

نام تصویر	مقدار PSNR به ازای Resizing_Factor برابر با 2							
	روش Bilinear	روش Nearest Neighbor	روش استفاده از فاصله Euclidean	روش استفاده از فاصله City block	روش استفاده از فاصله Chessboard	روش Bicubic تابع	روش من	زمان اجرا برحسب ثانیه
Boat	26.4007	25.5147	26.5192	26.1068	26.5154	26.9338	29.27	1.3
Peppers	29.3113	28.1116	29.4816	29.0077	29.4847	29.7267	32.35	1.5
Cameraman	30.1145	28.0316	29.7834	28.6636	29.7486	30.4992	38.67	1.5
House	28.5407	27.5452	28.6460	28.1419	28.6332	29.2794	32.37	0.3
متوسط PSNR	28.5918	27.300775	28.60755	27.98	28.595475	29.109775	33.17	1.15

روش من: من از روش ترکیبی استفاده کرده ام، ابتدا یک ایده ای که به ذهنم رسید این بود که اگر به عکس پایین نگاهی بیاندازیم:

	1	2	3	4	5
1	0.4980	0	0.4902	0	0.4941
2	0	0	0	0	0
3	0.5020	0	0.5020	0	0.4980
4	0	0	0	0	0
5	0.4941	0	0.5020	0	0.4941
6	0	0	0	0	0
7	0.4980	0	0.4941	0	0.4706
8	0	0	0	0	0
9	0.4941	0	0.4863	0	0.4902

متوجه میشویم که فقط در نقاطی که سطر و ستون زوج دارند 4 همسایه داریم که به نحوی منظم است و لازم است از هر 4 تا استفاده کنیم (بر خلاف سوال اول که واقعا چون گفته شده بود حتما از 4 همسایه استفاده کنید استفاده کرده

ام)، منظور این است که اگر بخواهیم خانه 2 و 1 را بسازیم و از خانه های 1 و 3 و 3 استفاده کنیم آنها خیلی دور هستند و ممکن است خیلی متفاوت با نقطه فعلی باشند و باید مثلا برای خانه های با سطر زوج و ستون فرد از دو همسایه بالایی و پایینی آن مثلا و برای خانه های هم سطر و هم ستون زوج از 4 همسایه و برای خانه های با سطر فرد (و منطقا ستون زوج) از دو همسایه چپ و راست تنها استفاده کنیم و با توجه به مقادیری که در سوال قبل بدست آوردیم، و تست های انجام شده توسط خودم بهترین روش برای حساب کردن سطح روشنایی خانه مد نظر با استفاده از همسایه های ذکر شده، روش روش استفاده از فاصله Euclidean است، این کار افزایش بسیار چشم گیری در کد داشت، همچنین در این متد، من رفتار پیکسل ها حول لبه ها بررسی کردم و متوجه شدم که چطوری یک ضریبی هم همراه با نسبت فاصله اقلیدسی هم در آن ضرب کنم (برای مثال اگر به لبه رسیدیم و از عددی کم به زیاد رفتیم باید عددی که میخواهیم انرا بدست آوریم نزدیک به عدد بزرگتر باشد که شرایط متفاوتی دارد که همه حالات تقریبا در کد ذکر شده است) که متد گفته شده در بالا را میتوانید در فایل **Resize_1** ملاحظه کنید و psnr خیلی خوبی به ما میداد.

اما خب منطقا در لبه ها ممکن است ضعیف عمل کند، راه های خیلی مختلفی را بکار بردم و متوجه شدم که اگر لبه ها بهتر شود، میانگین psnr باز هم خیلی زیاد میشود (برای بعضی عکس ها مثلا Boat کاهش ناچیزی داشت اما برای بقیه عکس ها افزایش چشم گیری داشت) و چون قبلا از تابع interp2 استفاده کرده بودم و دقت خوب انرا در spline و cubic دیده بودم، تصمیم گرفتم با متد خودم انرا ترکیب کنم، برای لبه ها بیشتر از interp2 استفاده کنم (ضرب بزرگتری به آن دادم) نسبت به متد خودم، ولی بقیه جاها از متد خودم بیشتر استفاده کنم و باعث شد که هم دقت میانگین کد خودم و هم دقت میانگین استفاده از تابع interp2 افزایش خوبی پیدا کند. در فانکشن interp2 یکبار از cubic استفاده کردم و یکبار هم از spline و متوجه شدم در صورت استفاده از spline، دقت به طور میانگین افزایش میابد (احتمالا بخاطر حالت smooth بودنش و اینکه برای اشکال پیچیده بهتر عمل میکند). متد دورا میتوانید در **Resize_2** ببینید، روش ترکیبی را نیز در **My_Resize** میتوانید ببینید برای تست هم میتوانید به فایل **Test_My_Resize** مراجعه کنید.

نکته پایانی که میخوام بگویم این بود که در متد 1 اصلا و ابدا روی عکسی اورفیت نشده است، حتی متد 2 همانطور که میدانید برای بعضی عکس ها (و روش خودم) ضعیف تر عمل میکند، اما بطور میانگین و عملکرد خوبی دارد.