به نام خدا

دانشگاه تهران
دانشکده مهندسی برق و کامپیوتر

پردازش سیگنال‌های زمان-گسسته

تمرین سری 4

نام و نام خانوادگی	حمیدرضا علی اکبری خویی
شماره دانشجویی	8101966514
تاریخ ارسال گزارش	11/05/99

فهرست گزارش سوالات

Spatial Domain

Kernel

یک فیلتر دوبعدی عموما متقارن میباشد که در این پروژه 3در3 تعریف شده است که به صورت کانولوشنی بر روی دیتا های عکس یا همان پیکسل ها اعمال میشود و به این صورت میباشد که بعد ایز ضرب در عمل کانولوشن جمع همه 9 درایه آن گرفته میشود و در یک پیکس وسطی ذخیره میشود و بعد کرنل به جلو حرکت میکند.

Most useful kernels-

Sharpen.

در ابین کرنل تاکیرد بر این است که لبه های تیز عکس را به صورت خوبی نامایانگر شود یعنی تاکید بیشتر بر لبه های تیز عکس داشته باشد، از این کرنل استفاده میشود و تاثیر به سزایی بر برجسته کردن نواحی روشن و تاریک دارد !

/\

Blur

در این کرنل سعی بر این است که با عبور یک فیلتر 3 در 3 میانگین گیر یا movingaverage باعث میشود که دیتاهای کنار هم به صورت خوبی با هم ادغام شوند که با این کار شفافیت و وضوح تصویر کمتر میشود.

/\

Outline

برای تشخیص نقاط ناهمپیوستگی عکس ها یا به اصطلاح گوشه(Edge() کاربرد دارد که با استفاده از این کرنل میتوان گوشه های یک تصویر را استخراج کرد.

/\

Gauss

همانند Blur تصویر را وضوحش راکم میکند، اما در این جا با توجه به تاکیر به وابستگی به تابع گوسی میزان ضرایب دارای اندازه بزرگتری حول مرکز خواهند داشت.

/\

Average Moving

همانند کرنل قبل باعث کمتر شدن وضوح میشود ولی اینبار با استفاده از میانگین پیکسل های مجاور اینکار را میکند و البته این کرنل باعث میشود که لبه های تیز حذف شوند.

/\

H-Line

در این کرنل باعث شفاف سازی و برجسته سازی لبه های عمودی موجود در تصویر میشود.

/\

V-Line

دراین کرنل باعث سفاف سازی و برجسته سازی لبه های افقی موجود در تصویر میشود .

/\

Identify

این کرنل مانند کرنل پاسخ ضربه همان تصویر اصلی را برمیگرداند و کاری انجام نمیدهد!

:/

/\

House

برای این که این فیلتر ها را در این تصویر خانه اعمال کنیم هر کدام را با تاثیر آن میبینیم:

برای اعمال کردن هر کدام از کرنل ها از imfilter استفاده کرده ام که تفاوتش با مدل کانولوشنی این است که برای imfilter خدش خروجی را ABS و UINT8 میکند و نمایش میدهد ولی در مدل کانولوشنی باید این کار را انجام میدادیم تا بتوانیم از دستور Imshow استفاده کنیم:

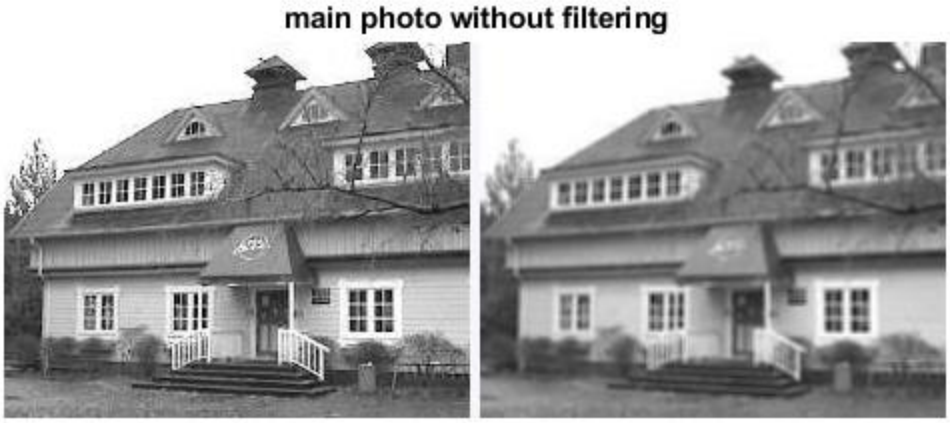


Figure 1

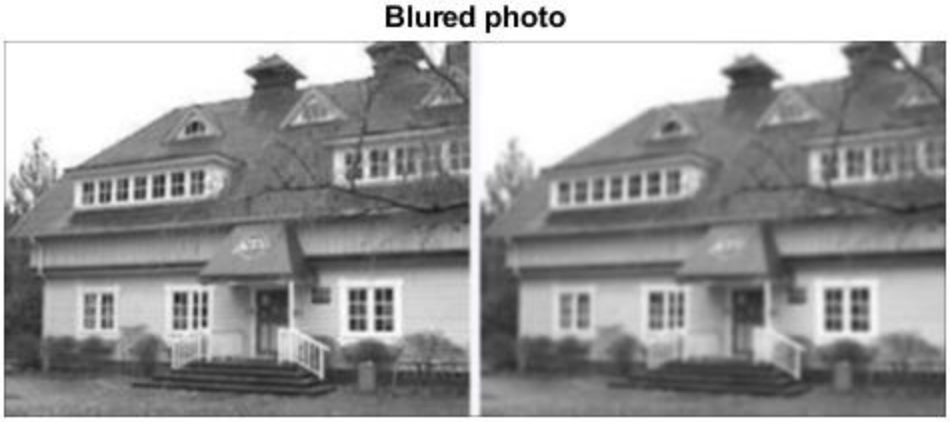


Figure 2

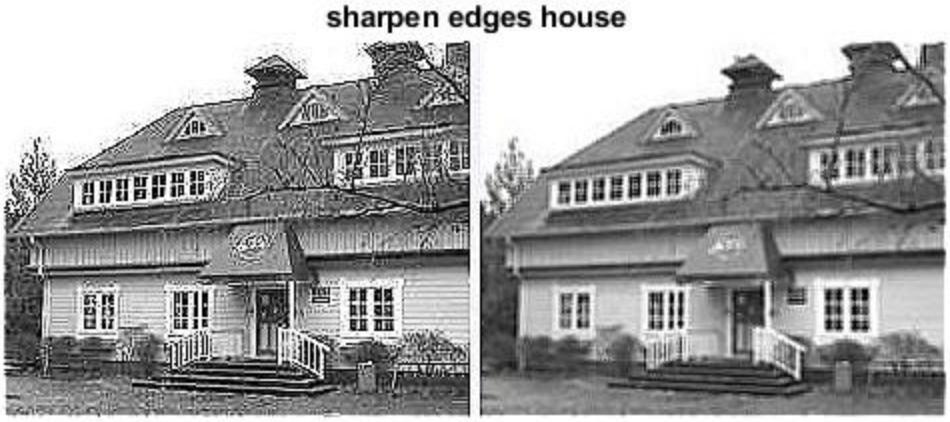
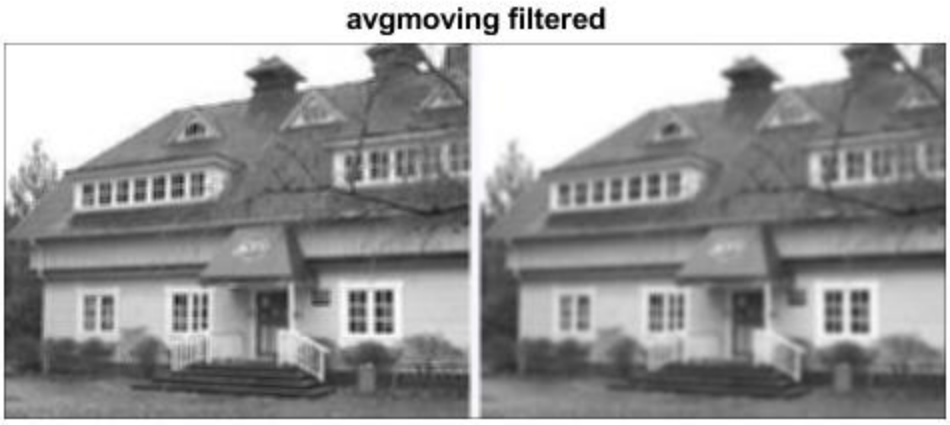


Figure 3

Figure 4

Figure 5



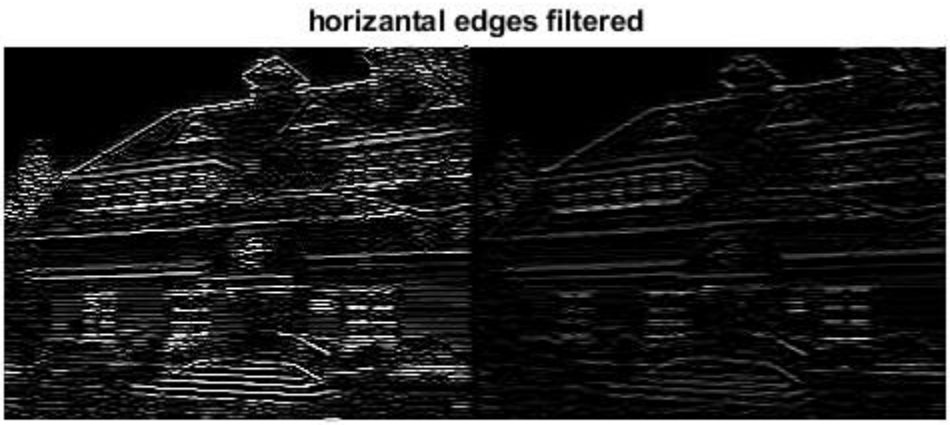


Figure 6

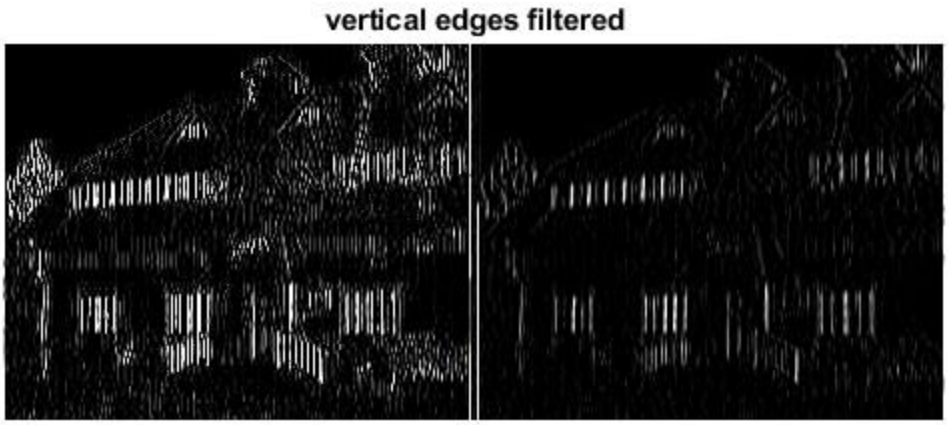


Figure 7

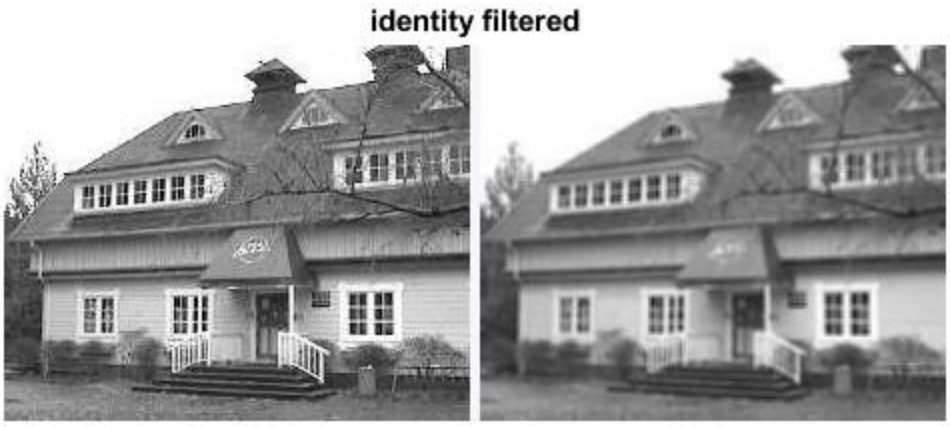


Figure 8

KOBE

برای این قسمت باید عکس اصلی را که از حافظه خوانده شده را با دستور IMresize عملگرDownsample را انجام دهیم و بعد با همان دستور قبلی و با متد Nearest تصویر را upsample کنیم ولی در این میان با توجه به اینکه تعدادی دیتا را از دیت داده ایم باید با ایتفاده از کرنل ها مقداری کیفیت تصویر بعد از اعمال دو عملگر را بهبود بخشیم که اینگونه عمل میکنیم:

- اول با استفاده از کرنلmovinavg تصویر را در نقاط تیزی بهبود میبخشیم تا نقاط تیزی به وجود آمده را را نرم تر کنیم که این با استفاده از این کرنل میتوان انجام گیرد.
- دوم با استفاده از کرنل Gauss انجام میگیرد که همانند کرنل قبلی میباشد ولی در این قسمت کرنل دارای ضرایب یکسانی نیست و البته باعث بهبود نقاط تیزی موجود در تصویر به خاطر از دست دادن دیتا شود.

البته میتواد دید که این عملگرد دو کرنل باعث میشود که این کرنل ها مانند فیلتر های پایین گذر عمل شوند که چون باعث بهبود لبه های تیز میشود ولی تفاوت این دوتا کرنل در این است که فرکانس قطع آنها باهم فرق میکند و برای کرنل movingavg دارای فرکانس قطع حول $\frac{\pi}{4}$میباشد ولی برای کرنل دیگر با توجه به ضرایب که نامتقارن میباشند نمیتوان دقیق گفت که فرکانس قطع فیلتر های پایین گذر دقیقا در چه محدوده ای میباشد.

Figure 9

Figure 10

enhanced with average moving kernel

enhhanced with gaussian kernel

Figure 11

برای اول باید عکسصفحه را باید بارگذاری کنیم در workspace و بعد از آن باید با توجه به اینکه مقادیر رنگی تصویر به درد نخواهند خورد باید تبدیل به سیاه سفید شود و بعد کرنل های ine_Hو line_V را بر آن اعمال کنیم (چرا؟) چون که این کرنل ها عملا باعث به وجو آمدن و آشکار سازی لبه های افقی و عمودی میشودند و با توجه به این که تصویر تقریبا در راستای عمودی افق قرار دارد با اعامل این کرنل ها میتوان به لبه های تصویر با دقت خوبی دست یافت.

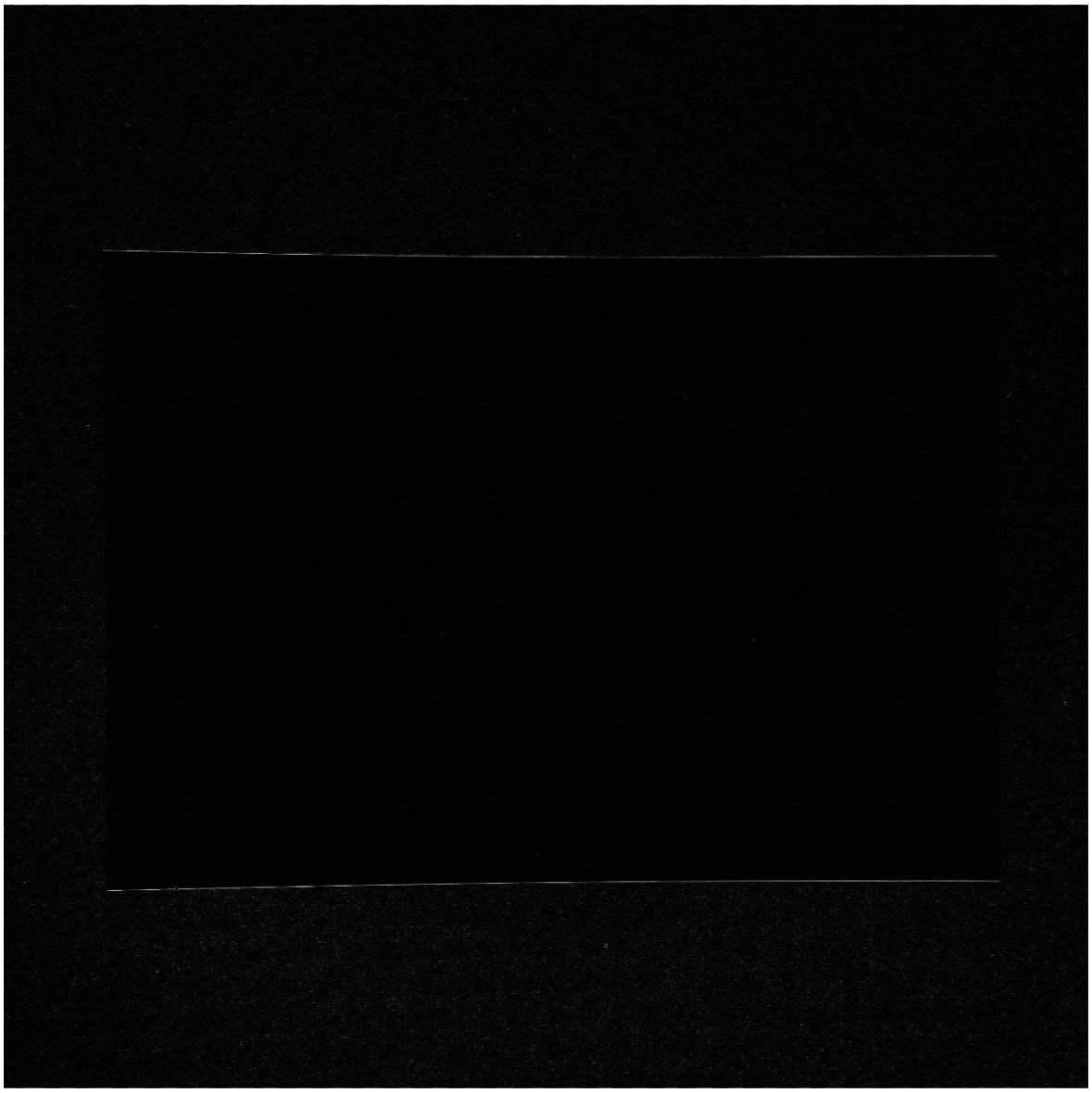


Figure 12

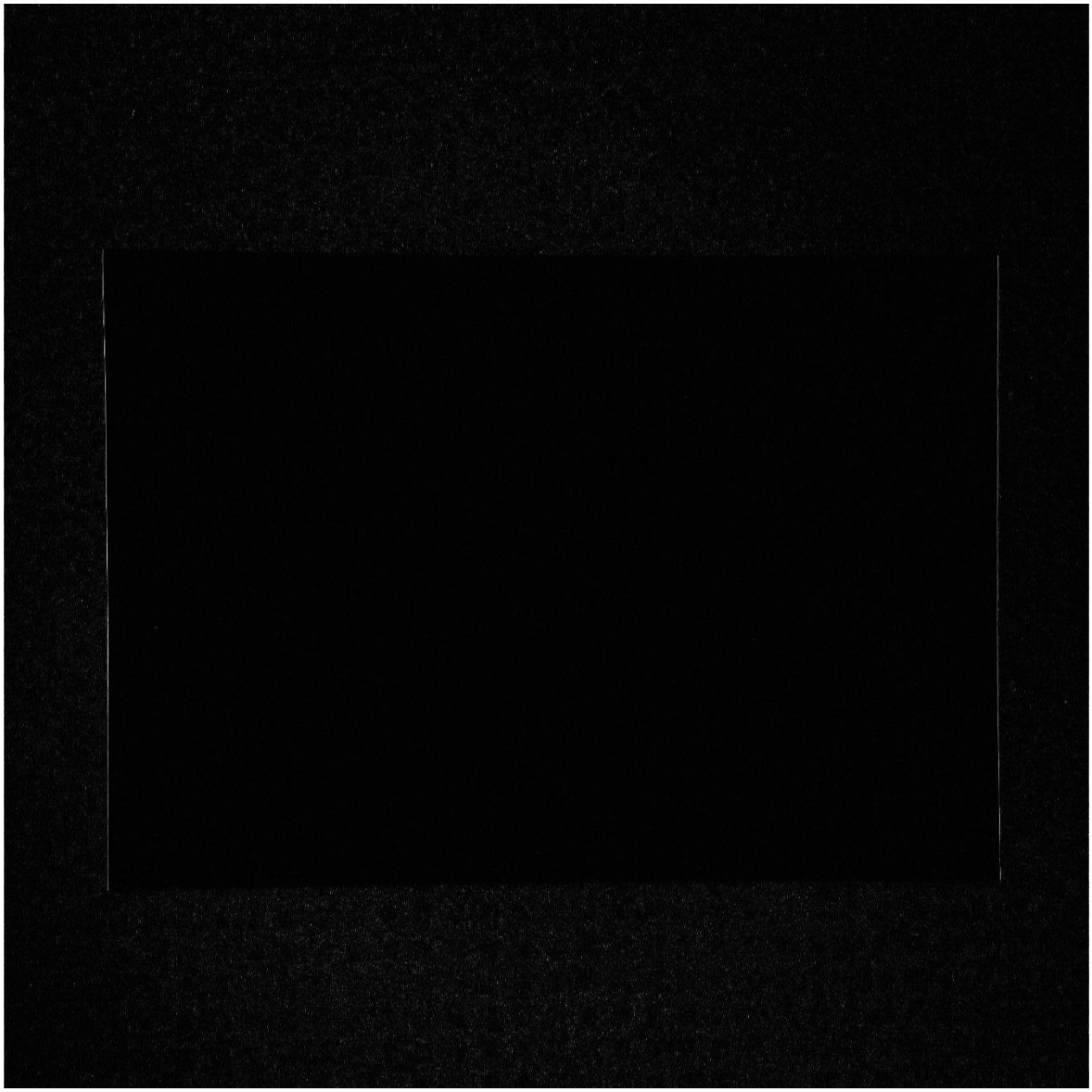


Figure 13

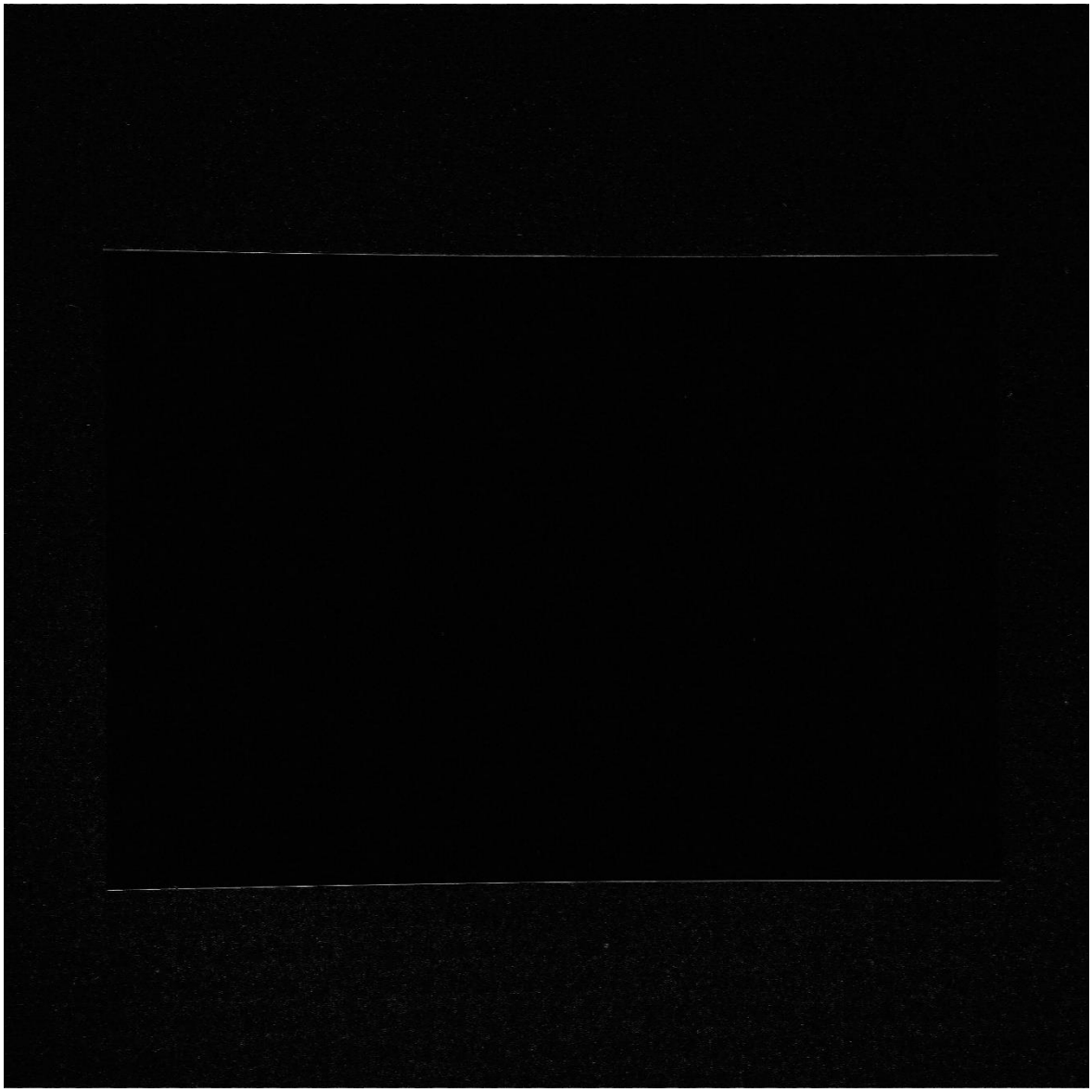


Figure 14

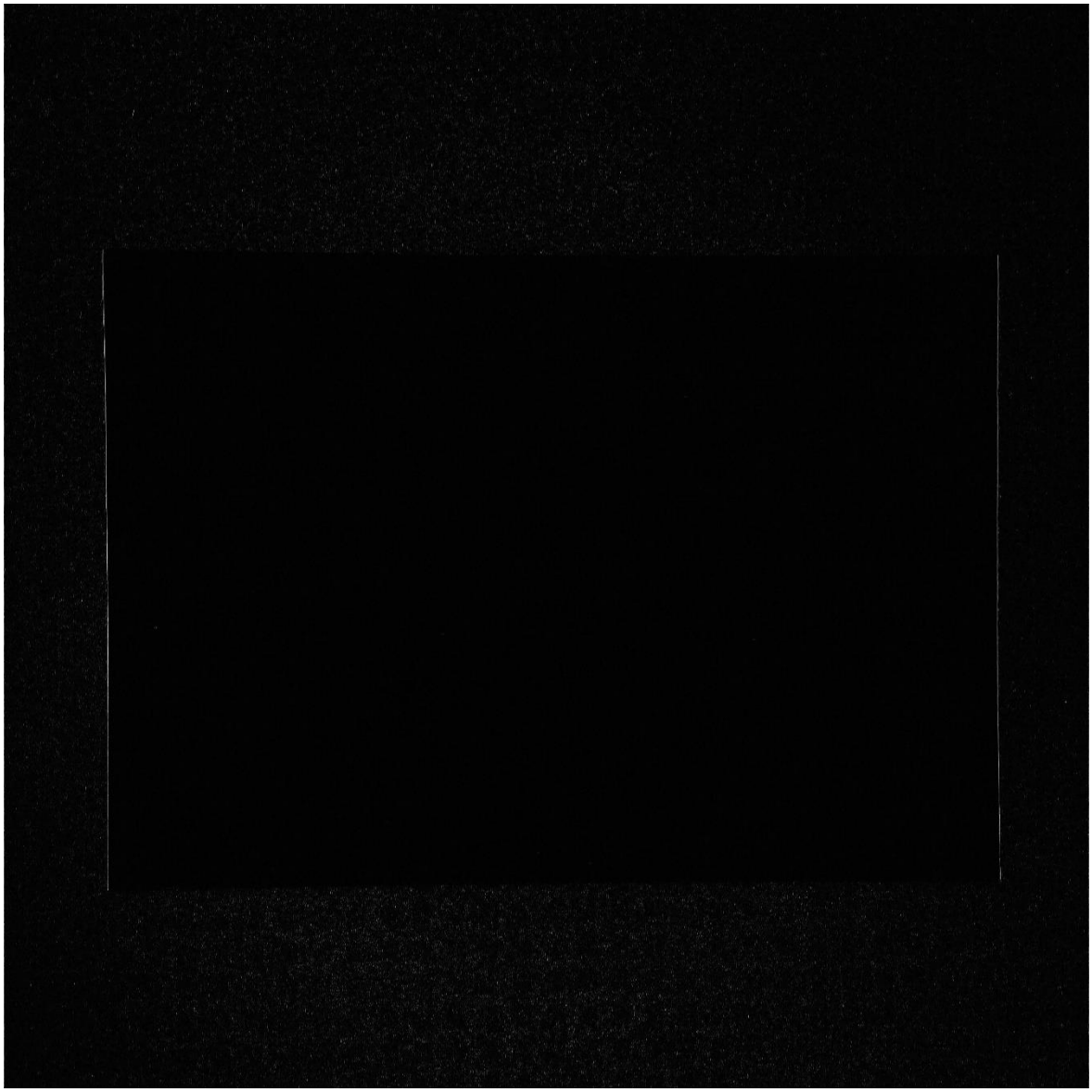
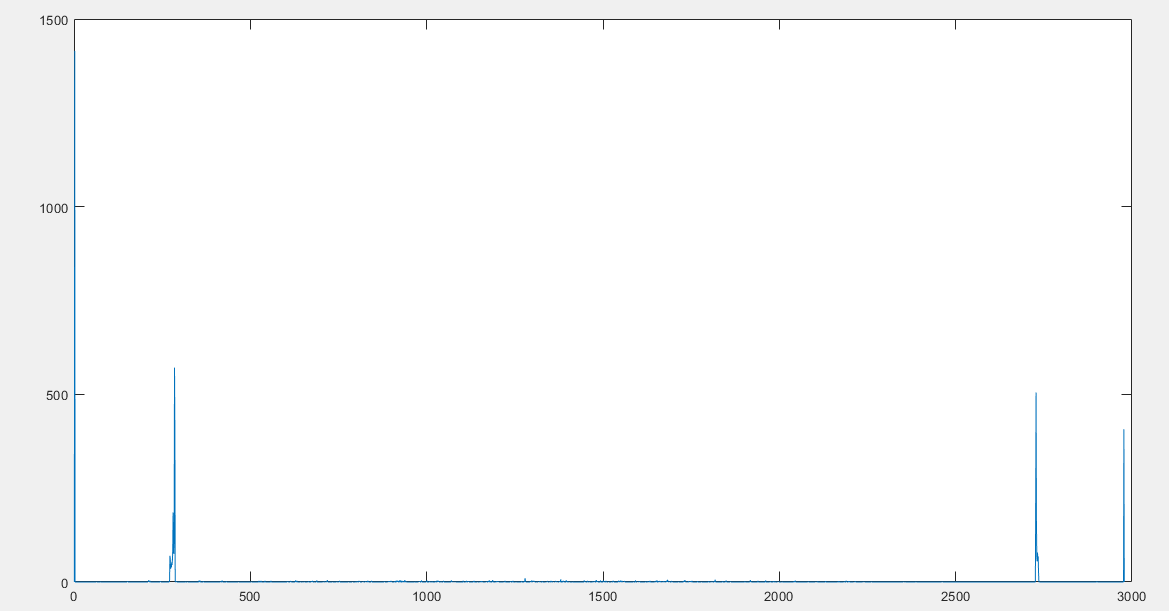


Figure 15

تصویر 12 و 14 خط های افقی و تصویر های 13 و 15 خط های عمودی عکس اولیه را نیان میدهند ولی با این تفاوت که دو تصویر 14 و 15 از مقدار نویز مربوط تصویر های اولیه کم شده است!

حال با توجه به mask کردن تصویر که با ترشولد 128 انجام گرفته است تصاویر خروجی لاجیکال خواهند بود که صفر یا یک خواهند بود یک ها سفید و صفر ها سیاه!

حال با توجه به تصاویر بالا میتواند فهمید که در لبه های اشکار سازی شده خط سفید دیده میشود که خوب این ایده به نظر میرسد که برای آشکار سازی لبه های افقی ماتریس شکل 14 را سطر هایش را جمع کنیم در نقاطی که ان سطر های آشکار سازی شده است سیگنال خروجی دارای اکسترممم های مشخص برای مختصات های مستطیل خواهد بود.



16Figure

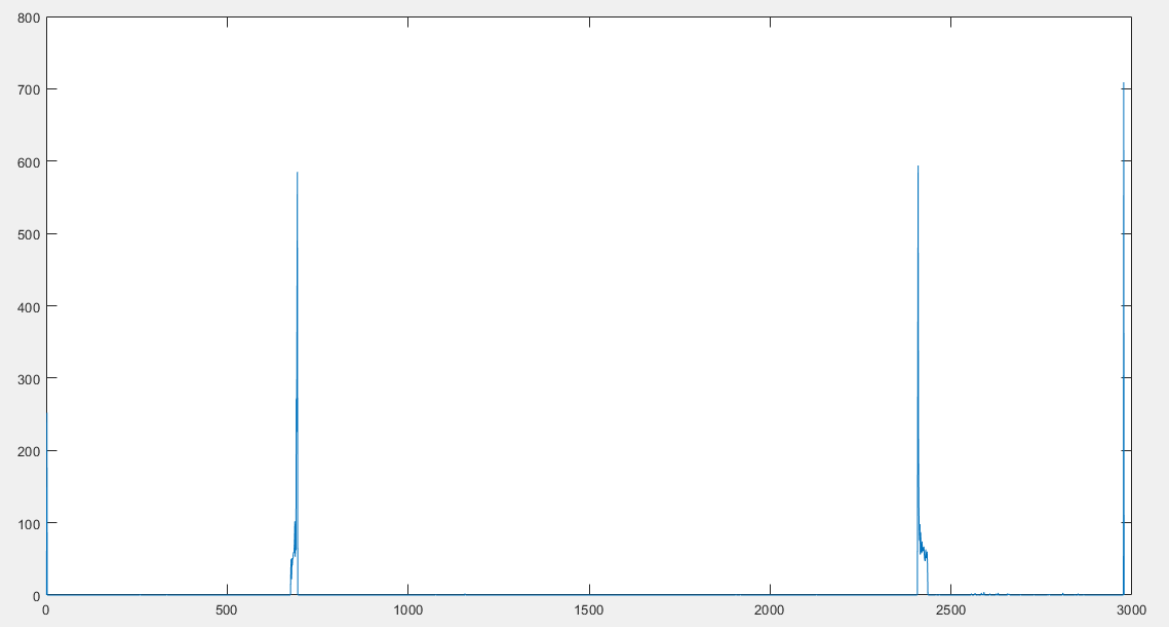


Figure 17

حال برای گرفتن دو تا از پیک های دو نمودار اول دوبار نمودار را smooth کرده ام که باعث میشود کنار یک های اصلی اکسترمم موضعی دیگری نداشته باشیم و بعد اول و آخر نمودار را حذف میکنیم چون که با توجه به لبه های کاغذ اصلی و این که خود متلب با توجه به کرنل ها ماتریس تصویر اصلی را یک ردیف صفر از هر طرف اضافه میکند آن پیک هارا گرفته ایم و حال چون انها عملا نیاز نداریم اول آخر سیگنال دریافتی یعنی شکل 17 و 16 را حذف میکنیم و بعد با استفاده از تابع findpeaks لوکیشن پیک هارا در میآوریم و حال با توجه به این که در وهله اول مقداری از سیگنال را حذف کرده ایم و با توجه به تابعsmooth لوکیشن های بدست آمده دارای خطا میباشد که برای جبران آن از همان شکل های 16 و 17 استفاده میکنیم (برای ارتعاشات 16 و 17 که پیک میدهند مقداری از سیگنال را از دامنه کم کرده ایم تا صفر مطلق شود) و حال با

توه به پیک های 16 و 17 مختصات جدید را بدست می آوریم و مستطیل را میکشیم که در پایین نتیجه الگوریتم را برای 3 نوع تصویر را میبینید:

Figure 18

Figure 19

Figure 20

Figure 21

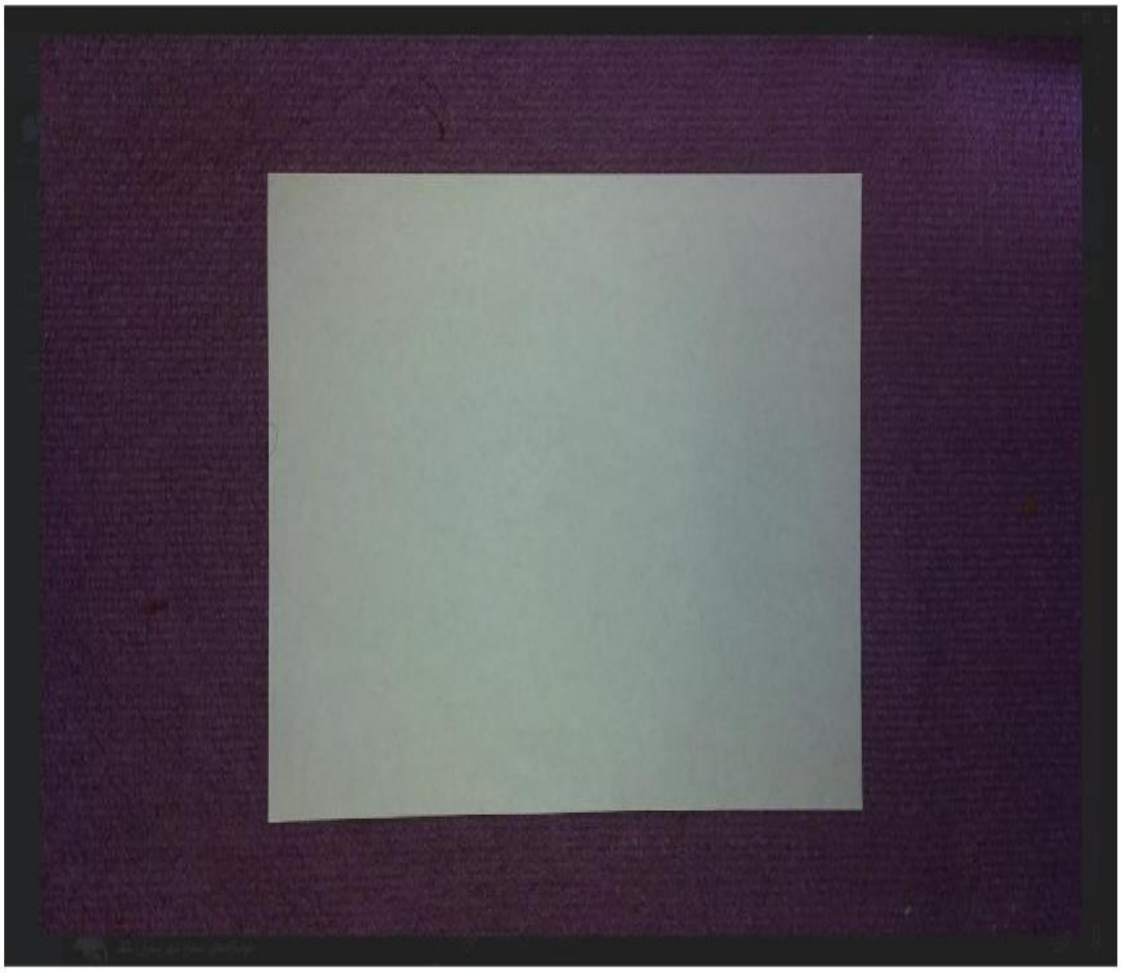


Figure 22

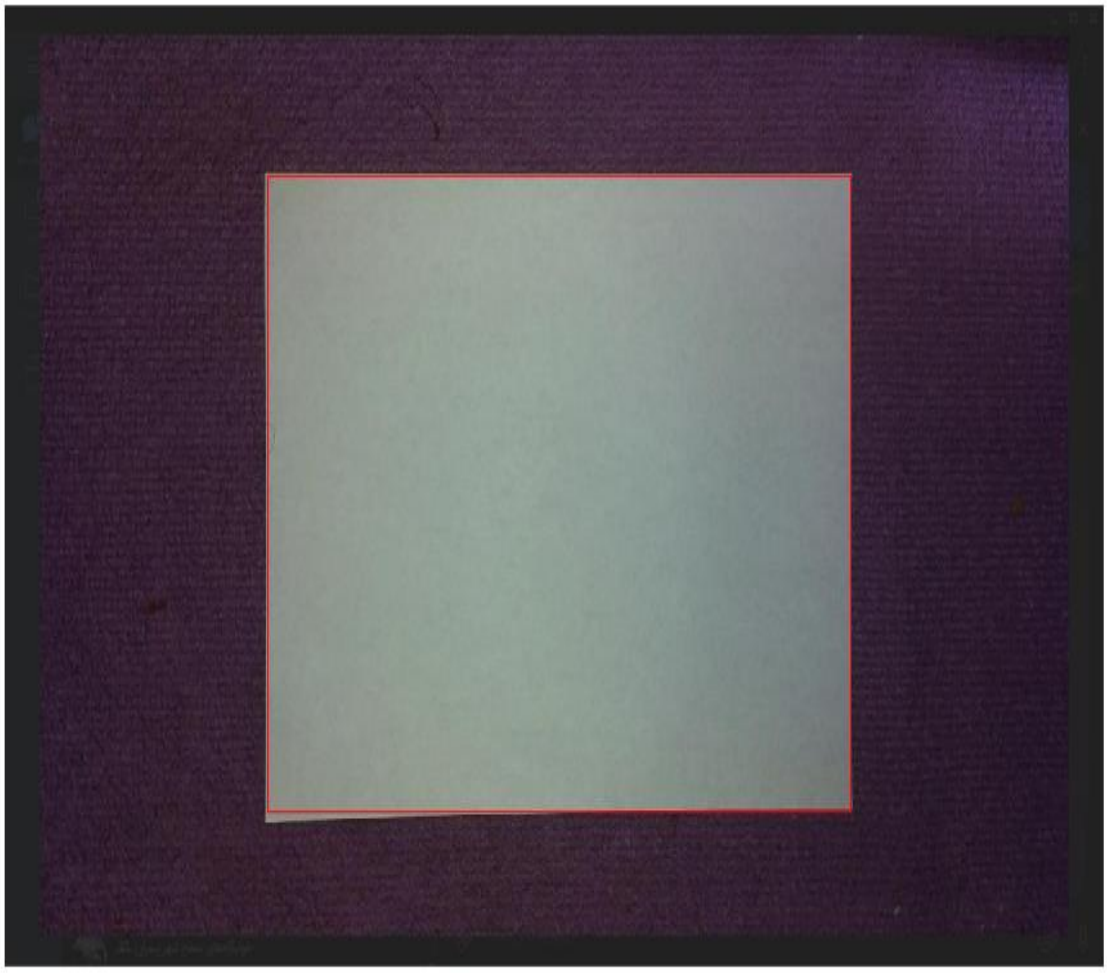


Figure 23

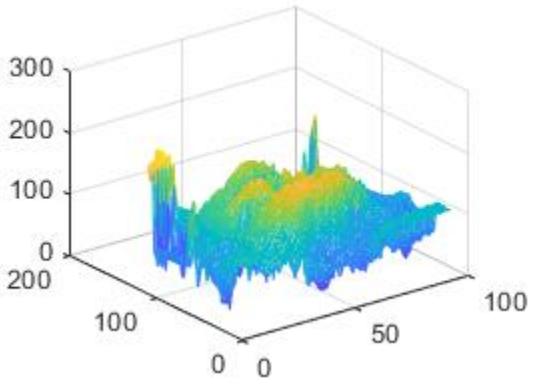
Frequency domain filtering

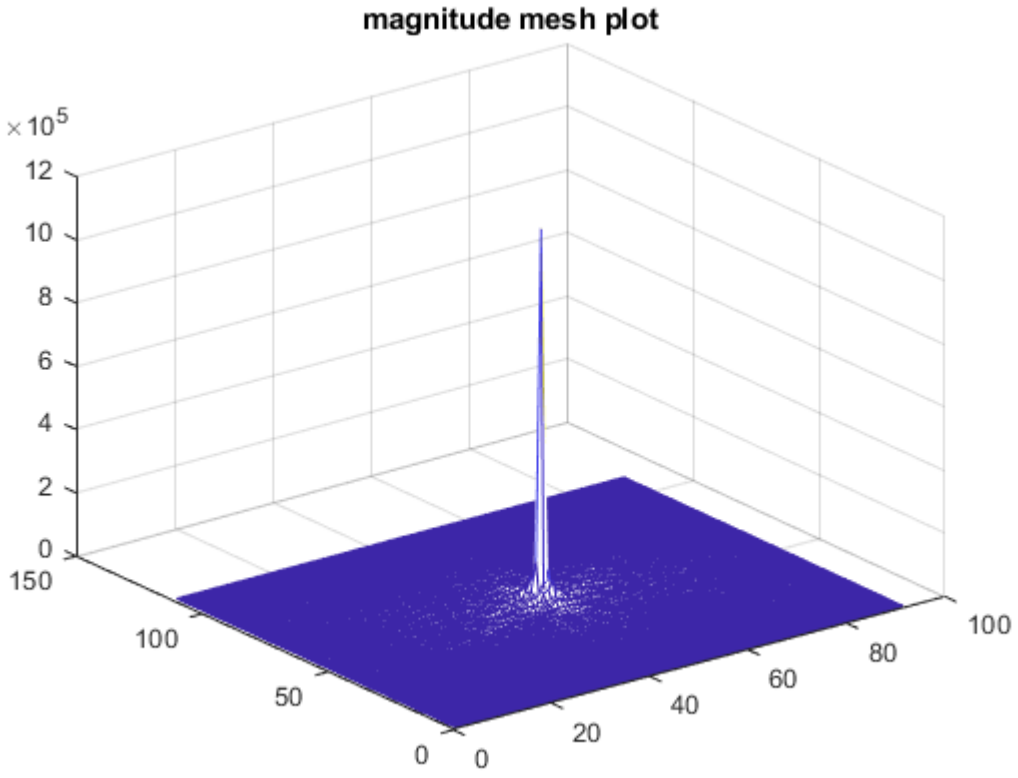
در این قسمت به قسمت فاز و اندازه تصویر هرکدام به یک نسبت معین SNR() اضافه میکنیم و بعد تصاویر را جدا جدا رسم میکنیم که مثلا اول با فوریه گرفتن قسمت فاز و اندازه تصویر را جدا میکنیم و بعد یک بار برای قسمت فاز آن نویز اضافه میکنیم و با قسمت اندازه بدون نویزی اولیه ترکیب کرده و عکس هارا دوباره میسازیم، و یکبار با قسمت اندازه نویزی شده و فاز دست نخوره ترکیب میکنیم و تصویر جدید را بازسازی میکنیم.

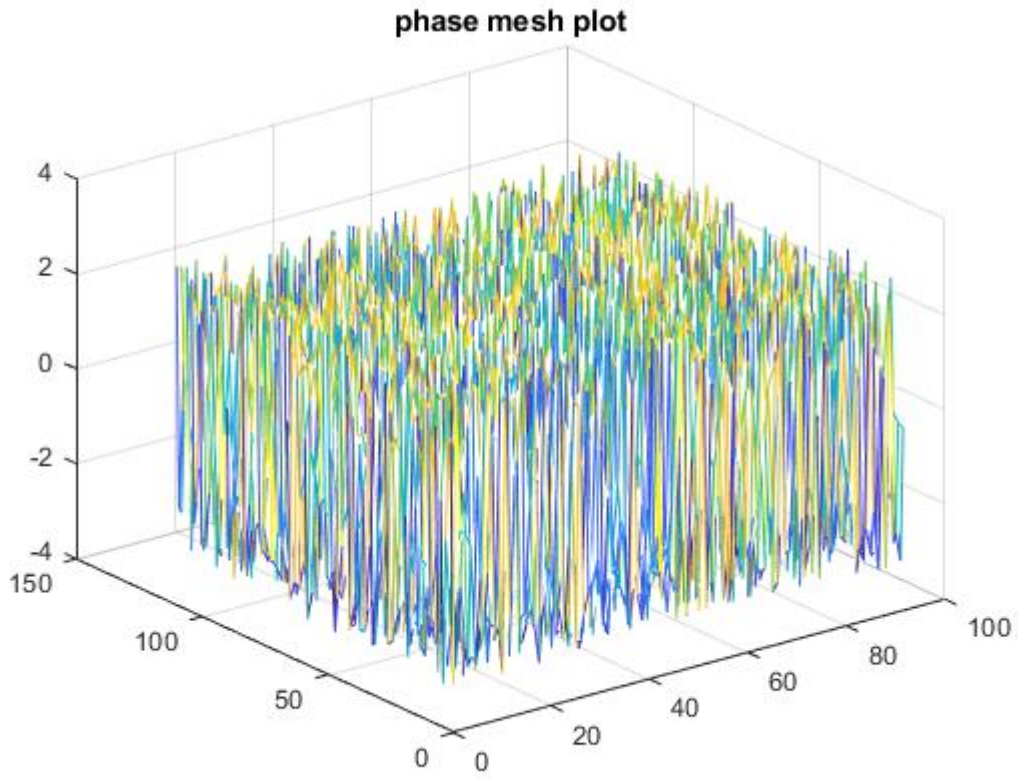
برای اینکه بتوانیم فاز های دست خورده(یا دست نخورده) را با اندازه دست نخورده(یا دست نخورده) ترکیب کنیم اینگونه پیش میرویم:

$$new_{image} = \{absolute\ value\ of\ FFT\} \times e^{j(phase\ of\ FFT)}$$

برای رسم تصاویر بعد از بازسازی باید مقادیر مثبت را در نظر بگیریم که باید از دستور abs() و بعد از دستور uint8() استفاده کنیم.




magnitude mesh plot
×10^5
12
10
8
6
4
2
0
150
100
50
0
0
20
40
60
80
100


phase mesh plot
4
2
0
-2
-4
150
100
50
0
0
20
40
60
80
100

Main photo

noisy amplitude SNR=0.25

noisy phase SNR=0.25

noisy amplitude SNR=0.5

noisy phase SNR=0.5

noisy amplitude SNR=1.00

noisy phase SNR=1.00

برای این قسمت نیز باید دو تصویر را انتخاب کنیم و قسمت فاز و اندازه آن هارا جا ب جا کنیم، و نتیجه قسمت قبل به وضوح این بود که چون وقتی قسمت فاز دجار نویز میشد در تصویر تداخلی بسیار زیاد بوجود می آمد.

ولی در قسمت اندازه که مورد اغتشاش با نویز قرار گرفته بود این گونه بود که تغییر زیادی نسبت به تصویر اصلی به وجود نیامد.

main image of p2

w img with phase of p2 and phase of p

main image of p3

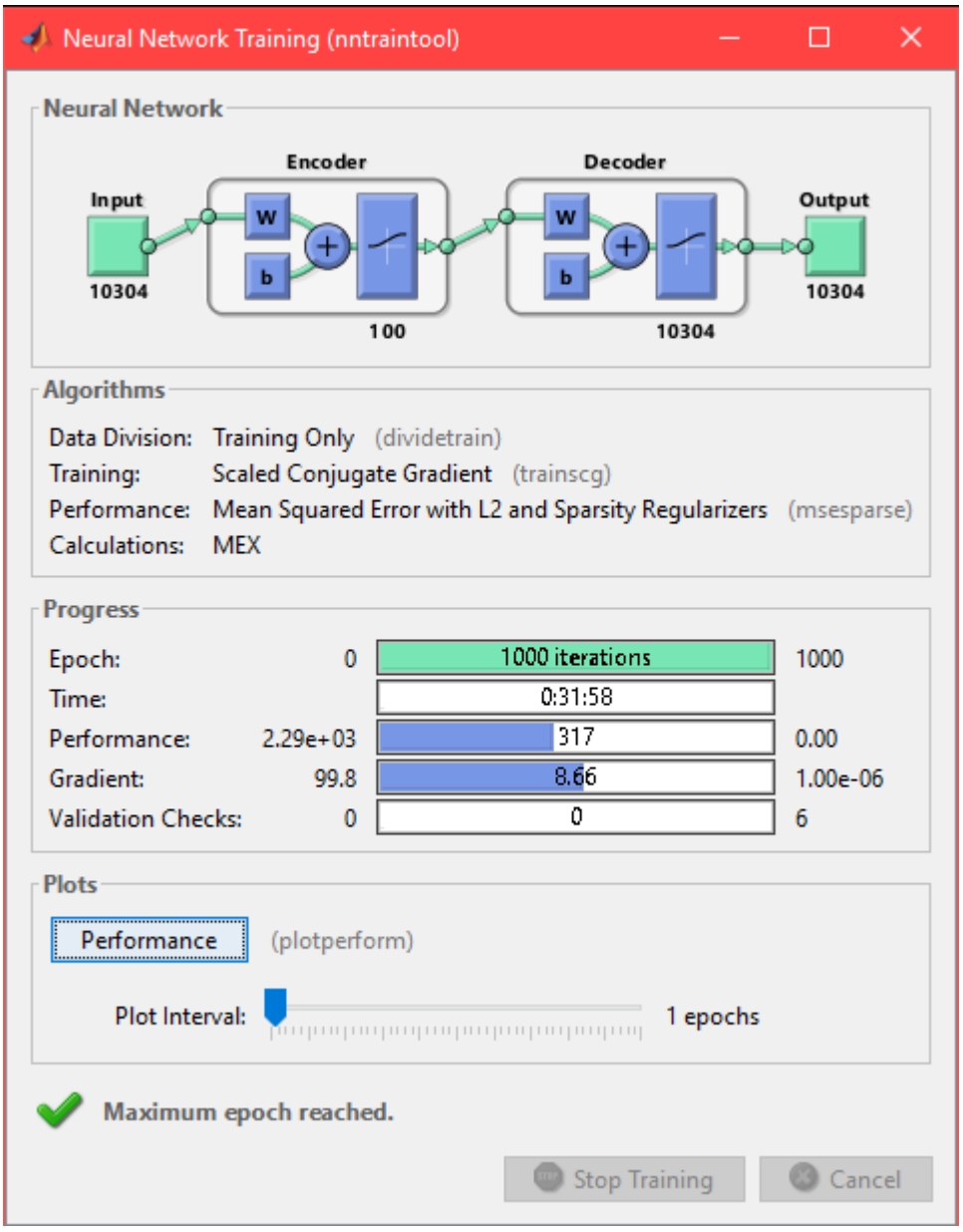
w img with phase of p3 and phase of p

ترتیب عکس هارا به گونه ای قرار داده ام که میتوان فهمید تصویری که ساخته میشود بیشتر به فاز آن مربوط است یعنی با توجه به کد اگر فاز تصویر مربوط به اولی بود و اندازه مربوط به دومی ، تصویر حاسله شبیه اولی میشد همانطور که در ترتیب بالا با توجه به تیتر ها میتوان فهمید گواه بر این قضیه است.

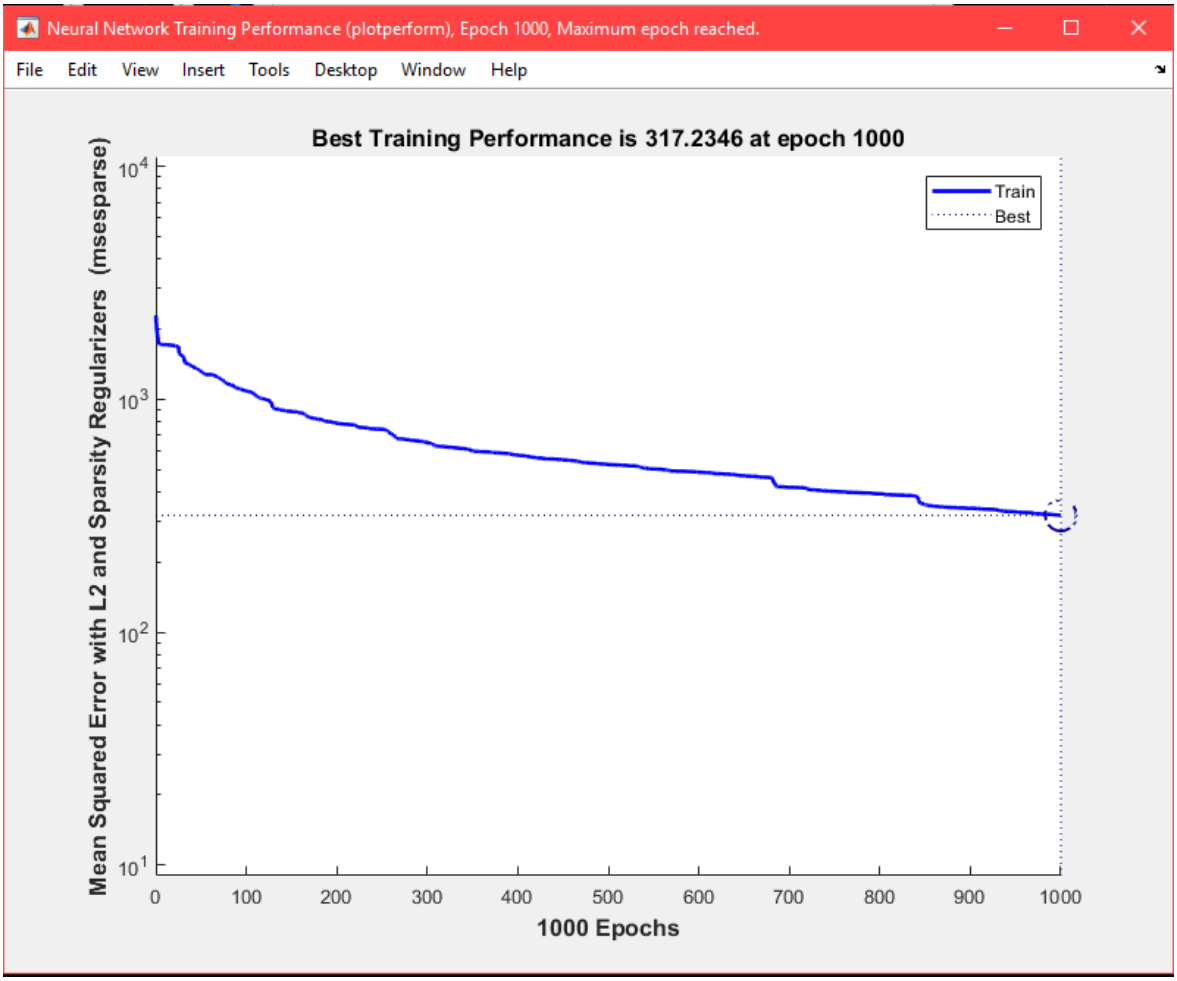
Face Recognition

اول تمامی دیتا هارا که در 40 فولدر و هر کدام 10 عکس میباشد را دی یک سلول 10 در 40 ذخیره میکنیم و بعد برای آموختن سیستم شبکه عصبی از این استفاده میکنیم که با این تفاوت که 8 عکس از 10 عکس را میگیریم و بعد از آن 2 عکس باقی مانده برای تست خروجی استفاده میکنیم .

برای خروجی های سیستم ترین شده و نمودار همگرایی سیستم میتوان به این عکس ها اشاره کرد:



و نمودار همگرایی برای epoch۱۰۰۰ مثل زیر است:



بعد از ترین کردن سیستم دو دسته عکس آماده کرده ام که خروجی ترین شده در دوحالت را بررسی کنم:

1.وقتی که خود عکس جزو عکس های ترین شدن سیستم شبکه عصبی بوده است:

2. وقتی که خود عکس ها جزو عکس های ترین نبوده اند و برای تست انتخاب شده اند:

حال بعد از این برای تست کردن ۸۰ عکس باقی مانده برای سیستم اسفاده میکنیم که نتایج را میتوان به صورت زیر دریافت کرد :

```
TotalAccuracy = 98.2500
TrainDataAccuracy = 100
TestDAtaAccuracy = 91.2500
```

ماتریس ترین شده در فایل ارسال شده قرار گرفته است!

.

.

.