

به نام خدا

گزارش تمرین سری ۵

ارسالان فیروزی ۹۷۱۰۲۲۲۵

(۱) من برای هر دو عکس ورودی ۴ اطلاعات اضافه میخواهم: کجای تصویر هدف و کجای تصویر سورس مورد نظر هست و چه طول و عرضی این قسمت مورد نظر دارد و یک ماسک از جسم که به نام Mask_i2.jpg در کنار فایل ها است.

برای ساخت ماتریس معادلات پواسون من ابتدا به نقاط label اختصاص دادم. تمامی پیکسل های معلوم ۱- هستند و تمام پیکسل های مجهول یک شماره اختصاصی دارند که متناظر با سطر و ستون مخصوص آن نقطه در ماتریس A و C است. این پیکسل های مجهول از طریق ماسک مشخص می شود.

با استفاده از حلقه تو در تو به ازای هر پیکسل مجهول سطر متناظر با آن را در A و C آپدیت می کنم. به اینصورت که اگر آن پیکسل همسایه مجهول بود ضریب متناظر با آن را در ماتریس A در آن سطر یک میکنم و در غیر اینصورت مقدار شدت آن پیکسل را در وکتور C در نظر میگیرم. همچنین تمامی درایه های قطر اصلی این ماتریس برابر با -۴ است. مقادیر گرادیان تصویر سورس در آن پیکسل ها نیز در بردار C در نظر گرفتیم. از طریق کرنل $[[0,1,0],[1,-4,-1],[0,1,0]]$ گرادیان تصویر سورس را بدست آوردم. (چون تابع filter2d خروجی uint8 میدهد در حالیکه گرادیان می تواند منفی نیز باشد، من از convolve2d در scipy استفاده کردم).

در نهایت یک ماتریس A به ابعاد تعداد پیکسل های مجهول x تعداد پیکسل های مجهول و یک بردار C به طول تعداد پیکسل های مجهول بدست می آورم. این پیاده سازی رابطه زیر برای هر پیکسل مجهول است:

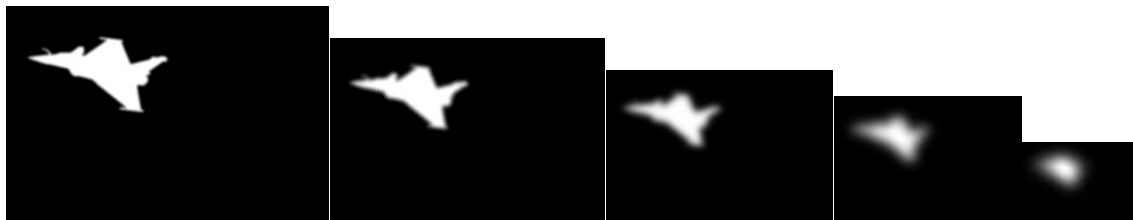
$$f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1} - 4 f_{i,j} = g_{i,j}$$

برای حل این ماتریس من با روش عادی حساب کردن وارون ماتریس و ضرب آن در C با استفاده از توابع numpy موفق به حل آن شدم اما خیلی زمانبر بود. برای همین با توجه به Sparse بودن ماتریس A از طریق تابع spsolve در scipy.sparse.linalg معادلات را حل کردم که از لحاظ زمانی خیلی به صرفه تر هست.

پس از حل معادلات مقادیر پیکسل های مجهول جایگذاری می شود و تصویر نهایی بدست می آید. برای هر کانال تصویر من اینکار را انجام دادم و در نهایت آن را ذخیره کردم.

(۲) در این سوال من از یک هرم ۴ لایه استفاده کردم. در سه لایه بزرگ، لاپلاسین تصاویر را با هم ترکیب میکنم و در بالاترین لایه هرم تصاویر Blur شده را ترکیب می کنم.

برای ترکیب تصاویر من از ماسک هواپیما استفاده کردم به نحوی که در هر لایه این ماسک را همراه با عکس های سورس و هدف کوچک می کنم. در هر لایه ماسک را در یک فیلتر گوسی با ابعاد ۴۱ و واریانس ۳ کانوالو میکنم. چون در لایه های پایین تعداد پیکسل های سفید تغییری نکرده است، پس از کانوالو طیف وسیع تری از نقاط مقادیرشان عوض می شود اما نسبت به لایه های بالاتر مقادیر کمتری دارند. به همین دلیل جهت بیش تر ترکیب شدن تصاویر در لایه های بالا تر من پس از کانوالو با فیلتر گوسی، ماسک حاصل را نرمالایز کردم. به این صورت چهار ماسک زیر به ازای هر لایه بدست آمد:



اما نرمالایز کردن ماسک حاصل باعث شد تا به ازای واریانس های کم یا زیاد نسبت به واریانس ۳ در بسیاری از پیکسل ها به اشباع رنگی بر بخورم. اما در ۳ نتیجه به نظر مطلوب شد. برای ترکیب دو عکس در هر لایه از نسبتی خطی با توجه به ماسک blur شده استفاده می کنم. هرم لاپلاسین و گوسی تصاویر را در یک تابع بازگشتی پیاده سازی کردم. در ابتدا برای کوچک کردن عکس ها من از تابع `resize opencv` استفاده می کردم. اما در توضیحات این تابع، کرنل مورد استفاده جهت blur کردن ذکر نشده بود و نتایج با فرض استفاده از یک فیلتر گوسی جهت این کار، به نتایج بدی ختم می شد. به همین دلیل از تابع `pyrUp` و `pyrDown` از `opencv` که کرنل مورد استفاده را مشخص کرده است استفاده کردم:

$$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

برای بدست آوردن لاپلاسین این تصاویر من این کرنل را در تصویر کانوالو می کنم و از تصویر اصلی کم می کنم و نتیجه ی این کار را در هر لایه برای هر عکس به عنوان لاپلاسین آن تصویر حساب میکنم.