

**بسمه تعالی**

**دانشگاه صنعتی شریف**

**دانشکده علوم کامپیوتر**

**Image processing - دکتر مصطفی کمالی**

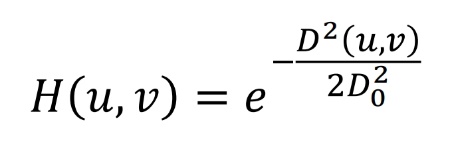
**HW2 q1 report**

**سوال یک -sharpening :**

یک تابع به نام normalize داریم که تقریبا در تمام قسمت ها استفاده شده ، این تابع به اینصورت است که ماتریس ورودی را گرفته ، کمترین مقدارش را از هم درایه ها کم میکند که باعث میشود کمترین مقدار برابر صفر شود ، سپس همه درایه ها را تقسیم بر بیشترین مقدار کرده و سپس ضربدر 255 کرده و به نوع uint8 میبرد ، لذا باعث میشود که درایه های ماتریس ورودی که اگر بین a تا b بوده اند بین 0 تا 255 و به صورت integer قرار بگیرند و مناسب نمایش و ذخیره کردن شوند.

**الف)** یک فیلتر گوس با اندازه 11×11 با σ=3 میسازیم ، به این صورت که ابتدا به صورت یک بعدی کرنل را ساخته و با np.outer دو بعدی میکنیم ، سپس این فیلتر را بر هر سه کانال عکس اعمال کرده ، و سپس حاصل را از تصویر اولیه کم می کنیم که برابر unsharp mask میشود ، سپس این unsharp mask را ضربدر 0.5 کرده و با تصویر اولیه جمع میکنیم و نتایج را normalize کرده و ذخیره میکنیم ، همچنین برای این که فیلتر گوس نمایس بهتری داشته باشد اندازه آن را از 11 در 11 به 600 در 600 ، resize میکنیم تا بهتر دیده شود.

ب) در این قسمت مشابه قسمت قبل یک فیلتر گاوس با اندازه 11 در 11 و σ=2.5 ساخته و با npimage.laplace از آن لاپلاسین میگیریم ، البته میتوان با دوبار استفاده از np.diff هم به نتیجه مشابهی رسید ولی ndimage کوتاه تر است و نتیجه هم فرقی نمیکند. سپس با signal. convolve2d این LOG را بر تصویر اعمال کرده و سپس حاصل را ضربدر 1.8 میکینم و از عکس اصلی کم میکنیم ، در نهایت هم نتایج را normalize کرده و ذخیره میکنیم . مثل قبل LOG را به اندازه 600 در 600 میبریم که بهتر دیده شود.

ج) برای این دو قسمت آخر یک تابع داریم به نام filter\_generate که یک عکس و یک D که همان cutoff است را میگیرد ، ابتدا یک ماتریس به اندازه img ورودی با همه مقادیر صفر میسازد و سپس طبق فرمول زیر درایه های این ماتریس را پر میکند تا یک ماتریس LPF gaussian ساخته شود.

سپس یک HPF به صورت 1-LPF میسازد و این دو فیلتر را خروجی میدهد. این فیلترها چون در حوزه فرکانس هستند لذا اندازه آنها را برابر با عکس ورودی میسازیم.

حال برای قسمت ج ابتدا عکس را به hsv برده و کانال value آنرا در نظر میگیریم و سپس با تابعی که توضیح دادیم یک فیلتر hpf با D0=80 میسازیم . همانطور که در اسلایدهای درس گفته شده بود از کانال value ، فوریه گرفته و ضربدر عبارت 0.2 HPF + 1 میکنیم .سپس از حاصل فوریه معکوس گرفته و normalize کرده و برابر با کانال value از عکس اصلی قرار میدهیم ، سپس عکس را دوباره به BGR برده و normalize کرده و ذخیره میکنیم . برای نشان دادن اندازه تبدیل فوریه ها ابتدا از مقادیر آنها که موهومی هم بودم abs گرفتیم که اندازه را به ما بدهد ، سپس برای اینکه احتمال داشت بعضی مقادیر صفر باشند همه درایه ها را با 0.001 جمع میکنیم سپس log10 گرفته و normalize شده آنرا ذخیره میکردیم .

د) این قسمت هم مثل قبل ابتدا عکس را hsv کرده و از کانال value فوریه میگریم ، سپس باید این مقدار که همان F است را در عبارت 4π2(u2+v2) ضرب کنیم ، ابتدا یک ماتریس به اندازه ابعاد F با همه مقادیر صفر میسازیم سپس از وسط آن یعنی (u,v)=(0,0) شروع کرده و تمام درایه ها را با فرمول i2+j2 که i,j از 0 تا به ترتیب h/2 , w/2 میرند ، سپس تمام درایه های این ماتریس را ضربدر 4π2 کرده و آنگاه درایه به درایه ضربدر F میکنیم که به عبارت مطلوب خود میرسیم . سپس از این مقدار عکس فوریه گرفته و مقدار real را نگه داشته و ضربدر 0.0000005 میکنیم و با value اولیه جمع میکنیم ، در نهایت مقدار بدست آمده را normalize میکنیم و برابر کانال value عکس قرار میدهیم و تبدیل به BGR میکنیم .

|  |  |
| --- | --- |
| 3 | σ فیلتر گوس در قسمت الف |
| 0.5 | مقدار α در قسمت الف |
| 2.5 | σ فیلتر گوس در قسمت ب |
| 1.8 | مقدار k قسمت ب |
| 0.2 | مقدار k قسمت ج |
| 0.0000005 | مقدار k قسمت د |