

توجه : در انجام تمرین استفاده از اینترنت و مشورت مجاز میباشد اما کپی کردن تمرین حتی یک قسمت مجاز نمیشود و در صورت مشاهده نمره کل تمرین صفر در نظر گرفته خواهد شد. برای همه تمرین نیاز است که گزارش کامل همراه کدها تحویل داده شود و گزارش نیز نمره قابل توجهی دارد. در کدها هم به میزان نیاز حتما کامنت گذاری انجام شود. در صورتی که در انجام تمرین مشکلی بود، آن را در قسمت "پرسش و پاسخ" تمرین مطرح نمایید.

لطفاً به ورودی و خروجی خواسته شده دقت نمایید. احتمال دارد برخی از تمرین ها توسط برنامه تصحیح شوند.

➤ همچنین برای دسترسی به ویدیوهای مورد استفاده در این تمرین می‌توانید از لینک زیر استفاده نمایید:

<https://drive.google.com/drive/folders/1cAxCbhlYYItRkJBURyEn4X5wNN6T4QMC?usp=sharing>

در این تمرین کار با Git (گیت محبوب ترین Version Control System موجود است) نمره ی امتیازی دارد. سعی کنید هر قسمت از تمرین را که انجام دادید یک Commit نیز به Repository Local خود اضافه کنید. فایل ".git" را همراه با بقیه پاسخ های خود آپلود کنید.

(الف)

در این تمرین قصد داریم با توابع موجود در OpenCV آشنا شویم و کار با آنها را تمرین کنیم.

۱) الف- در این تمرین بعد از نصب موفق پایتون و OpenCV تصویر 1.jpg ضمیمه را در کد خواننده شماره ی دانشجویی خود را در گوشه چپ تصویر به صورت texture آورده و آن را در دو حالت RGB و Grayscale نمایش دهید سپس به واسطه فشردن کلید s آنها را سیو کرده و پنجره را ببندید و یا با فشردن دکمه e تنها پنجره را ببندید.

ب- تصویر football.jpg را خواننده و دور توپ موجود در تصویر یک مستطیل سبز رنگ (به صورت تقریبی) قرار دهید. سپس سعی کنید با استفاده از توپ موجود در تصویر اولیه، توپ دیگری به تصویر در بخش روی چمن اضافه کنید و نتیجه را ذخیره کنید.

۲) برنامه ای بنویسید که تصویر space.jpg را خواننده و با استفاده از توابع Highgui OpenCV رابط کاربری طراحی کنید که کاربر با استفاده از یک اسلایدر میزان چرخش تصویر را دریافت و تصویر مورد نظر را دوران دهد در خروجی کاربر تصویر اصلی را در سمت چپ و دوران یافته را سمت راست به صورت کنار هم قرار دهید. همچنین نقطه ای دلخواه از

تصویر اصلی را به دوران یافته اش در تصویر راست ترسیم کنید. نمونه ای از این کار در فایل 2.mp4 قابل مشاهده است.

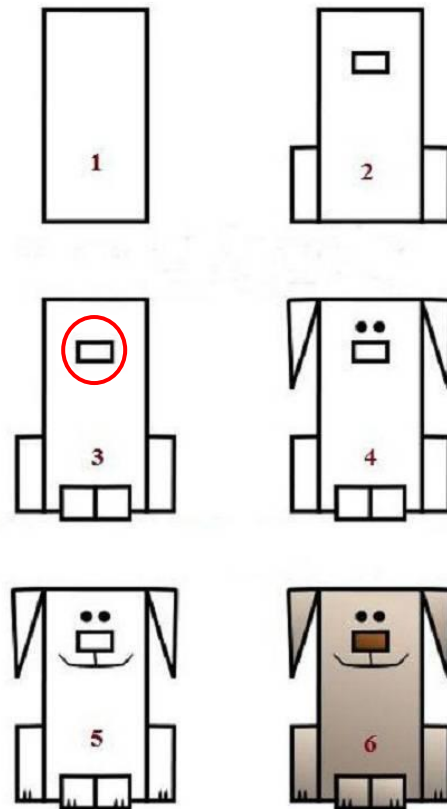
(۳) تصویر 2.jpg را خوانده، در کد سائز آن را به  $400 \times 300$  تبدیل کرده و بر روی آن یک فیلتر پایین گذر و فیلتر لبه یاب لبه های عمودی و افقی اعمال کنید و هر دو خروجی را به صورت مجزا ذخیره کنید همچنین توضیح دهید با استفاده از نتایج حاصل شده چگونه می توان به تصویر بالا گذر شده تصویر اولیه رسید و آن را نیز به طریقی که توضیح می دهید ذخیره کرده و در گزارش خود بیاورید. در گزارش نتایج را تحلیل کنید. (در این سوال نیاز است کرنل ها را خود تعریف کرده و کرنل را خود بر روی تصویر اعمال کنید)

(۴) با استفاده از عملگرهای لبه یاب لبه های موجود در هر دو تصویر 1.jpg و 2.jpg را بدست آورده و نتایج آنها را با هم مقایسه و تحلیل کنید. فیلترهای Canny، Sobel و LoG را بررسی کنید. در الگوریتم های لبه یابی جهت افزایش کیفیت تشخیص لبه ها و حذف نویز چه عملی را پیشنهاد می دهید.

(۵) تصویر limbo.png را خوانده بر روی آن فیلترهای Erosion و Dilation با کرنل ها با سائز متفاوت اعمال نمایید. با استفاده از دو فیلتر مذکور تبدیل های Opening و Closing را پیاده نمایید و نتایج خود را تحلیل کنید.

(۶) برای یافتن لکه (blob detection) توضیح دهید که چه روش هایی وجود دارد و چگونه باید عمل کرد. کدی بنویسید که تصویر شماره 3.jpg را خوانده و در آن بتواند لکه ها را شناسایی کند. توضیح دهید هر پارامتر چه کاری انجام میدهد و با تغییرات آنها چه اتفاقی در نتیجه خروجی می افتد.

(۷) تصویر 4.jpg را خوانده و روشی پیشنهاد دهید که بتوان با آن تمام اجزای مشابه با قسمت مشخص شده با دایره قرمز را در تصویر زیر یافت. چگونه میتوان محل دقیق این قسمت (دقیقا مستطیل مشخص شده) را یافت؟



(ب)

۱) برنامه ای بنویسید که از وب کم به صورت زنده تصویر را گرفته و نمایش دهد. همچنین قابلیت این را داشته باشد که با فشردن دکمه s شروع به ضبط ویدئو کند و سپس با فشردن e متوقف شود یا با فشردن e در دفعه اول متوقف شود.

۲) ابتدا توضیح دهید چگونه با استفاده از فیلتر median می توان تصویر پس زمینه را در فریم های متوالی یک ویدئوی چندثانیه ای تخمین زد. آیا راه های دیگری نیز برای استخراج فریم پس زمینه به نظر تان می رسد؟ سپس برنامه ای بنویسید که پس از خواندن ویدئوی video.mp4، با استفاده از داشتن فریم پس زمینه و موجود بودن هر فریم، ویدئویی تولید کند (به صورت grayscale) که تنها اجسام متحرک را در ویدئو نمایش دهد.

۳) با استفاده از کد قسمت ب-۱ ویدئویی ذخیره کنید و در این قسمت آن را خوانده و فیلترهای لبه یاب مختلف Sobel، Canny و Prewitt را به فریم های ویدئو اعمال کرده و نتایج را با هم مقایسه کنید. در مورد تفاوت نتایج حاصل از این قسمت و نتایج حاصل از حالتی که ابتدا فیلترینگ گوسی انجام شده و سپس لبه یاب ها اعمال شوند، بحث کنید.