

الف) همانطور که در خروجی مشاهده میکنید، چون پس زمینه عکس بیشتر تیره بوده، متعادل سازی به سمت روشن کردن تصویر رفته و تصویر روشن تر شده. همانطور که میبینید مجسمه به شدت روشن شده طوری که خط های صورت مجسمه دیگر پیدا نیست. این از معایب روش متعادل سازی هیستوگرام است. به دلیل اینکه سعی کرده ایم عکس را روشن تر کنیم، قسمت هایی از عکس مثل مجسمه که دارای پیکسل هایی با شدت روشنایی بالا بوده نیز روشن تر شده و جزییات آن دیگر قابل مشاهده نیست. در عکس هایی مانند این عکس که یک سری قسمت تیره و یک سری قسمت روشن داریم، خوب نمیتواند عمل کند.

ب) توضیح تابع ACE1:

در این قسمت یک سایزی برای پنجره در نظر گرفته و هر بار یک زیرماتریس را در نظر گرفته به عنوان عکس و روی آن زیرماتریس عملیات متعادل سازی را انجام داده و تمام پیکسل ها را به یک پیکسل جدید نگاشت میکنیم.

از مزایب این روش میتوان به این اشاره کرد که محلی نگاه کرده و نه سراسری. البته زمانی که سایز پنجره برابر سایز عکس باشد، سراسری است.

از معایب میتوان به این نکته اشاره کرد که باعث پیدایش مربع هایی در تصویر میشود همانطور که در خروجی میبینید، چرا که 2 پیکسل کناری را در نظر بگیرید که یکی در پنجره ای است که تصویر تیره و یکی در پنجره ای با تصویر روشن قرار گرفته. این باعث شده که یک تغییر شدت زیادی بین آن ها به وجود آید. همچنین محاسبات زیاد تر شده نسبت به هیستوگرام عادی.

توضیح تابع ACE2:

این قسمت شبیه به قسمت قبلی پیاده سازی شده و صرفا این فرق را کرده که پیکسل وسط را به یک پیکسل جدید نگاشت میدهیم و بقیه پیکسل ها بدون تغییر باقی میمانند. متغیر های up و down و left و right برای فاصله پیکسل وسط از 4 جهت شده و همچنین padding اضافه شده از نوع تکرار پیکسل های گوشه است.

از مزایا این قسمت میتوان به این اشاره کرد که تصویر بهتر شده و کنتراست بالاتری دارد. چرا که صرفا روی پیکسل وسطی هر پنجره نگاشت را انجام داده. از معایب میتوان به این اشاره کرد که اگر یک پنچره سیاه یا روشن باشد پس از نگاشت دارای نویز شده مثل گوشه راست بالا. همچنین محاسبات پیچیده تر است نسبت به قبل. میتوان آن را بهبود داد اما به هر حال محاسباتش بیشتر شده.

توضیح تابع CLAHE

در این قسمت برای جلوگیری از نویز پیش آمده در حالت قبلی، یک کلیپ لیمیت در نظر میگیریم که باعث شده هر پیکسل حداکثر فرکانسش به آن اندازه شود و مقادیر بریده شده را به صورت یکنواخت بین همه پیکسل ها توزیع کرده. این کار باعث جلوگیری از افزایش شدید مقدار در `cdf` شده و به همین دلیل آن نویز حذف شده. برای پیاده سازی `cdf` نیز از تابع `calc_cdf` قسمت `qb` استفاده کرده. مزایا حذف نویز اضافه شده قسمت قبل. معایب نیز محاسبات زیاد مورد نیاز.

ج) همانطور که در خروجی ها مشاهده میکنید، بهترین نتیجه برای قسمت 3 است. پنجره انتخاب شده بیش از حد بزرگ نیست که باعث میشود جزییات نزدیک هر پیکسل بیشتر در نظر گرفته شده و همچنین `limit` در نظر گرفته شده در حدی است که نویز در قسمت هایی که تصویر بسیار سیاه بوده به خوبی از بین برونده. در قسمت 4 چون حد در نظر گرفته شده زیاد شده، تصویر به شدت تیره شده و نویز همچنان در قسمت بالا سمت راست دیده شده. در قسمت های اول و دوم چون سایز پنجره زیاد است، هر پیکسل تحت تاثیر پیکسل های زیادی قرار گرفته و باعث شده که تصویر به خوبی قابل ملاحظه نباشد. در قسمت دوم که حد بیشتر شده نیز تصویر به شدت تیره شده در حالی که تصویر اول به نسبت بهتر است.