



گزارش کار تمرین اول بینایی ماشین

استاد : دکتر عظیمی فر

اعضای گروه :

الهه سالاری
شماره دانشجویی ۹۹۳۱۱۵۷

حیدر دهقان
شماره دانشجویی ۹۹۳۱۱۵۴

پاییز ۱۴۰۱

بخش اول

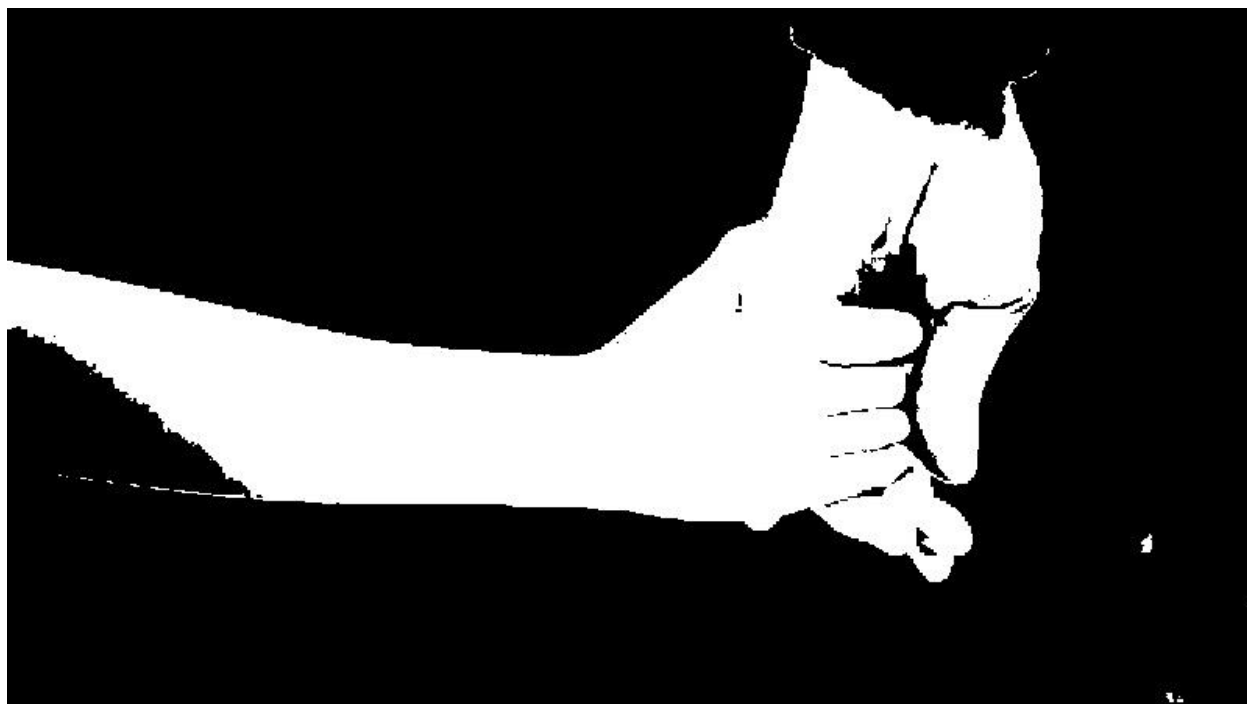
صورت سوال :

روشي براي جداسازي دست از پس زمينه تصوير png.hands ارائه دهيد. خروجي برنامه شما بايد، يك تصوير دودويي باشد كه، ناحيه دست بايد سفيد رنگ و پس زمينه سياه رنگ باشد.

پاسخ :

برای انجام این قسمت از تمرین از مقیاس رنگ HSV یا (Hue Saturation Value) استفاده شده و روش کار به این صورت است که یک رنج از تیف رنگ که گمان میکنیم رنگ پوست دست در این رنج باشد را به مقدار hsv تبدیل میکنیم و سپس با استفاده از متود inRange که در لایبرری openCV موجود میباشد این رنج را از بقیه جدا میکنیم . کد های این تمرین در فایل های ارسالی در فایلی به نام part_one.py قرار دارد .

نتیجه:



بخش دوم :

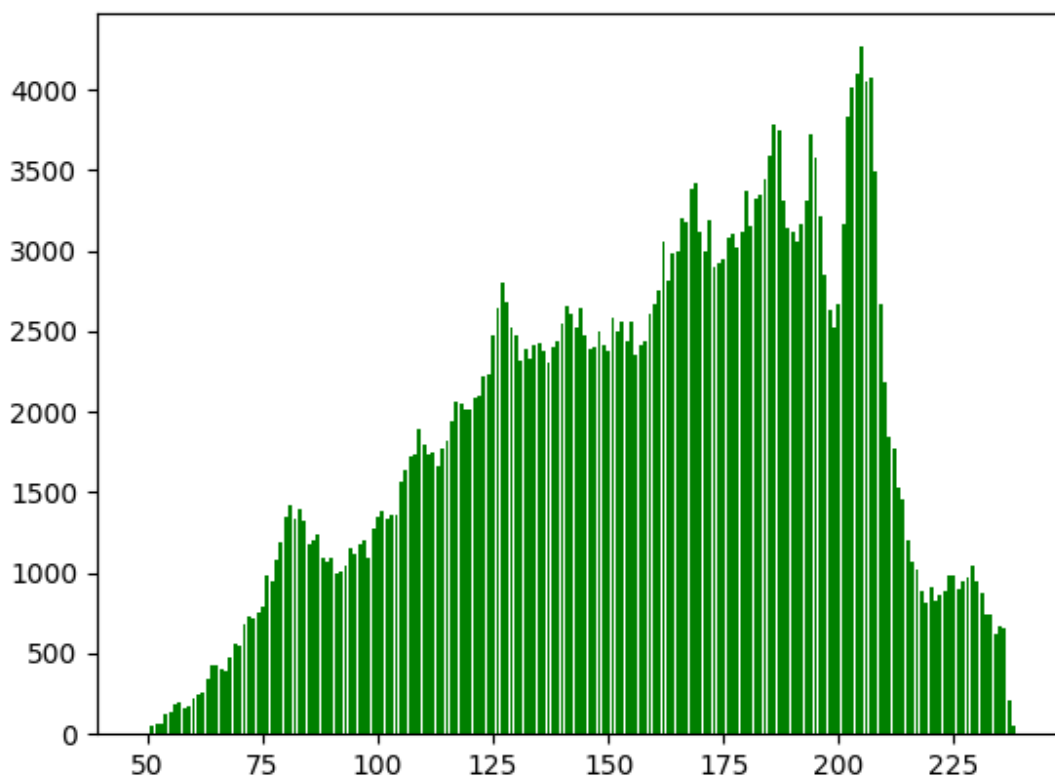
صورت سوال :

تصویر png.hands را به فضای خاکستری انتقال دهید و هیستوگرام آن را بدست آورید و آنرا تفسیر کنید.

پاسخ :

برای انجام این قسمت از تمرین ابتدا با استفاده از متود image از لایبری PLI تصویر را خوانده و سپس با استفاده از همین لایبری آن را به gray تبدیل میکنیم حال به راحتی میتوان تصویر را به یک آرایه تبدیل کرد سپس آرایه را برای استفاده از متد Counter از لایبری collections که از لایبری های بیسیک پایتون هست ریشپ میکنیم و سپس با استفاده از این متد تعداد تکرار های هر مقدار را به دست می آوریم و در هایت با استفاده از این اطلاعات هیستوگرام تصویر را رسم میکنیم . کد های این تمرین در فایل part_two.py در فایل ها ارسال شده قرار دارد .

با توجه با هیستوگرام به دست آمده از تصویر برداشت ما این است که طیف رنگ بین ۱۵۰ تا ۲۰۰ بشتترین رنگ های موجود در تصویر است که احتمالا مربوط به پس زمینه میشوند و سپس طیف رنگی ۱۰۰ تا ۱۵۰ را داریم که احتمالا مربوط به دست ها می شود



بخش سوم (الف) :

صورت سوال :

یکی از نویزهای معروف در تصویر، نویز سفید یا گوسی است. جهت برطرف کردن این نویز از فیلترهای پایین گذر یا میانگین گیری استفاده میشود. برای اینکار یک فیلتر میانگینگیری روی تصویر لغزانده میشود (کانوولوشن). ابعاد فیلتر تاثیر مستقیم روی خروجی دارد. برنامه ای بنویسید که روی تصاویر ۱ تا ۳، فیلتر با ابعاد ۳*۳، ۵*۵، ۷*۷ اعمال گردد. نتایج فیلترهای مختلف را مورد مقایسه قرار دهید.

پاسخ :

برای انجام این بخش از تمرین در ابتدا تصویر را خاکستری و سپس با استفاده از فرمول توضیح نرمال یک نویز به تصویر اعمال میکنیم که نتیجه به این صورت میشود .



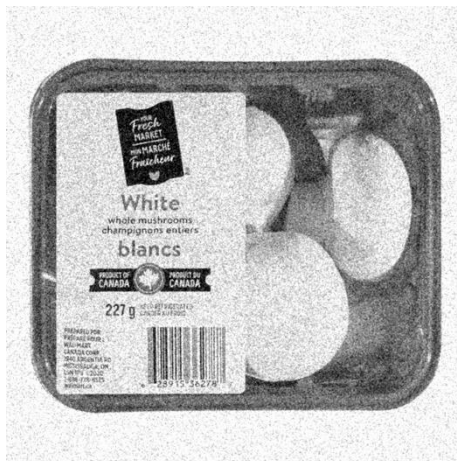
برای از بین بردن این نوع نویز کافی است که از فیلتر میانگین گیری استفاده استفاده کنیم به این صورت که در ابتدا یک پنجره تعریف میکنیم و سپس با استفاده از کانوالو کردن آن روی تصویر مقدار پیکسل ها را با مقدار به دست آمده از میانگین پنجره جایگزین میکنیم .

```
for i in range(data.shape[0]):
    if i + size <= data.shape[0]:
        for j in range(data.shape[1]):
            if j + size <= data.shape[1]:
                w = data[i:i + size, j: j + size]
                # temp[i, j] = np.mean(w)
                temp[i, j] = np.median(w)
```

نتایج برای پنجره ۳*۳ :



نتایج پنجره ۵*۵ :



نتایج پنجره 7×7 :



Image	Window size = (3, 3)	Window size = (5, 5)	Window size = (7, 7)
1.jpg	PSNR Gaussian-Noise: 28.554065626285126 SSIM Gaussian-Noise: 0.2110221105138402	PSNR Gaussian-Noise: 28.418479788101298 SSIM Gaussian-Noise: 0.08976911710416134	PSNR Gaussian-Noise: 28.84549101163861 SSIM Gaussian-Noise: 0.09069520077803014
2.jpg	PSNR Gaussian-Noise: 28.600928567360207 SSIM Gaussian-Noise: 0.2049914549878971	PSNR Gaussian-Noise: 28.468371428888783 SSIM Gaussian-Noise: 0.08505930623180197	PSNR Gaussian-Noise: 28.424895670657392 SSIM Gaussian-Noise: 0.0515452751901611
3.jpg	PSNR Gaussian-Noise: 28.94237439349258 SSIM Gaussian-Noise: 0.21006151456804037	PSNR Gaussian-Noise: 28.84549101163861 SSIM Gaussian-Noise: 0.09069520077803014	PSNR Gaussian-Noise: 28.843669886161763 SSIM Gaussian-Noise: 0.054795308348277616

تمرین سوم (ب) :

صورت سوال :

یکی دیگر از نویزها، نویز فلفل و نمک (pepper and Salt) است که با فیلتر میانه برطرف میگردد. روی تصاویر ۱ تا ۳ نویز فلفل و نمک ایجاد نمایید و سپس فیلتر میانه با ابعاد 3×3 ، 5×5 و 7×7 را روی تصویر اعمال نمایید. نتایج فیلترهای مختلف را مورد مقایسه قرار دهید.

پاسخ :

در این قسمت هم روال کار مانند قسمت الف ولی بجای اعمال نويز گوسی نويز salt and pepper اعمال شده و برای رفع آن از فیلتر میانه استفاده میکنیم که به این صورت است که مفادیر موجود در پنجره را سرت میکنیم و میانه را بر میداریم و جایگزین مقدار پیکسل میکنیم که این کار باعث از بین رفتن مقادیر ۰ و ۲۵۵ در تصویر نويزی میشود .

تصاویر بعد از اعمال نويز salt and pepper :



نتایج برای پنجره ۳*۳:



نتایج برای پنجره ۵*۵ :



نتایج پنجره ۷*۷ :



Image	Window size = (3, 3)	Window size = (5, 5)	Window size = (7, 7)
1.jpg	PSNR Salt&Pepper-Noise: 34.430082248959245 SSIM Salt&Pepper-Noise: 0.7283156228295867	PSNR Salt&Pepper-Noise: 33.559006954163614 SSIM Salt&Pepper-Noise: 0.706869562487496	PSNR Salt&Pepper-Noise: 33.21941817849061 SSIM Salt&Pepper-Noise: 0.6860080538451364
2.jpg	PSNR Salt&Pepper-Noise: 35.60235083097695 SSIM Salt&Pepper-Noise: 0.7368503234421379	PSNR Salt&Pepper-Noise: 33.9388738700474 SSIM Salt&Pepper-Noise: 0.6865464702240313	PSNR Salt&Pepper-Noise: 33.9307161223673 SSIM Salt&Pepper-Noise: 0.7323312666052569
3.jpg	PSNR Salt&Pepper-Noise: 35.09011489407193 SSIM Salt&Pepper-Noise: 0.7398275908468438	PSNR Salt&Pepper-Noise: 33.25937792976972 SSIM Salt&Pepper-Noise: 0.6543530214696669	PSNR Salt&Pepper-Noise: 32.81801606029798 SSIM Salt&Pepper-Noise: 0.5976047219681124

تمرین چهارم :

صورت سوال:

لبه یابی و عملیات مورفولوژیکی: روشی برای تشخیص (مکان قرارگیری) بارکدها در تصاویر ۱ تا ۳ ارائه دهید (در این تمرین مجازید از کلیه ابزارهای آماده استفاده کنید). راهنمایی: برای تشخیص بارکدها میتوانید از ابزارهای لبه یابی و عملیات مورفولوژیکی استفاده کنید.

پاسخ:

ابتدا تصاویر را به فرمت Gray scale تبدیل می کنیم. با استفاده از sodel edge detection لبه های آبجکت های داخل تصویر را مشخص می کنیم. تصویر بدست آمده از edge ها را blur کرده و سپس با ترشولد ۲۰۰ تصویر را باینری می کنیم. با استفاده از عملیات مورفولوژی اقدام به پیدا کردن محدوده بارکد ها می کنیم. ابتدا morphology close با کرنل (۲۵x۲۵) اعمال می کنیم و سپس erosion , dilation را اعمال می کنیم. در نهایت در تصاویری که عملیات مورفولوژی روی آنها اعمال شده findContour زده و دور محدوده بارکد را با مستطیلی نشان می دهیم.

نتایج:





تمرین پنجم

صورت سوال:

معیارهای رایج در بررسی تصویر: با تغییر در تصویر بر اساس شیفت، چرخش در درجات ۳۰ و ۴۵ درجه معیارهای MSE، PSNR و SSIM بر اساس تصویر اصلی و تصویر پایه محاسبه شود.

پاسخ:




جهت شیفت دادن تصویر یک آرایه با مقادیر صفر به اندازه ابعاد تصویر ایجاد می کنیم و تصویر اصلی را به اندازه چند پیکسل در سطر و ستون شیفت می دهیم. برای چرخاندن نیز از ارتاباع `getRotationMatrix2D` استفاده می کنیم که نقطه مرکزی تصویر را می گیرد و به اندازه زاویه ی مشخص شده تصویر را می چرخاند. معیارهای خواسته شده را نیز بر اساس فرمول های زیر به دست می آوریم




$$MSE = \text{mean}((Original - changedImage)^2)$$




$$PSNR = 10 * \log_{10} \div \frac{(MAX_I)^2}{MSE}$$

$$MSSIM(\mathbf{X}, \mathbf{Y}) = \frac{1}{M} \sum_{j=1}^M SSIM(\mathbf{x}_j, \mathbf{y}_j)$$

نتائج:

Shifted image	Rotate 30 Degree	Rotate 45 Degree
		
MSE Main-Shift: 25155.505403703704 PSNR Main-Shift: 4.124473136313196 SSIM Main-Shift: 0.2651203579064444	MSE Main-Rotate-30: 0.28915004442388675 PSNR Main-Rotate-30: 0.28915004442388675 SSIM Main-Rotate-30: 0.28915004442388675	MSE Main-Rotate-45: 0.28034678391127704 PSNR Main-Rotate-45: 0.28034678391127704 SSIM Main-Rotate-45: 0.28034678391127704

Rotate 45 Degree	Rotate 30 Degree	Shifted image
		
MSE Main-Shift: 23969.09148 PSNR Main-Shift: 4.334287879398206 SSIM Main-Shift: 0.3581326456808229	MSE Main-Rotate-30: 0.3427058061549578 PSNR Main-Rotate-30: 0.3427058061549578 SSIM Main-Rotate-30: 0.3427058061549578	MSE Main-Rotate-45: 0.33065407798130014 PSNR Main-Rotate-45: 0.33065407798130014 SSIM Main-Rotate-45: 0.33065407798130014

Rotate 45 Degree	Rotate 30 Degree	Shifted image
		
MSE Main-Shift: 27173.648460812634 PSNR Main-Shift: 3.789324081319483 SSIM Main-Shift: 0.263196687123688	MSE Main-Rotate-30: 0.2697967027568704 PSNR Main-Rotate-30: 0.2697967027568704 SSIM Main-Rotate-30: 0.2697967027568704	MSE Main-Rotate-45: 0.2623331519726986 PSNR Main-Rotate-45: 0.2623331519726986 SSIM Main-Rotate-45: 0.2623331519726986

