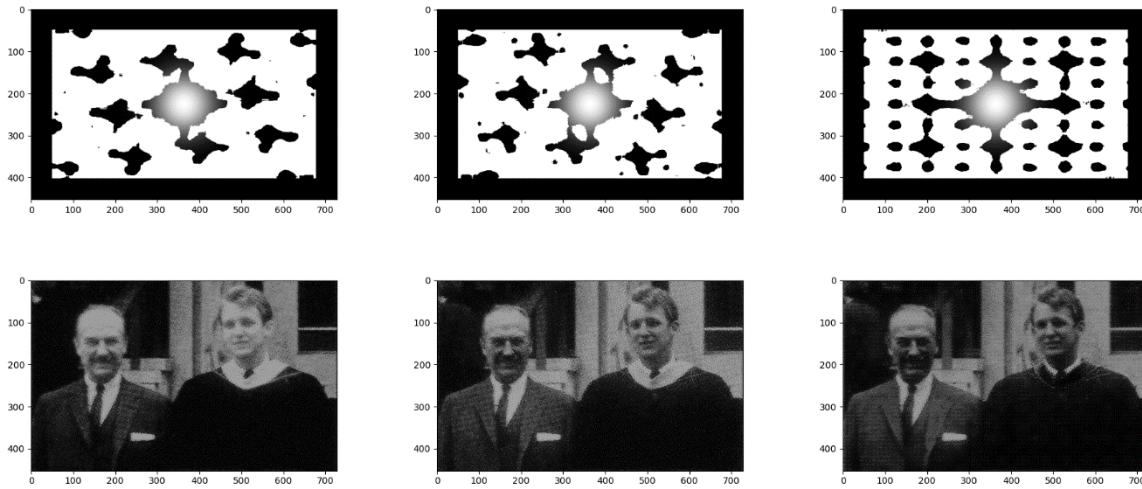
در شکل‌های ۱۵-۲ و ۱۷-۲ فیلترهای بدست آمده از روش فوق و همچنین نتیجه اعمال آن بر کاتال مربوطه، برای هر دو تصویر نمایش داده شده است. همچنین تصاویر نهایی در شکل‌های ۱۶-۲ و ۱۸-۲ ارائه شده‌اند.



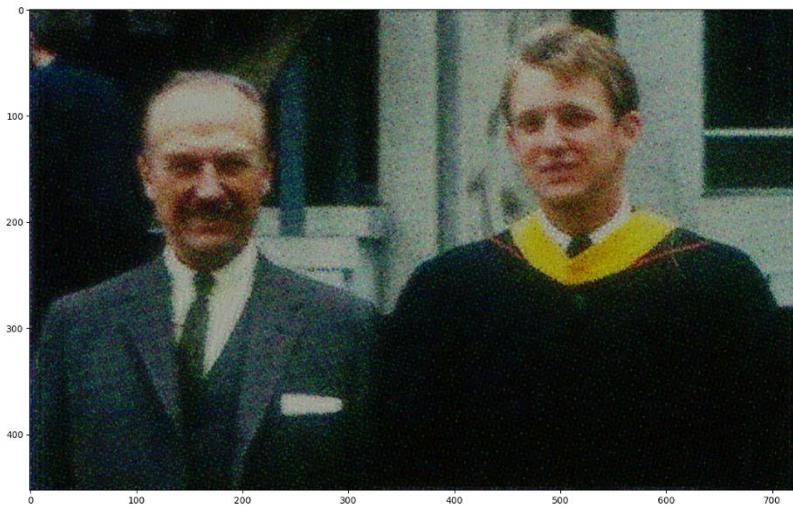
شکل ۲-۱۵



شکل ۲-۱۶



شکل ۱۷-۲



شکل ۱۸-۲

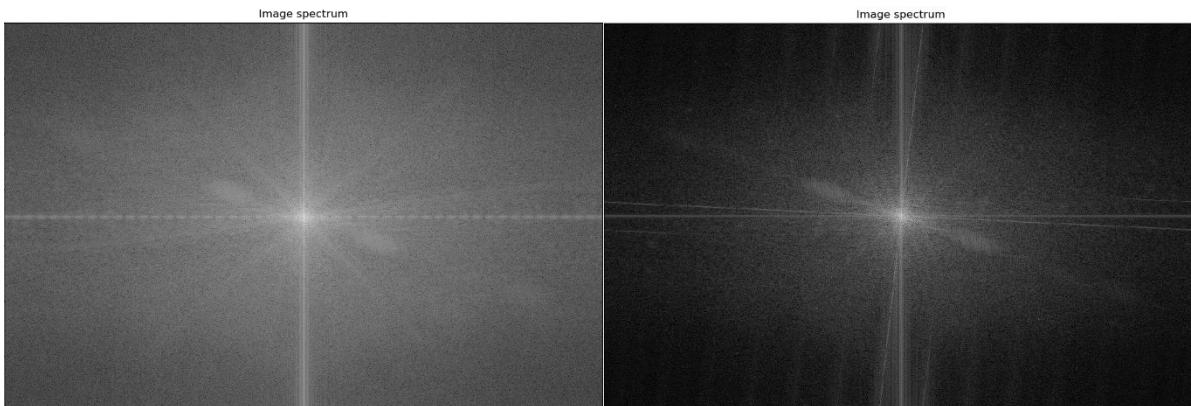
(f) در نتیجه بهبود تصویر در دامین مکان، الگوی هافتون کمتر دیده می‌شود اما بهبود تصویر به روش انجام شده در دامین فرکанс، لبه‌ها را بهتر حفظ کرده است.

(۳)

(a) کد مربوط به این قسمت در فایل `a.py` قرار دارد. دو جفت تصویر تراز شده به ترتیب در شکل‌های ۱-۳ و ۳-۳ و تبدیل فوریه آن‌ها در شکل‌های ۲-۳ و ۴-۳ نمایش داده شده است.



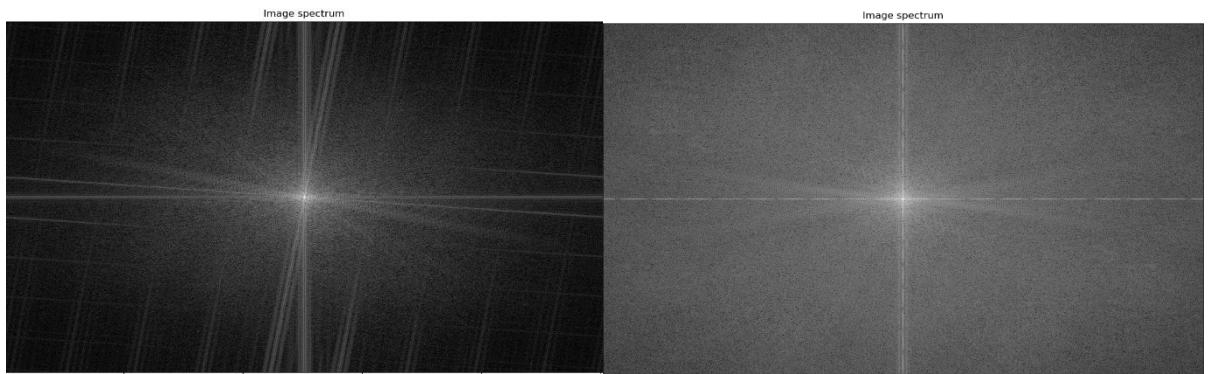
شکل ۱-۳



شکل ۲-۳



شکل ۳-۳

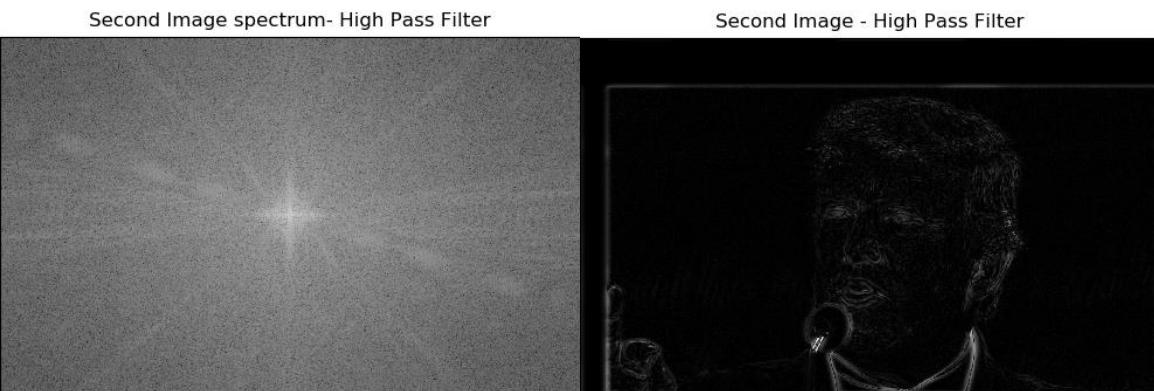


شکل ۳-۴

(b) کد مربوط به این قسمت در فایل **b.py** قرار دارد. نتایج برای هر دو جفت تصویر در تصاویر ۵-۳ تا ۸-۳ نمایش داده شده است.

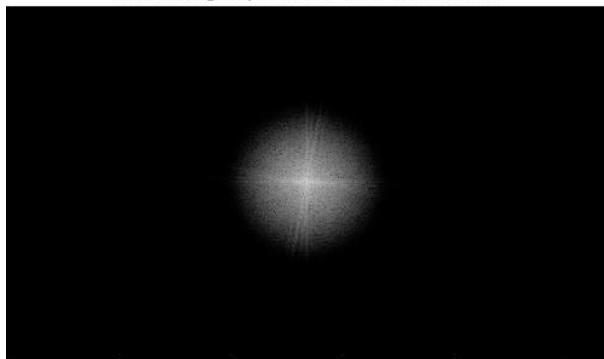


شکل ۳-۵



شکل ۳-۶

First Image spectrum- Low Pass Filter



First Image - Low Pass Filter

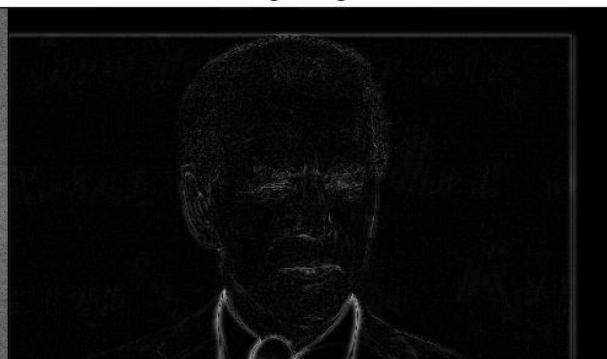


شکل ۷-۳

Second Image spectrum- High Pass Filter



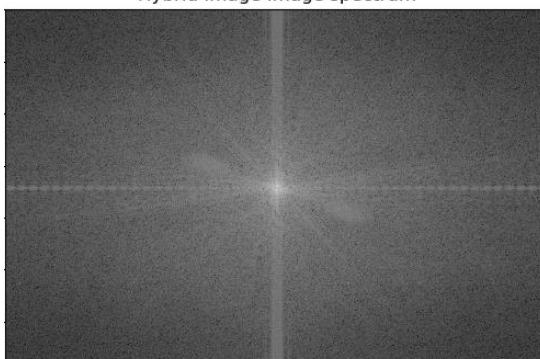
Second Image - High Pass Filter



شکل ۸-۳

(c) کد مربوط به این قسمت در فایل `C.py` قرار دارد. برای فرکانس برش مقدار 3×3 انتخاب شده است. نتایج به ترتیب در شکل ۹-۳ و ۱۰-۳ نمایش داده شده است.

Hybrid Image Image spectrum



Hybrid Image

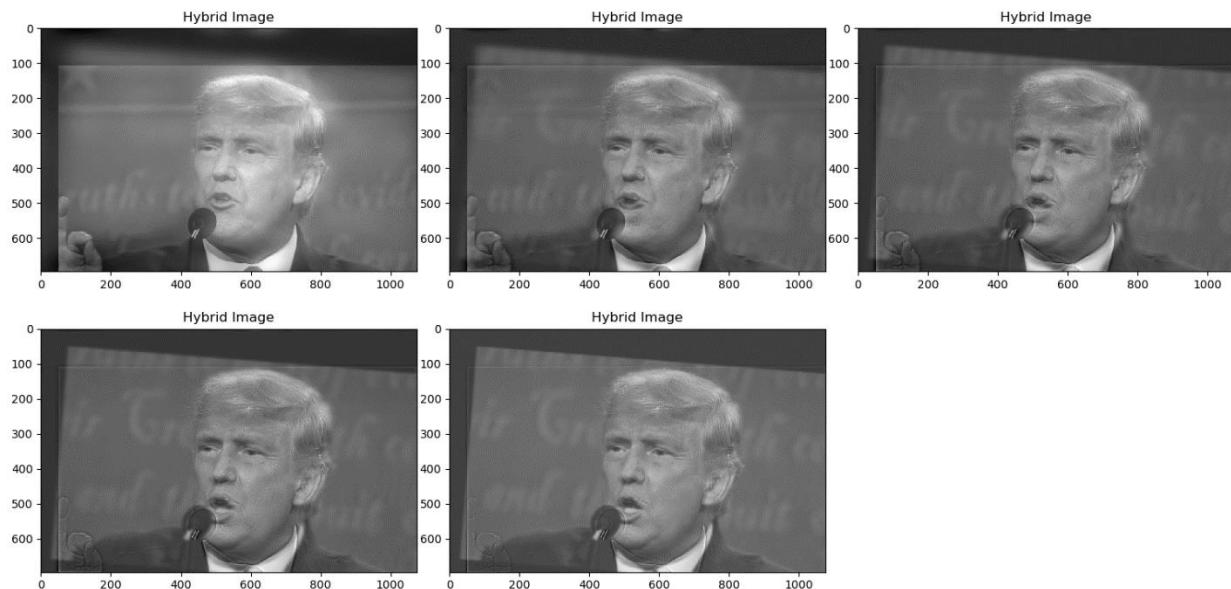


شکل ۹-۳

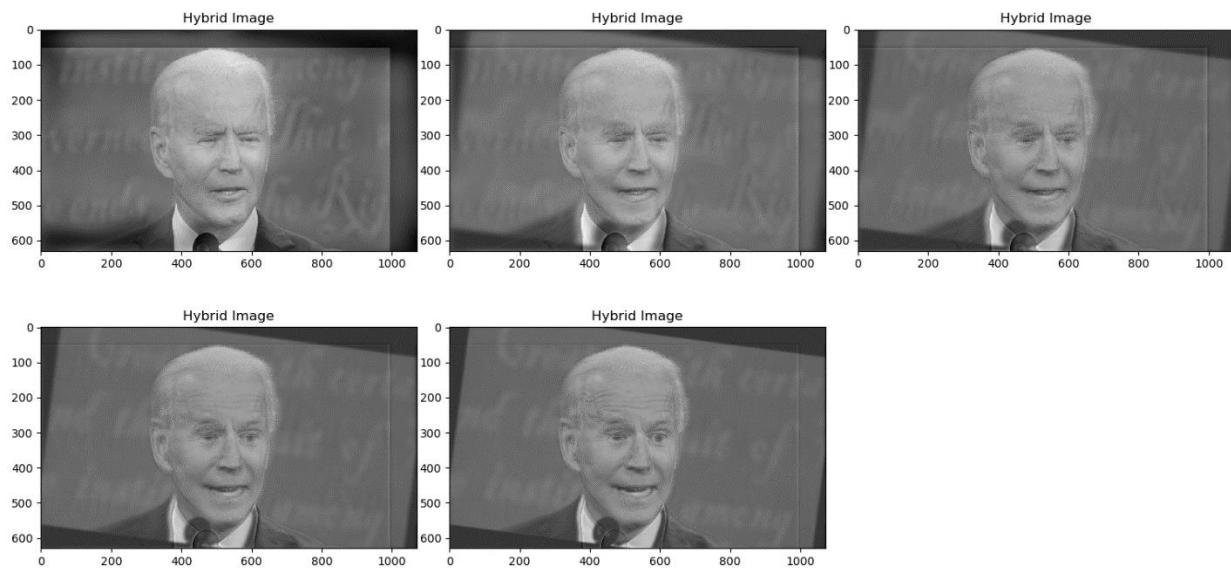


شکل ۱۰-۳

(d) کد مربوط به این بخش در فایل `d.py` قرار دارد. فرکانس‌های برش به ترتیب ۵، ۱۵، ۲۵، ۳۵ و ۵۵ انتخاب شده‌اند. نتایج در شکل‌های ۱۱-۳ و ۱۲-۳ نشان داده شده‌اند.



شکل ۱۱-۳

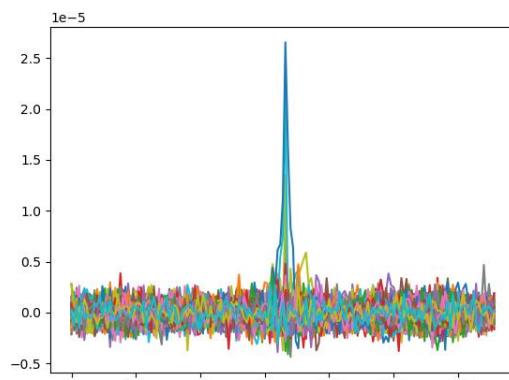


شکل ۱۲-۳

(۴)

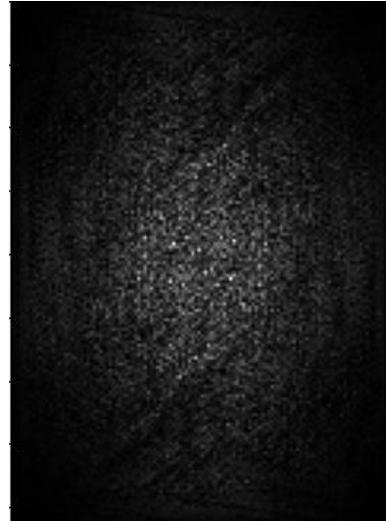
به علت کند بودن محاسبات مربوط به ساخت فیلترها، این عملیات در فایل `create_H.py` انجام می‌شود. این فایل به ازای ۱۰۰ شخص موجود در دیتابست، با توجه به شماره عکس‌ها ۱۰۰ فیلتر ساخته و با فرمت اکسل ذخیره می‌کند. در بخش‌های بعدی این فیلترها جهت انجام عملیات استفاده می‌شوند. لازم به ذکر است عکس‌های دارای پیشوند `M` از کلاس ۱ تا ۵۰ و عکس‌های دارای پیشوند `W` از کلاس ۵۱ تا ۱۰۰ ذخیره شده‌اند.

(a) کد مربوط به این قسمت در فایل `a.py` قرار دارد. فیلترهای مورد نیاز که قبلاً با فایل `create_H.py` ساخته شده‌اند از پوشه `a` وارد می‌شوند و پس برای تصویر ۱۴ از کلاس ۱، خروجی‌ها را در شکل ۱-۴ و ۲-۴ نمایش داده شده‌اند. علاوه بر این، برای تشخیص تعلق یک عکس به یک کلاس از معیار PSR^1 با حد آستانه ۵ استفاده می‌کنیم. با توجه به این معیار این عکس از کلاس ۱ تشخیص داده می‌شود (خروجی خط ۵۴ کد) که این نتیجه از روی شکل ۱-۴ مشهود است.



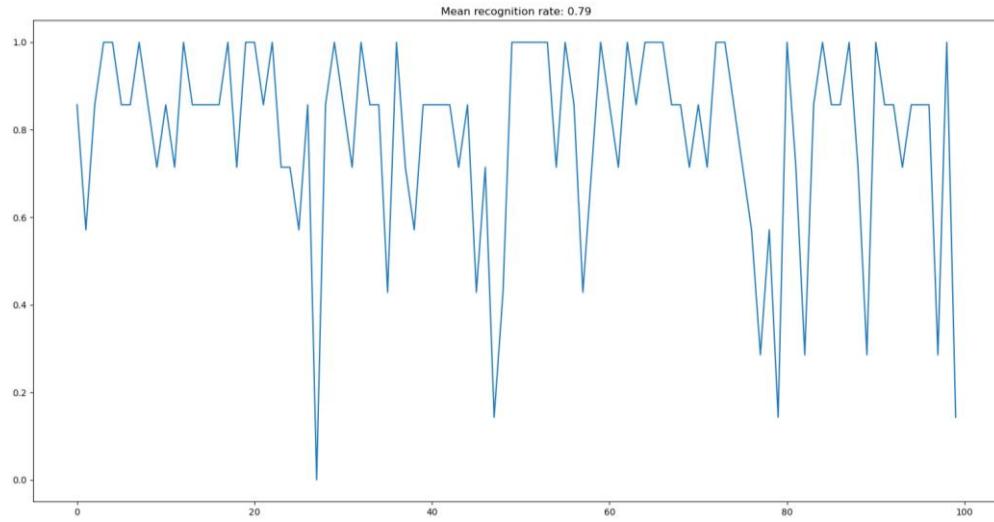
شکل ۱-۴

¹ M.Savvides, B. Kumar, "Face verification using correlation filters", Proc. Of the Third IEEE Automatic Identification Advanced Technologies (2002)



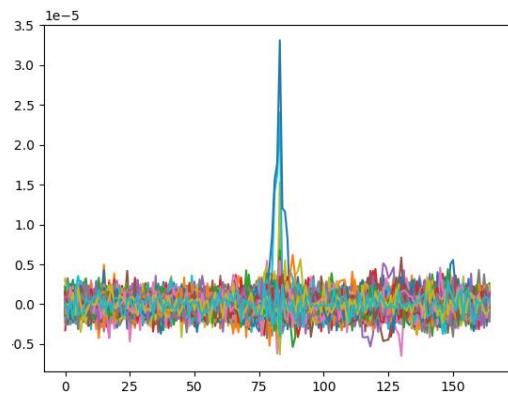
شکل ۲-۴

(b) کد مربوط به این بخش در فایل `b.py` قرار دارد. **Recognition rate** برای هر کلاس، برابر است با درصد تصاویر تشخیص داده شده از هر کلاس با استفاده از فیلتر مربوط به همان کلاس. با توجه به این تعریف و با در نظر گرفتن حد آستانه ۵ برای معیار **PSR**, **Recognition rate** برای هر کلاس و میانگین آن در شکل ۳-۴ نمایش داده شده است.

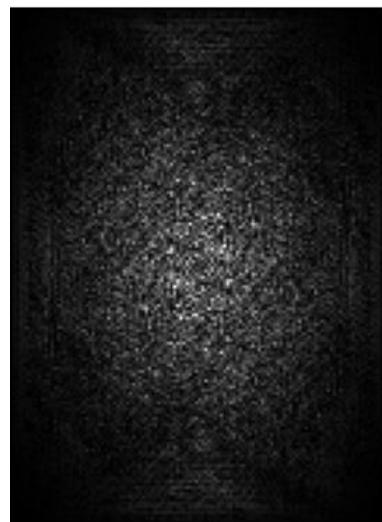


شکل ۳-۴

(c) کد مربوط به این قسمت در فایل `c.py` قرار دارد. مجدداً از فایل `create_H.py` برای ساخت فیلترها استفاده می‌کنیم. مشابه قسمت a از تصویر ۱۴ از کلاس ۱ به عنوان تصویر آزمون استفاده می‌کنیم. نتایج در شکل ۴-۴ و ۵-۴ نمایش داده شده است. همانطور که مشاهده می‌شود در مقایسه با قسمت a خروجی دارای **peak** بزرگتری است. این اتفاق احتمالاً به این دلیل رخ داده که در این قسمت، از تصاویر بیشتری برای ساخت فیلتر استفاده شده است.

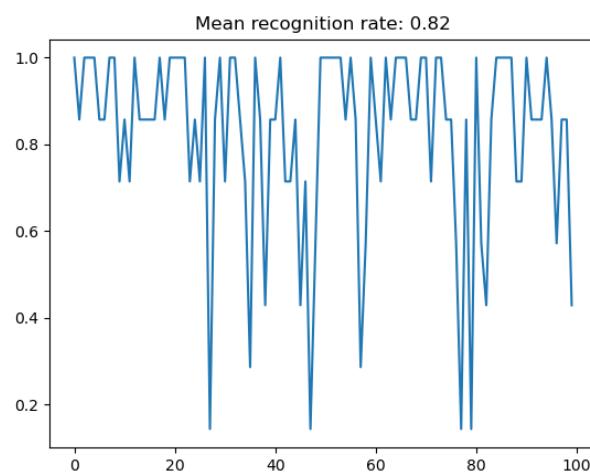


شکل ۴-۳



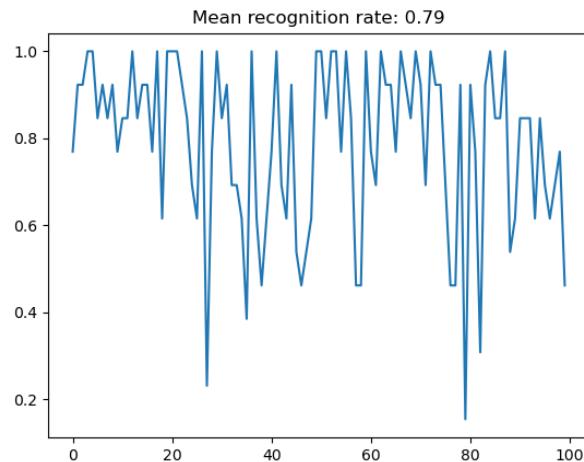
شکل ۴-۴

(d) کد مربوط به این قسمت در فایل `d.py` قرار دارد. ابتدا جهت مقایسه با قسمت b، تنها عکس‌های با پسوند `14` تا `20` را به عنوان تست در نظر می‌گیریم. نتیجه در شکل ۴-۶ نمایش داده شده است. همانطور که میبینید با افزایش داده‌های آموزش، مقدار Recognition rate ببهود یافته است.



شکل ۴-۶- داده‌های تست پسوند ۱۴ تا ۲۰

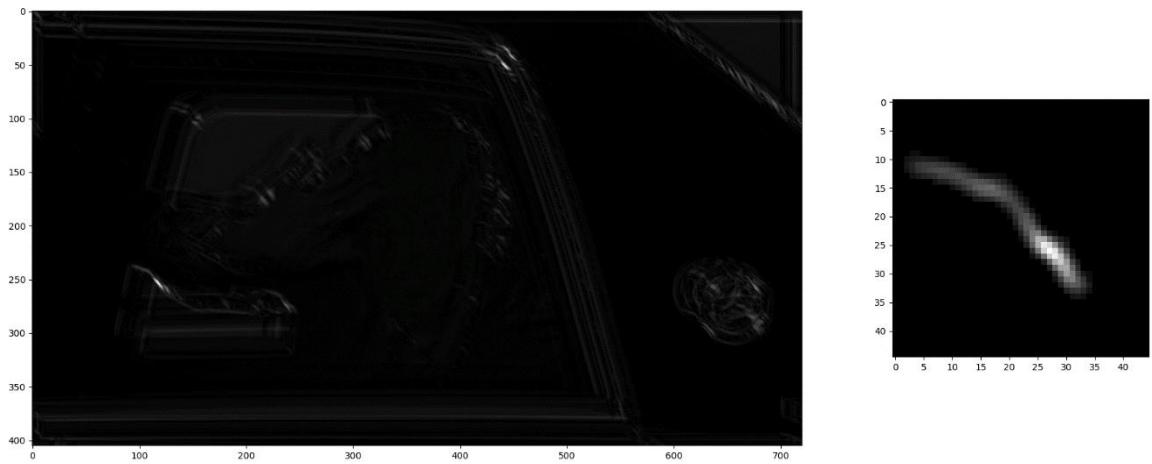
حال طبق خواسته سوال نتایج را برای داده‌های تست شامل پسوند ۲۶-۱۴ رسم می‌کنیم. شکل ۷-۴ نتیجه را برای این حالت نشان می‌دهد. با توجه به این شکل، مقدار Recognition rate نسبت به بخش b تغییر خاصی نداشته است. البته با توجه به چالشی بودن داده‌های تست با برچسب ۲۰ تا ۲۶، این عدم بهبود تا حد زیادی توجیه می‌شود.



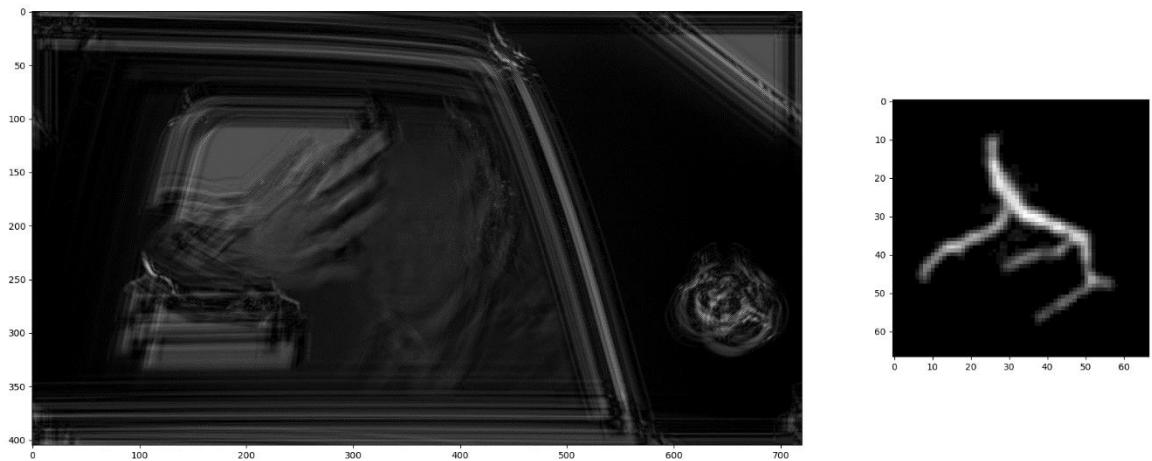
(۵)

(a) کد مربوط به این قسمت در فایل `a.py` قرار دارد. نتایج در شکل‌های ۱-۵ تا ۸ نشان داده شده‌است.

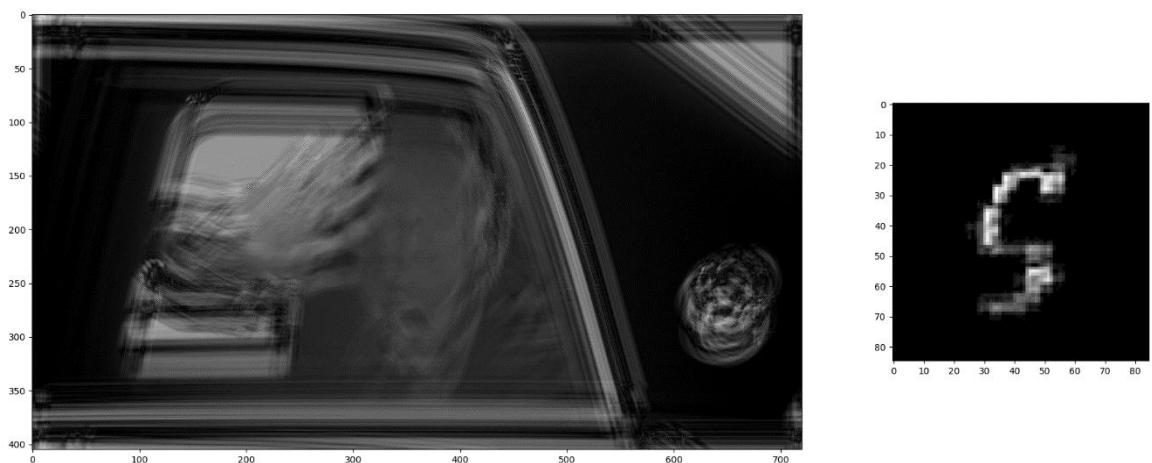




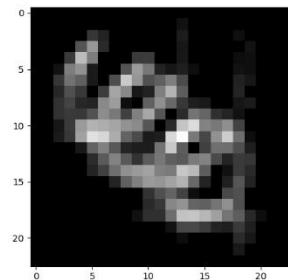
شكل ٢-٥



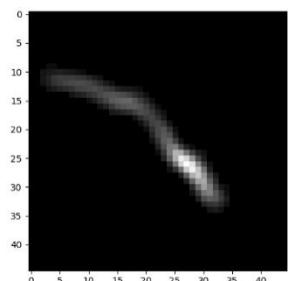
شكل ٣-٥



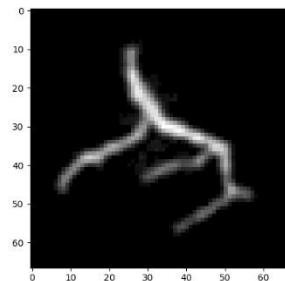
شكل ٤-٥



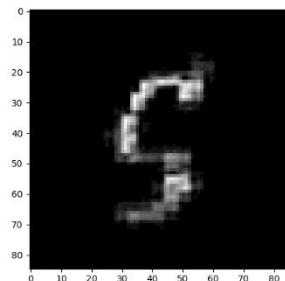
شكل ٥-٥



شكل ٥-٦

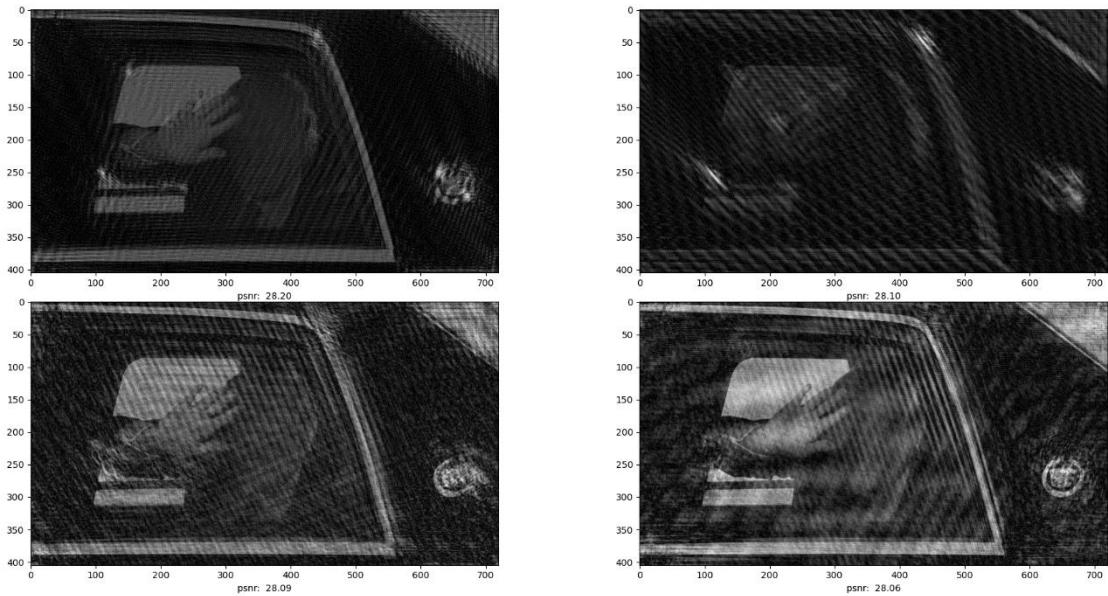


شکل ۷-۵

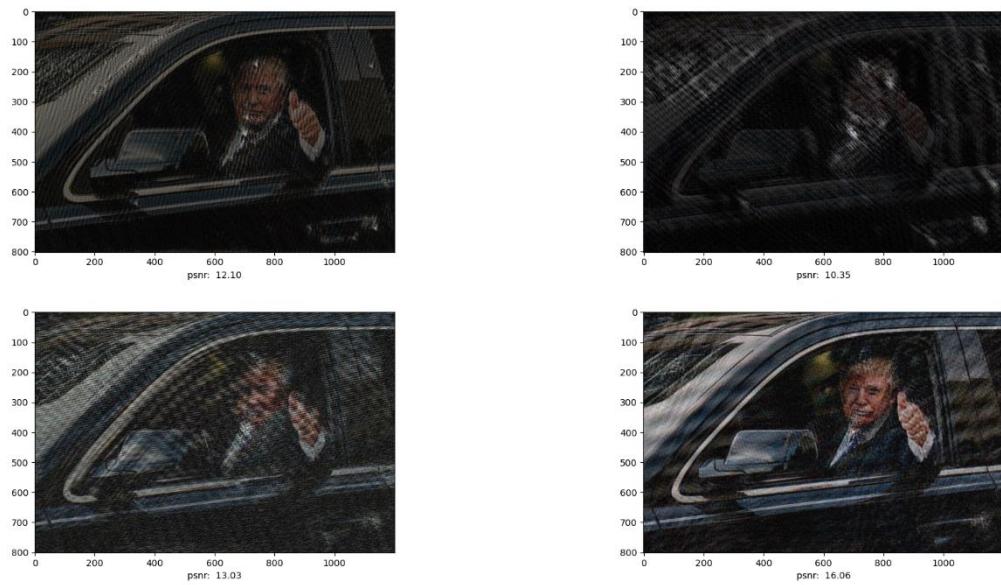


شکل ۸-۵

(b) کد مربوط به این بخش در فایل `b.py` قرار دارد. نتایج برای هر دو تصویر و به ترتیب کرنل‌هایی که در قسمت a نشان داده شد در شکل‌های ۹-۵ و ۱۰ ارائه شده است. معیارهای PSNR برای هر تصویر، زیر آن نوشته شده است.

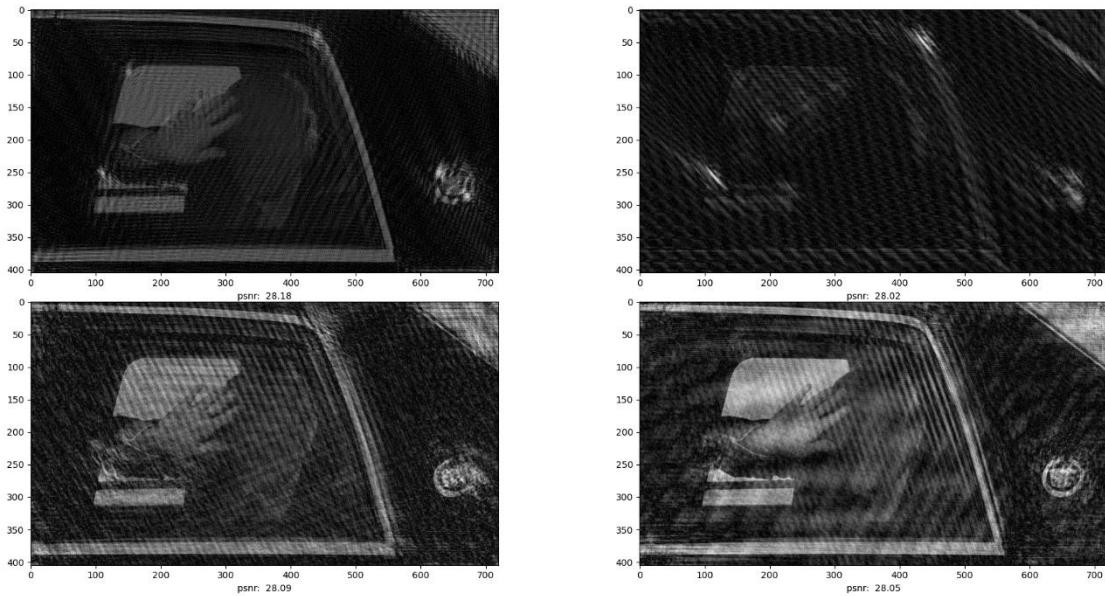


شکل ۹-۵

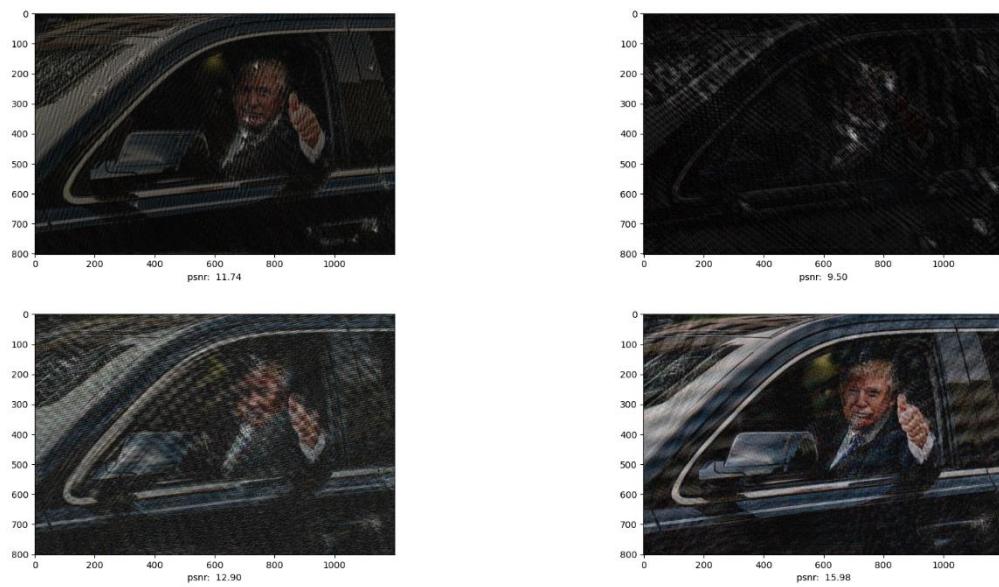


شکل ۱۰-۵

C) کد مربوط به این بخش در فایل C.py قرار دارد. مقدار لاندا ۰۰۰۱ است. با توجه به نتایج که در شکل های ۱۱-۵ و ۱۲-۵ نشان داده شده است، افزودن A باعث بهبود خاصی نشده و حتی کیفیت تصاویر را اندکی کاهش داده است.

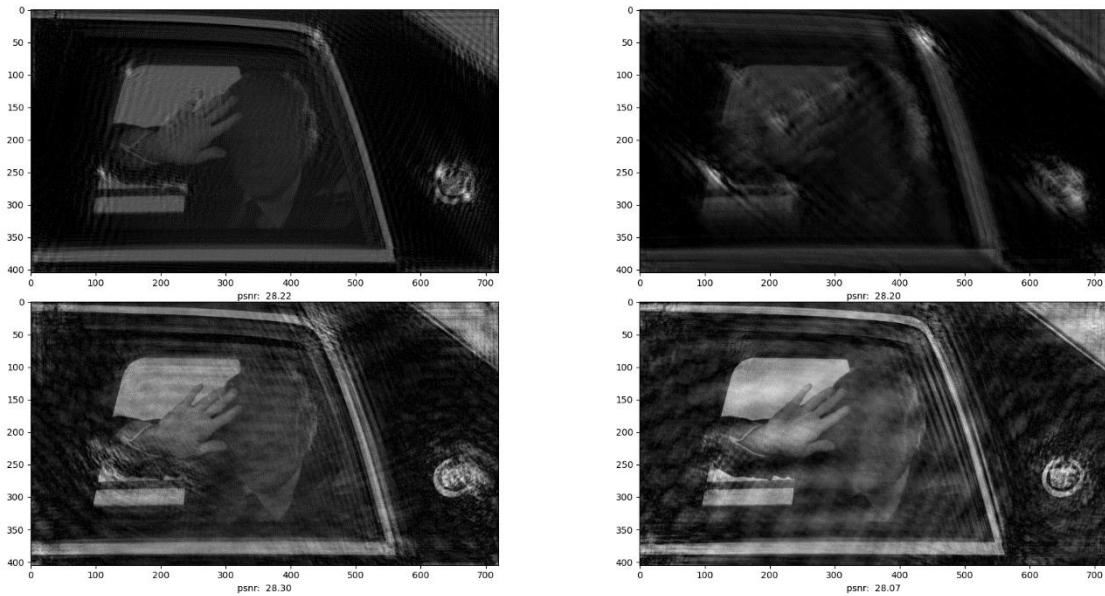


شکل ۱۱-۵

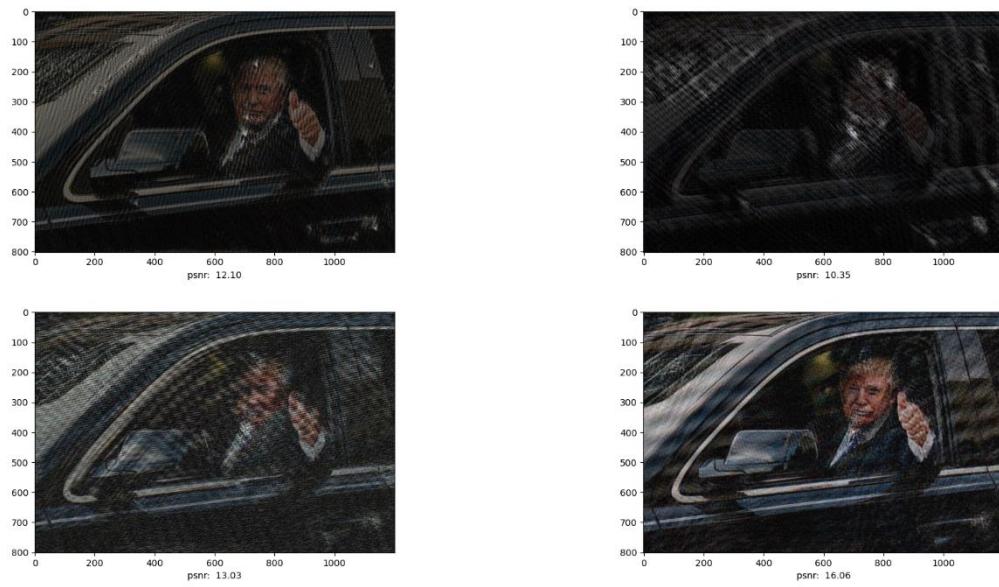


شکل ۱۲-۵

(d) کد مربوط به این بخش در فایل `d.py` قرار دارد. مقدار لاندا برابر با 5×10^{-5} انتخاب شده است. با توجه به نتایج که در شکل های ۱۳-۵ و ۱۴-۵ ارائه شده است، خروجی برای تصویر اول هم از نظر بصری و هم از نظر معیار PSNR بهبود یافته است اما تصویر دوم (رنگی) بهبود خاصی نداشته است.



شکل ۱۳-۵



شکل ۱۴-۵

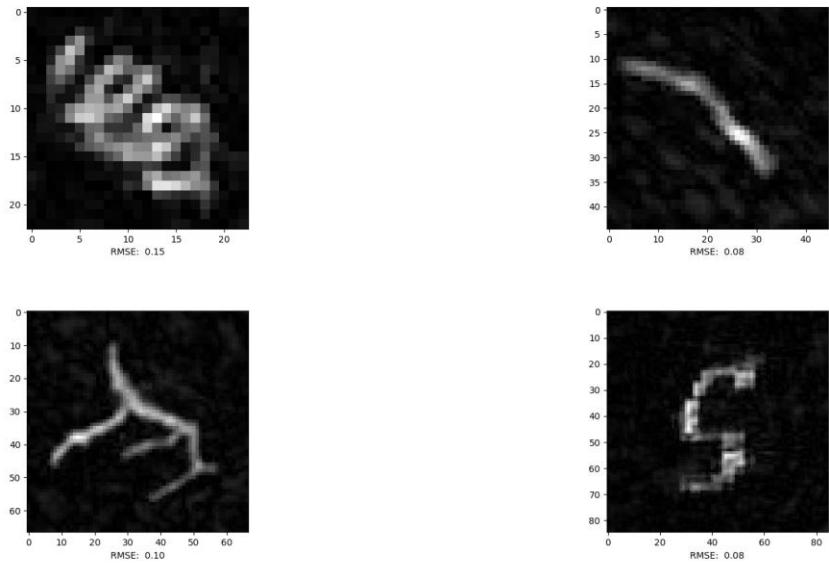
(e) کد مربوط به این قسمت در فایل e.py قرار دارد. از آنجا که رابطه ۲ در این سوال، ضرب نقطه‌ای است، می‌توان با جابه‌جا کردن H و X از همان روابط ۵ و ۶ استفاده کرد. به بیان دقیق‌تر:

$$H' = \operatorname{argmin}_H \|y - Hx\|_2^2 + \lambda \|AH\|_2^2$$

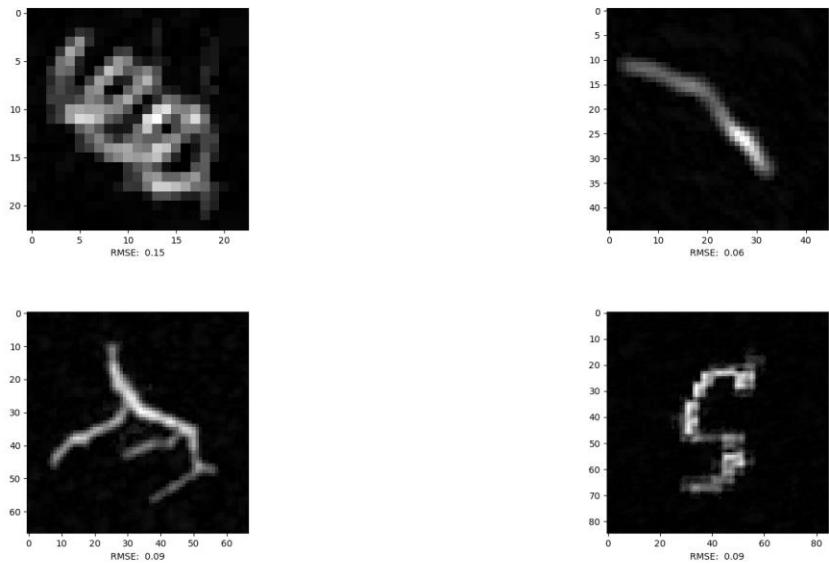
و فرم بسته‌آن:

$$H' = (x^t x + \lambda A^t A)^{-1} x^t y$$

سیس برای اعمال تبدیل فوریه معکوس روی H ، از تابع $\text{otf2psf}()$ استفاده می‌کنیم. با توجه به این توضیحات نتایج در شکل‌های ۱۵-۵ و ۱۶-۵ به ترتیب برای تصویر اول و دوم نمایش داده شده‌است. جهت محاسبه معیار RMSE، کرنل‌ها بین صفر و یک نرمال شده‌اند. با توجه به خروجی‌ها، معیار RMSE مقدار مطلوبی دارد.



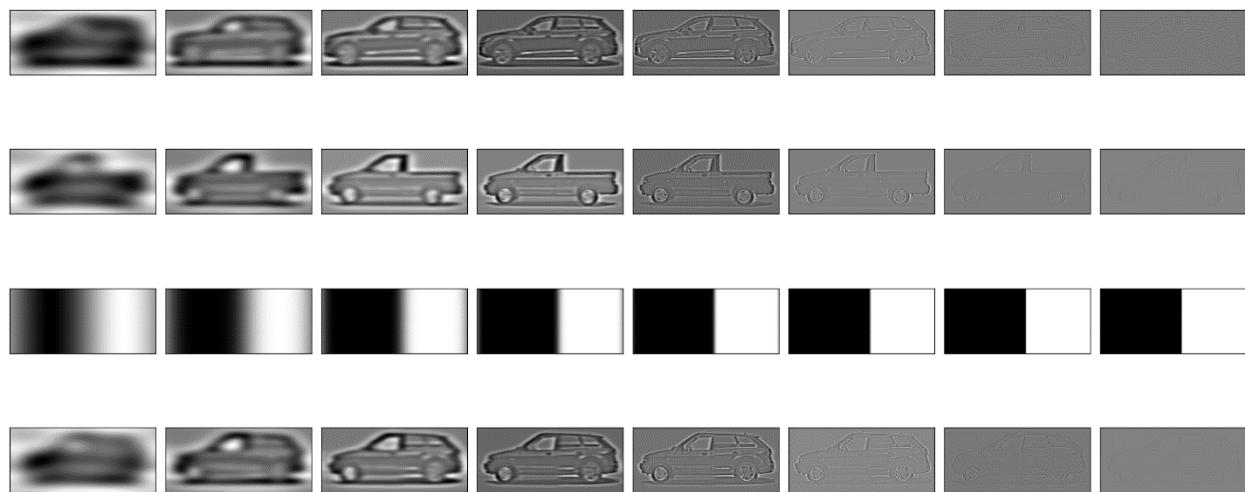
شکل ۱۵-۵



شکل ۱۶-۵

(۶)

(a) کد مربوط به این قسمت در فایل `a.pv` قرار دارد. در شکل ۶-۲ نتایج سطوح مختلف و در شکل ۶-۳ نتیجه نهایی برای $n=8$ ارائه شده‌است.

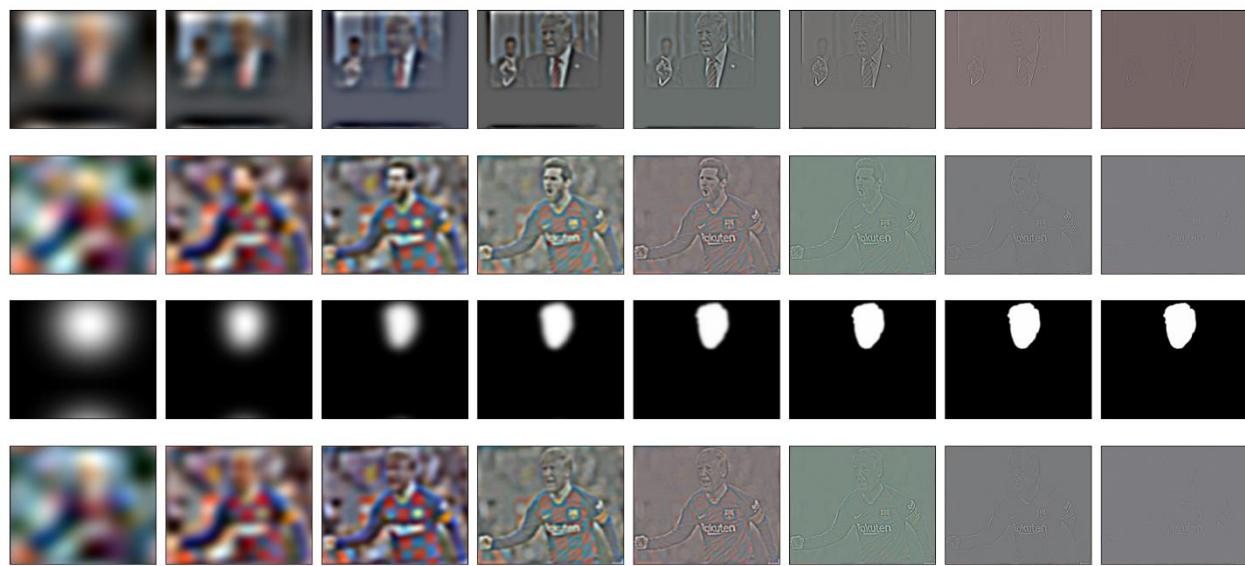


شکل ۱-

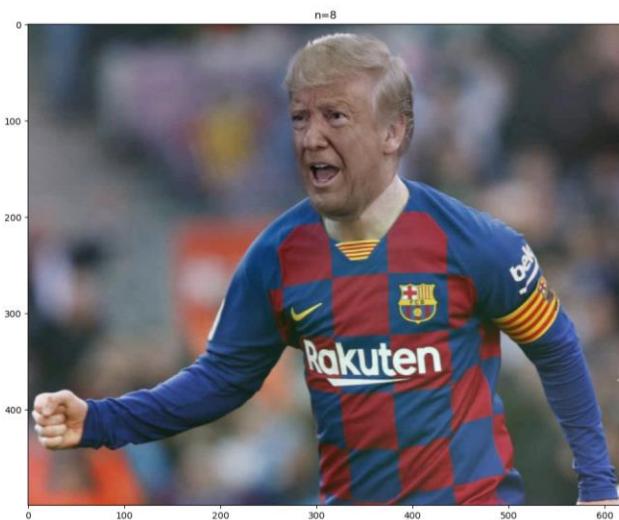


شکل ۲-

(b) کد مربوط به این قسمت در فایل **b.py** قرار دارد. در شکل ۳-۶ نتایج سطوح مختلف و در شکل ۴-۶ نتیجه نهایی برای $n=8$ ارائه شده است.



شکل ۳-



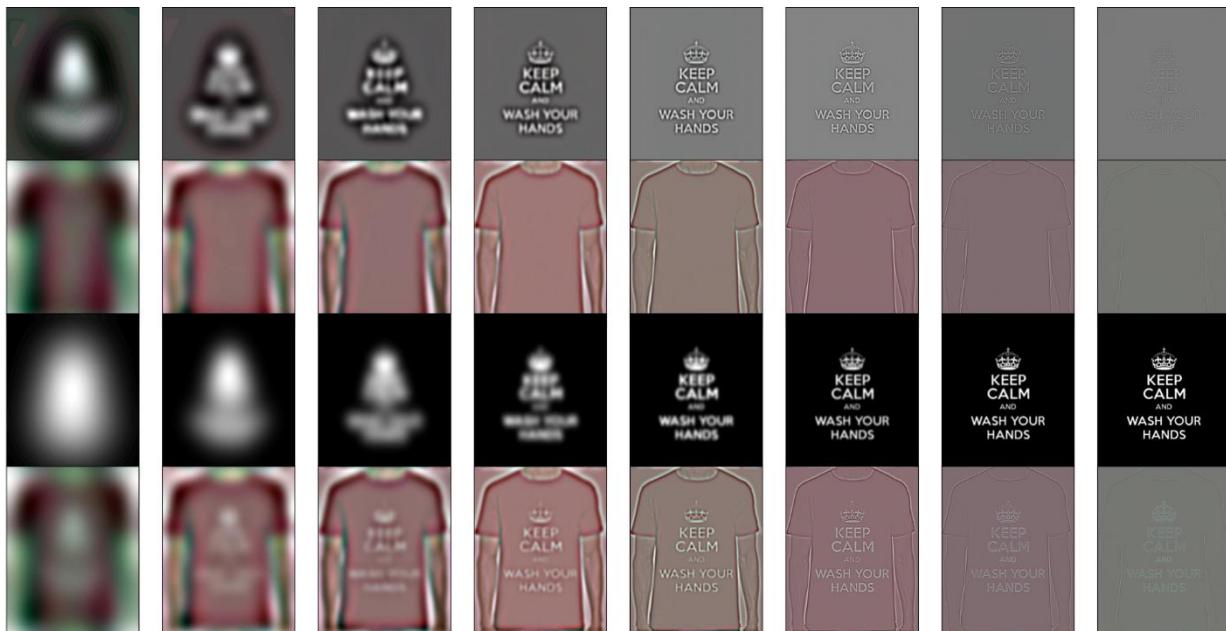
شکل ۴-

(c) تصاویرمورد نظر با نامهای `text` و `mask` در مسیر `III/P6/6 ذخیره شده‌اند. همچنین این تصاویر در شکل ۵-۶ نمایش داده شده‌اند.`



شکل ۵-۶

(d) کد مربوط به این قسمت در فایل `b.py` قرار دارد. در شکل ۶-۶ نتایج سطوح مختلف و در شکل ۷-۶ نتیجه نهایی برای $n=8$ ارائه شده است.



شکل ۶-۶



شکل ۷-۶

(۷)

۷) حتی در صورتی که تابع $f(x,y)$ یک تابع باند محدود و نرخ نمونه برداری آن مناسب باشد، در عمل همواره نیاز داریم که محدوده مشخصی از آن را برداشت کنیم (به عنوان مثال وقتی یک تصویر با ابعاد ۱۰۰۰ در ۱۰۰۰ میگیریم). این کار با ضرب تابع $f(x,y)$ در یک تابع باکس $h(x,y)$ انجام می‌شود. ضرب $f(x,y)h(x,y)$ معادل کانولوشن $F(x,y)$ و $H(x,y)$ در دامین فرکانس است. از طرفی می‌دانیم تبدیل فوریه تابع باکس $h(x,y)$ یک تابع با دامنه نامحدود است. بنابراین خروجی کانولوشن $F(x,y)$ و $H(x,y)$ دامنه نامحدودی دارد. در نتیجه هیچ تابعی با دامنه محدود نمی‌تواند باند محدود باشد.