بسمه تعالی
دانشگاه صنعتی شریف
دانشکده علوم کامپیوتر
Image processing – دکتر مصطفی کمالی
HW3 q2 report

سوال دو – Texture Synthesis :

این سوال از دو فایل تشکیل شده ، فایل اول به نام function.py که توابع مربوطه در آن قرار دارند و فایل کد اصلی به نام q2.py ، در ابتدا توابع موجود در function.py را توضیح میدهیم که 3 تابع در آن موجود است .

Random_patch : این تابع برای اینست که وقتی یک بلوک دیگر از تصویر در حال ساخت را میخواهیم پر کنیم ، نیازه داریم که یک margin از بالای آن بلوک (برای پر کردن ستون اول) یا یک مارجین از کنار آن بلوک (برای پر کردن سطر اول) و یا یک مارجین از بالا و کنار آن به صورت L درنظر گرفته و با استفاده از template match در تصویر اولیه که میخواهیم از آن Synthesis کنیم به دنبال مشابهات بگردیم و پس از یافتن آنها با استفاده از الگوریم DP به تصویر فعلی اضافه کنیم. برای اینکار اولا کناره های پایین و چپ عکس اصلی را در نظر نمیگیریم که بتوانیم مارجین بدست آمده را به اندازه بلوک بزرگ کنیم ، همچنین اگر بخواهیم یک مارجین سطر یا ستون را پیدا کنیم همان را سرچ مکینیم و از normalized cross corr استفاده میکنیم اما اگر بخواهیم که مارجین L شکل را شرچ کنیم ؛ این L شکل را در یک بلوک خالی قرار میدهیم و یک ماسک هم برای آن درست میکنیم که فقط این L شکل را در تصویر جستجو کند و نه کل آن بلوک که بخشی از آن خالی است ، پس از پیدا شدن match ها ؛ آنهایی که خیلی به هم نزدیک هستند را یکی فرض کرده و فقط یکی از آنها را در نظر گرفته و در آرایه good میریزیم ، سپس در 500 match پیدا شده ، آنهایی که مختصات نقطه شان به اندازه کافی با هم فاصله دارد را در good میریزیم و سپس good را shuffle میکنیم و اولین عضو آنرا به عنوان match بدست آمده خروجی میدهیم ، چون که shuffle کرده ایم ، از بین match های پیدا شده عملا یکی را رندم برداشته ایم .

min_cut_path : این تابع برای پیدا کردن یک minimum error cut برای دو margin داده شده است ،اولا که این تابع دو مارجین را به صورت عمودی و سایه سفید شده درنظر میگیرد که میخواهیم از بقل به هم وصل کنیم ، یعنی یک مسیر عمودی میخواهیم ، اگر بخواهیم این کار را افقی انجام دهیم ، قبل از دادن به تابع باید transpose کنیم . پس از گرفتن دو مارجین ابتدا آنها را از هم کم کرده و به توان دو میرساند تا error بدست آید .سپس با استفاده از الگوریتم DP دقیقا به همان صورتی که در کلاس بحث شد مسیر را پیدا میکنیم ، به این صورت که دو ماتریس costو path_matrix به شکل ماتریس error و خالی در نظر میگیریم ، سپس یک ستون در اول و آخر ماتریس cost ,

error با مقدار بی نهایت اضافه میکنیم که برای پیاده سازی مقادیر کناری نیازی به if نباشد و چون دو ستون کناری inf هستند در cost همیشه کمتر میشوند .سپس از سطر آخر ماتریس cost شروع میکنیم ، مقدار آن را برابر سطر آخر error میگذاریم ؛ سپس به سطر بالا تر و درایه اول میرویم ، نگاه میکنیم که در سطر پایین و درایه های مجاور زیر آن کدام کمتر است ؛ هر کدام کمتر بود با error همان درایه ای که رویش قرار داریم از اضافه میکنیم و در cost مینویسیم ؛ همچنین نگاه میکنیم که کدام index از سطر زیری را برداشتیم ، یکی از آن کم میکنیم تا اثر اضافه شدن inf به اول cost کم شود و به آن خانه از path_matrix اضافه میکنیم ، همچنین آخرین ردیف از path_matrix هم صفر است ، دو ماتریس cost و path_matrix را به همین ترتیب پر میکنیم. سپس به سطر اول cost رفته و میبینیم که کدام هزینه کمتری دارد ، به درایه متناظر از path_matrix رفته که نشان میدهد اولین گام cut کجاست ؛ آنرا به اول ماتریس path قرار میدهیم ، سپس به آن گام رفته و در path_matrix نگاه میکنیم که گام بعدی کجاست ، به آنجا رفته و مقدارش را در path اضافه میکنیم ، این کار را تا رسیدن به پایین path_margin انجام میدهیم که حاصل آن cut دلخواه ما با کمینه خطا است .سپس این مسیر را به خروجی میدهیم .

combine_margin : این تابع دو مارجین را میگیرد و با توجه به axis آن که مقدار اولیه آن h است که یعنی دو مارجین در یک سطر بوده اند و لذا horizontal مشود و میخواهیم عمودی آنها را به هم بچشبانیم . اگر بخواهیم که آنها را به صورت افقی کنار هم بچسبانیم این دو مارجین را با transpose میچرخانیم که عمودی شوند و سپس نتیجه نهایی را دوبار با transpose میچرخانیم که انگار افقی کارکرده‌ایم ، سپس آنها را سیاه سفید کرده ، به تابع min_cut_path میدهیم تا مسیر بهینه را به ما بدهد ، پس از آن که مسیر بهینه را داد یک ماتریس خالی به اندازه مارجین در نظر میگریم و سطر اول را تا آن index که در آرایه path وجود دارد را از margin1 و از آن به بعد را از margin2 میگذاریم و تا پایین آن میریم که درنهایت تلفیق دو مارجین بدست امده است و آنرا به خروجی میدهیم .

حال که توابع میانی را توضیح دادیم به فایل q2.py میرویم که یک تابع main دارد که تصویری که میخوهیم آنرا Synthesis کنیم میگیرد ، مقدار تصویر خروجی و همچنین اندازه block_size و همپوشانی آنها را برای پر کردن تصویر نهایی میگیرد و به ما تصویر سنتز شده نهایی و همچنین یک تصویر سفید به اندازه تصویر سنتز شده آخر که تصویر اصلی هم در وسط آن قرار دارد به ما میدهد.

در ابتدا با توجه به مقدار اندازه خروجی و اندازه بلاک ها و همپوشانی ، ممکن است که کامل نتوانیم تصویر را بپوشانیم و کناره راست و پایین آن مقداری خالی بماند ، برای این کار با روابطی که در کد مشخص شده نگاه میکنیم که چند عدد بلوک میتوان قرار داد به اینصورت که مقداری از بلوک آخر یا کاملا fit شود یا مقداری خارج عکس نهایی بیفند ، سپس مقداری که خارج عکس نهایی میفتد را بدست آورده و اندازه عکس نهایی را بزرگ میکنیم که کاملا بلوک آخر fit شود ، سپس در نهایت هم تصویر بدست آمده را به اندازه دلخواهی که در ورودی گفته شده بود crop میکنیم .

در قدم بعدی یک x,y رندم تولید کرده که مختصات گوشه بالاراست اولین patch است که میخواهیم در تصویر خالی قرار دهیم (در بالا چپ آن). این x,y را محدودش طوریست که این patch اولیه تماما در تصویر اصلی ما قرار گیرد .

سپس سطر اول عکس را میخواهیم بسازیم ، یک for با 500 مرحله میسازیم ، وقتی عکس پر شد در مرحله بعدی for به error میخورد که در آنصورت میفهمدد عکس پر شده و از for خارج میشود. سپس در هر مرحله نگاه میکند که اخرین مارجین از سطر اول چیست ، مشابه آن را با تابع random_path بدست آورده و سپس margin جدید بدست آمده را به تابع به combine_margin به margin ی که داشتیم میچسباند ، سپس مقدار path بدست آمده از تابع random_patch را به صورت overlap در جای خود قرار میدهد و در محدوده overlap ، مقدار patch_combined که از تابع combine_margin بدست امده بود را قرار میدهد ، سپس در همین مرحله از for همین کار را برای ستون اول میکند ، در نهایت سطر و ستون اول از تصویر بدست آمده است .

سپس میخواهیم بقیه عکس را پرکنیم برای این کار سطر به سطر جلو میریم و یک مارجین از بالا و از بقل در نظر میگیرم و سپس یک mask هم متناظر آن درست میکنیم ، سپس این pattern , mask را به تابع random_patch میدهیم ، خروجی آن را ، یک باز از بالا و یک بار از بقل آن یک margin در نظر گرفته و با marginی که قبلا در ناحیه overlap بود یک min error cut زده و به هم میچسبانیم ، سپس patch بدست امده را در جای خود قرار داده و سپس margin های combine شده در بالا و چپ را در جای خود قرار میدهیم . با این کار کل شکل پر شده و در نهایت مقداری که اولا به شکل pad کرده بودیم را میبریم و عکس بدست آمده و همینطور عکس اصلی را که وسط یک عکس به اندازه عکس بدست آمده که دورش سفید است در خروجی میدهیم

برای دو texture داده شده و دو texture هم خودمان انتخاب کردیم و با تابع main مقادیر Synthesis شده و اصلی را بدست آورده ، به هم میچسبانیم و ذخیره کردیم .