

بسمه تعالی



دانشگاه صنعتی شریف

دانشکده مهندسی برق

فاز اول پروژه درس سیگنال‌ها و سیستم‌ها

نگارنده

امیرمهدی سلیمانی‌فر

شماره دانشجویی

۹۸۱۰۱۷۴۷

استاد درس

دکتر خلج

بهار ۱۴۰۰

۱- در این قسمت دو فیلتر تصویر میانگین و گاوسی را بر روی یک نقاشی پرتره اعمال می‌کنیم. نتیجه برای فیلتر Mean بصورت زیر در آمده است:



برای تصویر سیاه و سفید که بر روی آن فیلتر Mean اعمال شده است، نتیجه زیر بدست می‌آید:



در این قسمت مشاهده می‌شود که اگرچه در تصویر رنگی مقداری تصویر ساختگی (خصوصاً در مرزهای لبها و دماغ) در تصویر سیاه و سفید چنین اثری مشاهده نمی‌شود، بلکه تنها مقداری از نویز (که البته در اینجا مشهود نیست) کم شده و جزئیات از دست می‌روند.

با اعمال فیلتر Gaussian بر روی تصویر رنگی نیز نتایج زیر بترتیب برای تصویر رنگی و سیاه و سفید قابل مشاهده است:

Original (Self portrait with bandaged ear)



Gaussian Filter (Retains detail and produces few artifacts)



Original (Gray scaled image of the painting)

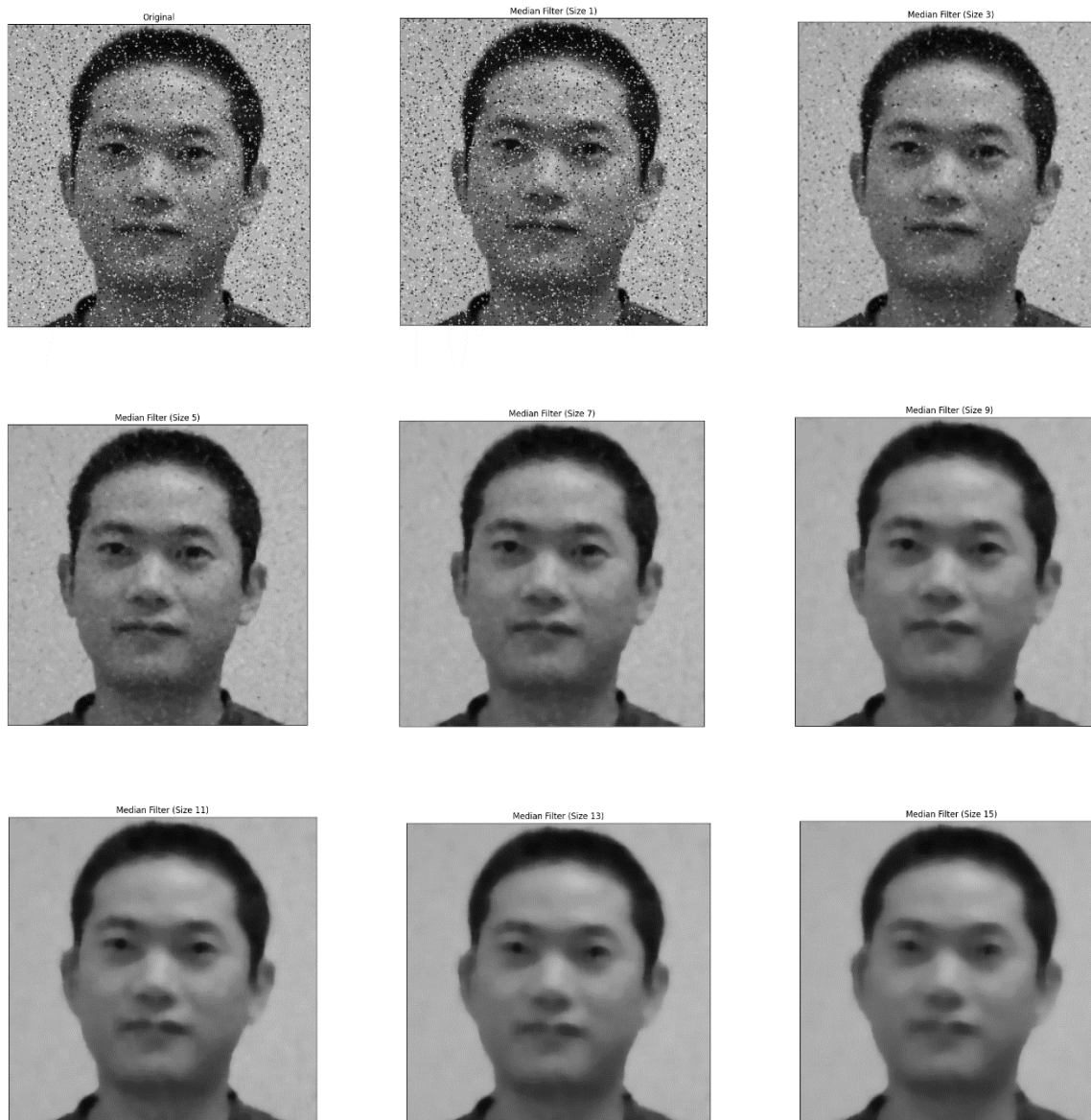


Gaussian Filter (Retains detail and does not produce artifact)



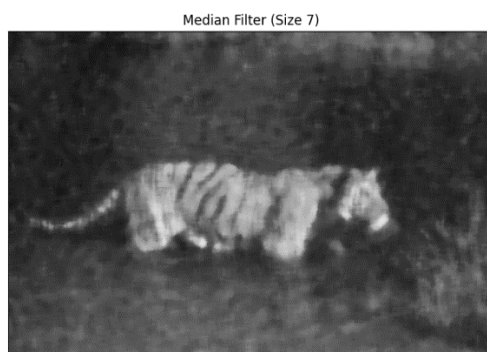
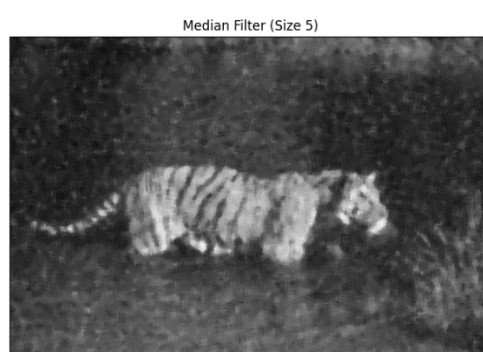
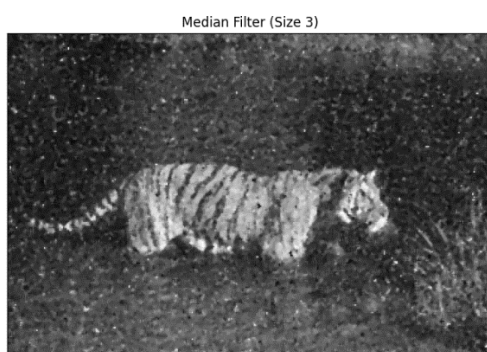
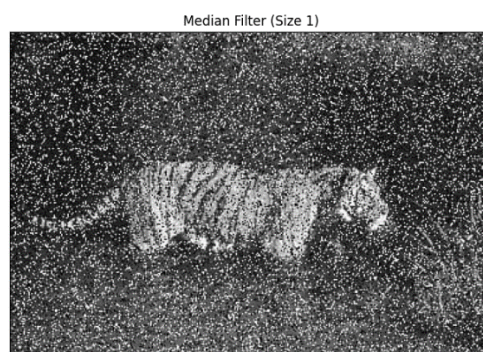
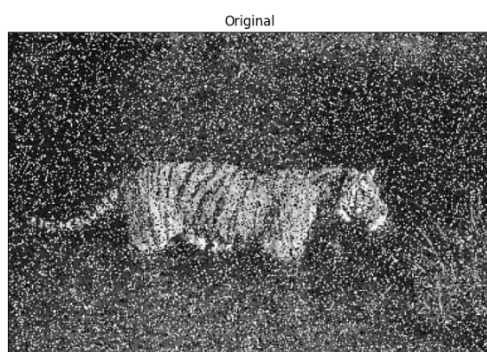
در فیلتر Gaussian مقدار موارد ساخته شده توسط فیلتر کمتر است (این مورد در چند تصویر دیگر نیز بررسی شده است) البته نمی‌توان از آن نتیجه کلی گرفت چرا که در فیلتر در موارد خاصی بهتر عمل می‌کند، اما بطور کلی بنظر می‌رسد برای اکثر موارد فیلتر Gaussian گزینه بهتری برای حذف نویز بدون بوجود آوردن موارد اضافی جدید باشد. مشابه با فیلتر Mean در اینجا نیز در عکس سیاه و سفید، artifactهای جدید مشاهده نمی‌شوند.

نکته مهم دیگر آن است که با مقایسه در فیلتر Gaussian اطلاعات کمتری نسبت به فیلتر Mean در حالت‌های مشابه از دست رفته است. برای بخش دوم تصاویر داده شده با اعمال اندازه‌های مختلف فیلتر Median نویزگیری شده‌اند که نتایج در زیر آورده شده‌اند:



در میان این موارد اعمال فیلتر Median با اندازه `figure_size=5` بهترین نتیجه را در برداشت چرا که مقدار زیادی از نویز گرفته شده است و در عین حال میزان جزئیات تصویر به حد قابل قبولی بالا است. البته اندازه 3 نیز گزینه مناسبی برای دریافت جزئیات بیشتر در ازای نویز بیشتر است.

نتایج مربوط به بکارگیری فیلتر با اندازه‌های مختلف بر روی تصویر دوم نیز بصورت زیر است:

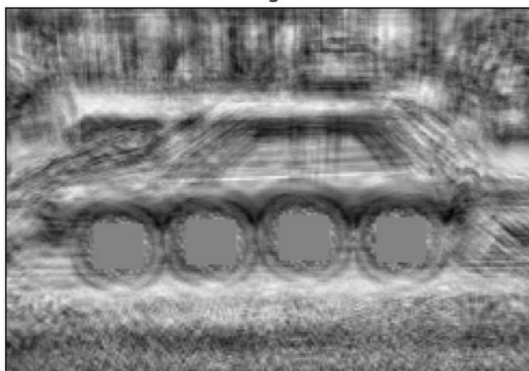


در میان این موارد نیز بکارگیری فیلتر با اندازه 3 موجب بدست آمدن بهترین نتیجه شده است، البته با این که کمی نویز در تصویر وجود دارد اما جزئیات عکس بسیار بیشتر از فیلتر با اندازه 5 است و به همین دلیل به آن ترجیح داده شده است. مقایسه بهترین نتیجه و تصویر اصلی نیز در فایل `res03.jpg` آورده شده و قابل مشاهده است.

۲- در این قسمت با پایین آوردن threshold تا میزان 0.4 تابع قادر به پیدا کردن چهار چرخ است که در زیر آورده شده است:

Template matching correlation coefficient

Matching Result



Detected Point

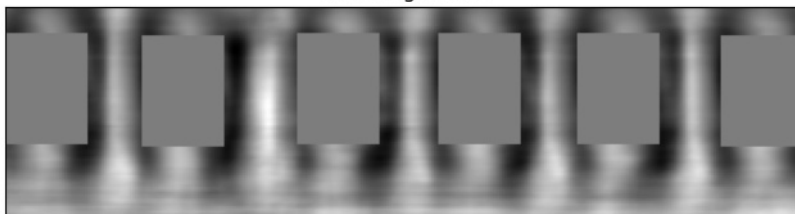


با بالاتر بردن میزان دقت بدلیل وجود تفاوت در این مقادیر اولین چرخ از سمت راست و با افزایش بیشتر threshold اولین چرخ از سمت چپ یافت نخواهند شد. البته میزان شباهت دو چرخ وسط به یکدیگر بالا بوده و بالاتر از 0.8 است. همچنین با پایین تر آوردن مربع‌های سبز نامربوط بسیار شکل خواهند گرفت. برای match making نیز از میان 6 مدل موجود TM_CCOEFF_NORMED کمک گرفته‌ایم که در بررسی تصاویر مشابه دقت بیشتری داشت (که البته این نتیجه‌گیری قابل تعمیم به کل حالات نیست).

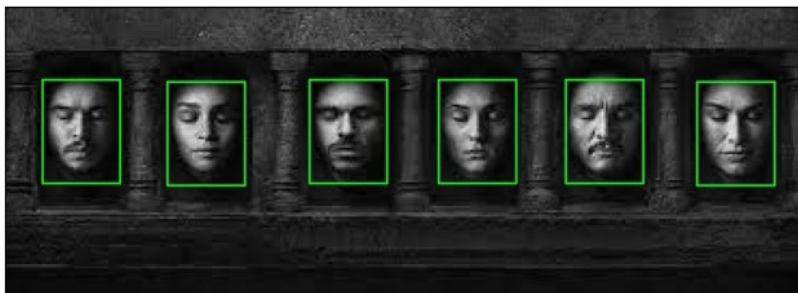
برای حالت بعدی از تصویر آورده شده در پایین استفاده کرده‌ایم که کیفیت آن برای کاهش توان پردازشی مورد نیاز به مقدار لازم پایین آورده شده و سیاه و سفید شده است. در اینجا دقت تشخیص چهره‌ها بدلیل کیفیت پایین و شباهت حالت قرارگیری بسیار بالا بوده و تا threshold برابر با 0.86 تمامی تصویرها شناسایی می‌شوند که نتایج آن در زیر موجود است (نتایج کامل تصاویر بخش قبل و تصاویر اولیه مورد استفاده در پوشه images موجود است):

Template matching with correlation coefficient

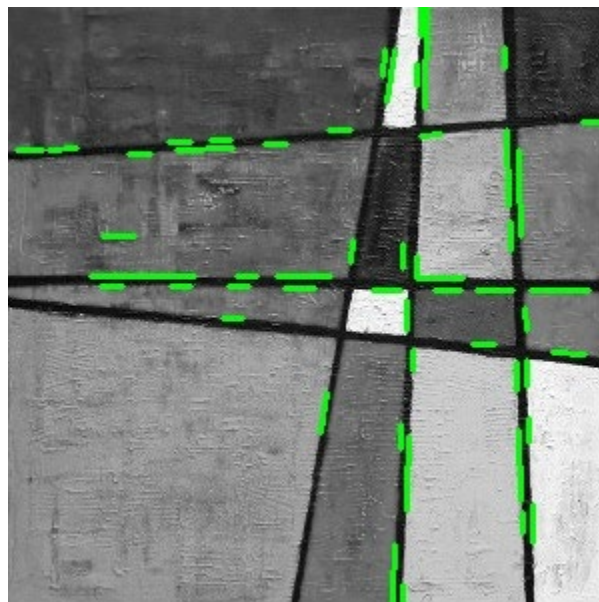
Matching Result



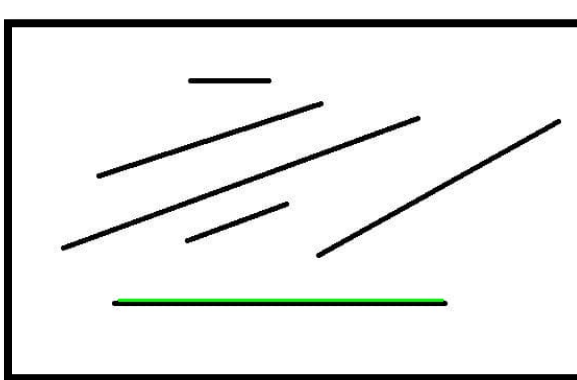
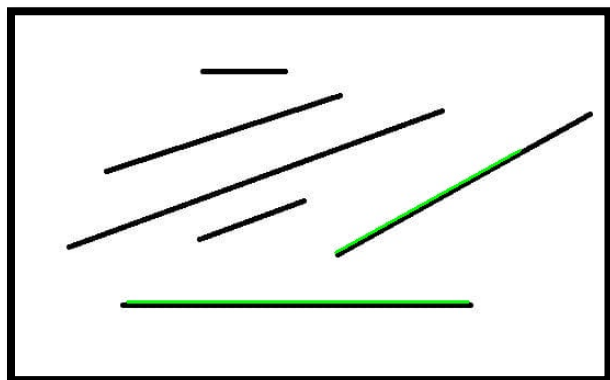
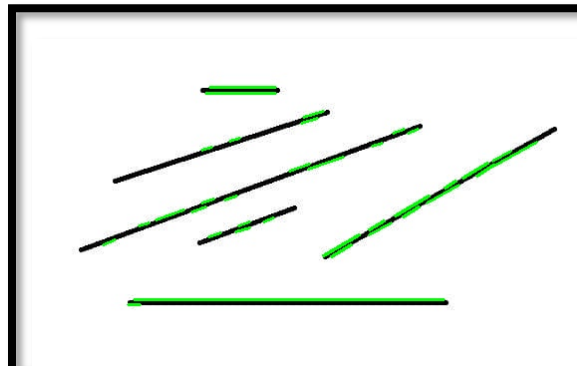
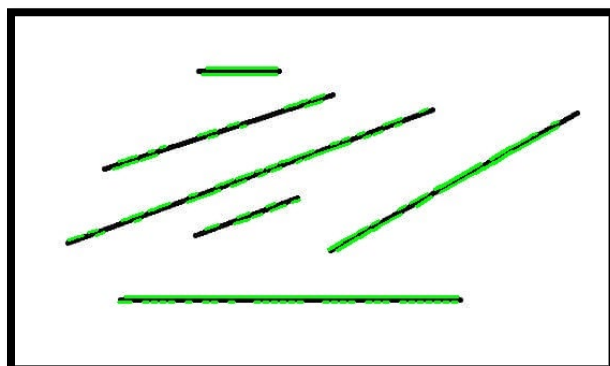
Detected Point



۳- تصویر انتخابی و خطوط سبزی که نشان‌دهنده خطوط شناسایی شده با استفاده از تابع HoughLinesP در زیر آورده شده است:

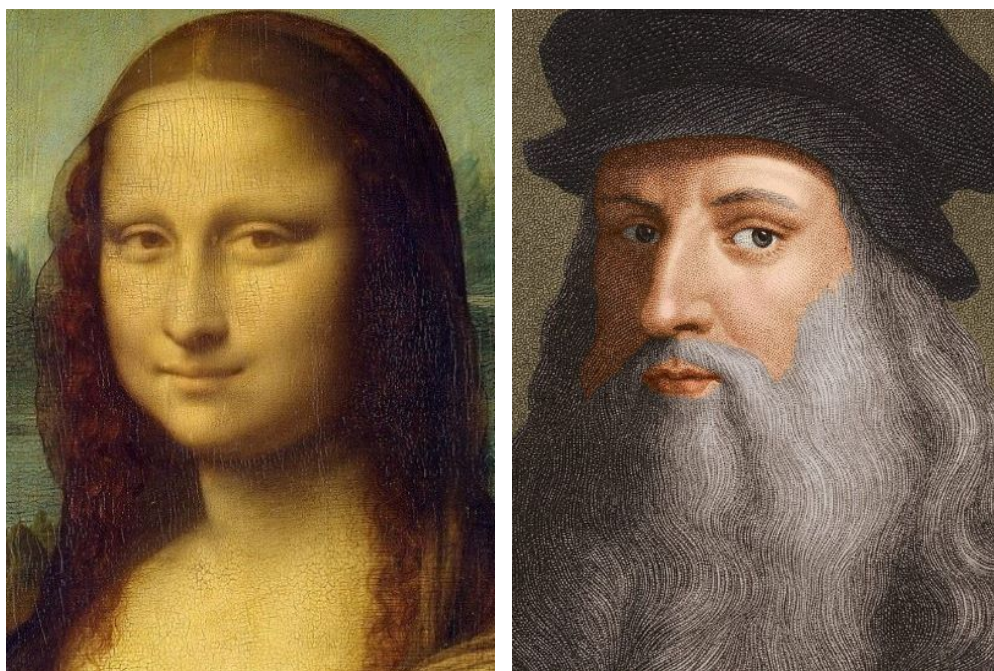


در بخش بعدی اقدام به شناسایی خطوط با یک اندازه‌ی حداقلی خواهیم کرد که نتایج مربوطه برای minlength به مقادیر ۵، ۱۰، ۱۰۰ و ۲۵۰ به ترتیب در زیر آورده شده است:

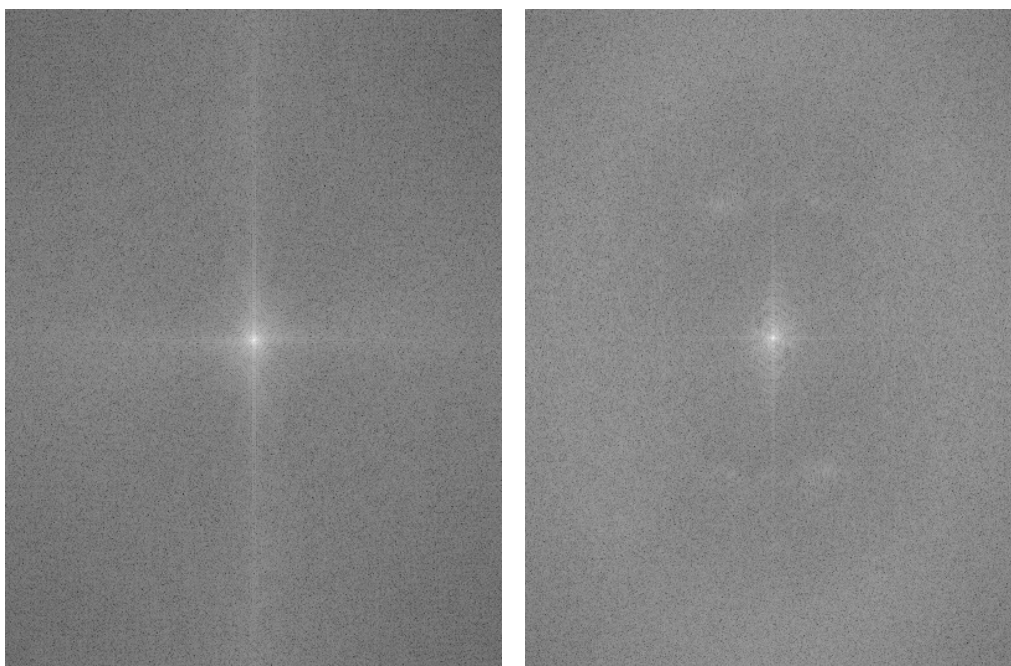


در حداکثر مقدار طول تنها یک پاره‌خط و در مقدار بعد از آن تنها دو تا تشخیص داده می‌شوند. اما با زیاد شدن مقدار تعداد پاره‌خط‌های پیدا شده روی خطوط اصلی زیاد شده و در کنار یکدیگر خط اصلی را تشکیل خواهند داد.

۴- تصاویر اصلی در پوشه تصاویر موجود است، همچنین تصاویر منطبق شده که از نقاشی مونا لیزا و نقاشی ای که احتمالا پرتراهی از لئوناردو داوینچی است نیز در زیر آورده شده اند:



هر دو تصویر را به دامنه فرکانس انتقال می‌دهیم که بصورت زیر خواهد بود:



حال برای مرحله بعدی می‌خواهیم از تصویر سمت راست فرکانس‌های پایین و از تصویر سمت چپ فرکانس‌های بالا را حفظ کنیم تا بتوانیم یک تصویر هایبرید را ایجاد کنیم. برای آن از یک فیلتر دو بعدی گاوسی استفاده خواهیم کرد.

با توجه به اینکه گفته شده است مقادیر r و s در گزارش مطرح شوند، در نتیجه نهایی $s = 15$ و $r = 25$ است. البته نتایج با دو اندازه دیگر که در آن $s = 25$ و $r = 15$ نیز بدست آمده است که البته نتیجه اول بدلیل بهتر بودن بعنوان نتیجه نهایی در نظر گرفته شده است.

بقیه موارد مربوطه در تصاویر همانطور که خواسته شده بود وجود دارند، تنها در اینجا به نمایش تصاویر پس از اعمال فیلترهای پایین گذر و بالاگذر بسنده می‌کنیم:



همچنین می‌توان تصویر نهایی را که از ترکیب این دو بوجود آمده است، مشاهده کرد. در آخر لازم به ذکر است که بخش‌هایی که از کد که برای تولید برخی از نتایج به کار رفته بودند، هنوز هم در کد موجود و قابل بررسی هستند. با توجه به اینکه نامگذاری‌ها بصورت مرتب و خوانا انجام شده است تنها در مواردی که عمل جدیدی صورت گرفته است دلیل آن ارائه گردیده و به موارد دیگر نیاز نیست.

بعضی از dependencyهای موجود در کد مانند CV2 نیز براحتی بوسیله pip قابل راه‌اندازی هستند.