فایل helper.py

همراه تمامی کدها، یک فایل helper.py قرار دارد که به منظور خوانایی کدهاست. تابع هایی که در فایل اصلی کدها هستند (Qi.py(در فایل helper پیاده سازی میشوند تا کد اصلی کوتاه تر و خواناتر باشد. در توضیحات سوال ها هم کدهای اصلی هم تابع های پیاده سازی شده را تا جای ممکن توضیح میدهم.

کل برنامه در یک for تو در تو رخ میدهد که قرار است کل عکس ما رو(ریزالت خواسته شده رو) با استفاده از پچ هایی که از سمپل در هربار اجرای حلقه انتخاب می شود پر کند.
اگر i و j هر دو صفر بودند یعنی میخواهیم خانه بالا سمت چپی رو پر کنیم. برای این موضوع از همان تابع get_random_patch که در تمرین ۱ پیاده سازی شده بود استفاده میکنیم.
اگر i صفر باشد یعنی میخواهیم سطر اول را پر کنیم. یک previous_patch داریم که در واقع قسمتی از پچ قبلی است که میخواهیم به وسیله آن بهترین (شبیه ترین) پچ به آن را پیدا کنیمget_most_similar_patch تابعی است که در فایل helper پیاده سازی شده است و با استفاده ازprevious_patch شبیه ترین پچ برای پر کردن ادامه راه را به ما میده.
سپس باید قسمت میانگین گیری پیاده سازی شود که به این منظور قسمتی که دو پچ اورلپ دارند میانگین گیری میشود و قسمت دیگر هم که جدید هست به همون صورت قرار میگیره.
برای j=0 که پر کردن ستون اول هست هم به همین ترتیب عمل میشه.
برای پر کردن سایر قسمت ها، previous_patch دیگه باسایز ۵۰*۱۰ یا ۱۰*۵۰ نیست و سایز یک پچ رو داره (۵۰*۵۰).
این قسمت به تابع get_most_similar_patch داده میشود و شبیه ترین پچ گرفته شده از تابع با روش میانگین گیری در عکس جایگذاری میشود.
درنهایت هم عکس با اسم im3.jpg ذخیره میشود.
توضیح تابع get_most_similar_patch در فایل helper :
اساس این تابع پیاده سازی یک ssd است ولی به خاطر تفاوت بین ssd برای سطر اول، ستون اول و باقی قسمت ها به سه قسمت تقسیم بندی شده است. پیاده سازی ssd به وضوح مشخص است(کم کردن دو قسمت از هم و به توان ۲

رساندن و سپس جمع کردن درایه های ماتریس حاصل و مقایسه این مقدار به ازای تمام i و j های تکسچر و پیدا کردن بهترین i و j) و خروجی دادن اون قسمت تکسچر به عنوان شبیه ترین پچ.