

به نام خدا

تمرین چهارم درس مبانی بینایی کامپیوتر

امیررضا حسینی ۹۸۲۰۳۶۳

سوال ۱)

الف) با توجه به توضیحاتی که در مورد آپشن‌های تابع `imfilter` در داک متلب وجود دارد، حالت‌های زیر برای `padding` هنگام اعمال کرنل موجود است:

Padding Options	
numeric scalar, X	Input array values outside the bounds of the array are assigned the value X. When no padding option is specified, the default is 0.
'symmetric'	Input array values outside the bounds of the array are computed by mirror-reflecting the array across the array border.
'replicate'	Input array values outside the bounds of the array are assumed to equal the nearest array border value.
'circular'	Input array values outside the bounds of the array are computed by implicitly assuming the input array is periodic.

حالت اول همان `zero padding` تعمیم یافته به جایگزاری هر عدد دلخواه میباشد. یعنی پیکسل‌های خارج از تصویر اصلی با عدد X جایگزین میشوند.

حالت دوم تحت عنوان `symmetric` و یا `mirror padding` میباشد و به این نحو کار میکند که به نوعی مانند یک آینه، پیکسل‌های همسایه با پیکسل‌های لبه را با پیکسل‌های داخل تصویر به طوری که رفلکس شده آن است، مقداردهی میکند. (قرینه نسبت به محور مرزی افقی یا عمودی)

13	12	11	12	13	14	15	14	13
8	7	6	7	8	9	10	9	8
3	2	1	2	3	4	5	4	3
8	7	6	7	8	9	10	9	8
13	12	11	12	13	14	15	14	13
18	17	16	17	18	19	20	19	18
13	12	11	12	13	14	15	14	13
8	7	6	7	8	9	10	9	8

mirror padding

این نوع `padding` برای زمانی مناسب است که جزییات تصویر در گوشه‌های آن قرار دارند.

گزینه بعدی `padding` به صورت `replicate` میباشد که برای جایگزینی مقدار جدید، نزدیک‌ترین داده از تصویر اصلی را برمیگزیند.

آخرین گزینه نیز `circular` است که به صورت گسترش لبه تصویر با پیکسل‌هایی از لبه مخالف تصویر است که به صورت دورانی به لبه تصویر اضافه می‌شوند. در `padding` دایره‌ای، مقدار هر پیکسل در منطقه جدید، برابر با مقدار پیکسل مقابل در لبه مخالف تصویر قرار می‌گیرد. این تکنیک تضمین می‌کند ویژگی‌های نزدیک به لبه تصویر به همان

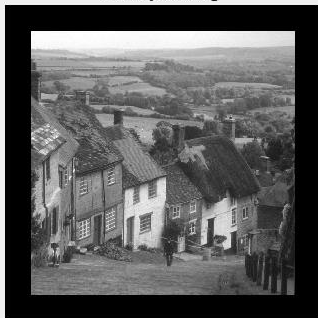
شیوه‌ای که ویژگی‌های در مرکز تصویر هستند، با وزن و اهمیت یکسانی برای سیستم بینایی ماشین به نظر می‌رسند لذا می‌تواند به بهبود دقت الگوریتم‌های پردازش تصویر کمک کند.

Circular padding

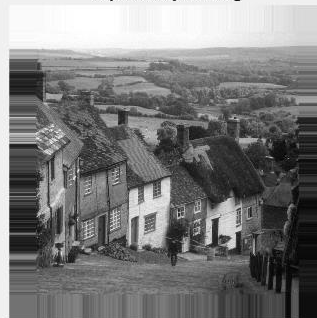
36	31	32	33	34	35	36	31
6	1	2	3	4	5	6	1
12	7	8	9	10	11	12	7
18	13	14	15	16	17	18	13
24	19	20	21	22	23	24	19
30	25	26	27	28	29	30	25
36	31	32	33	34	35	36	31
6	1	2	3	4	5	6	1

خروجی زیر حاصل از اجرای تمامی حالات padding روی تصویر street میباشد که توسط تابع padarray تولید شده و در فایل Q1_a.m قابل دسترس میباشد.

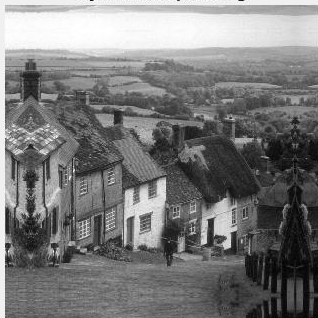
zero padding



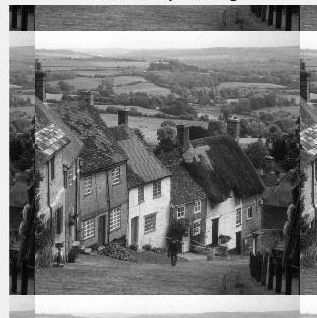
replicate padding



symmetric padding



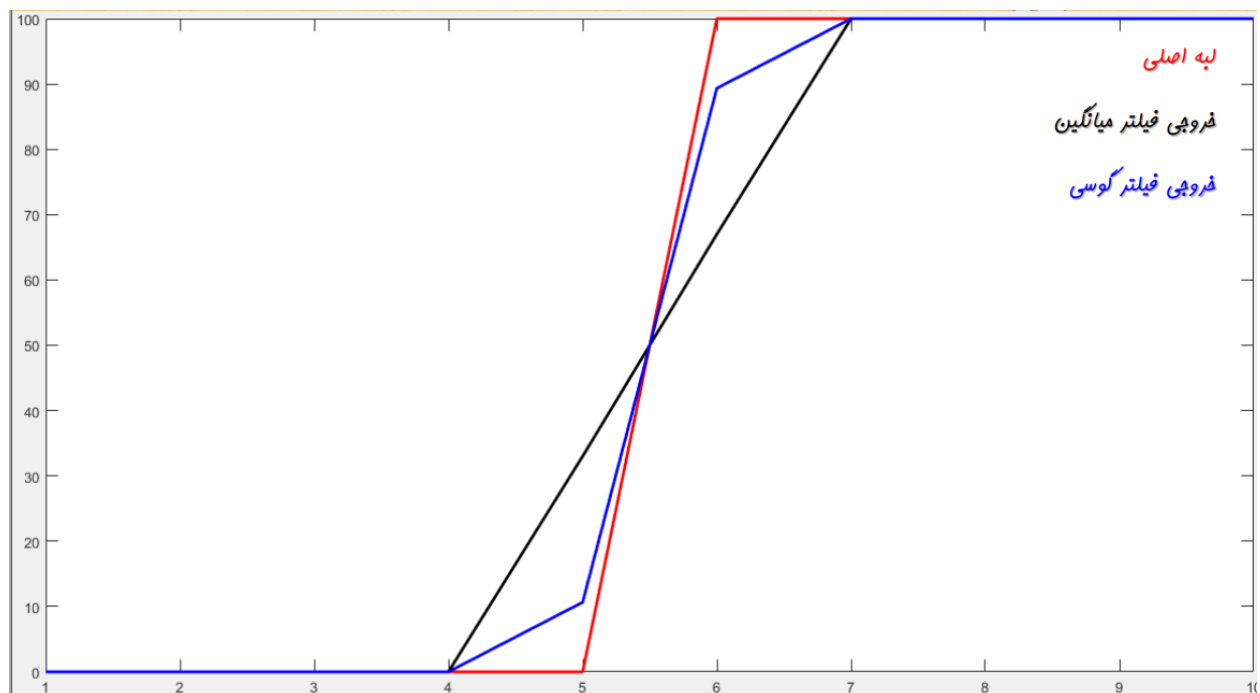
circular padding



اندازه ناحیه padding در مثال بالا برابر است با: 50x50

ب) در حالت کلی فیلتر گوسی نسبت به فیلتر میانگین برای حفظ لبه‌های تصویر در کرنل سایز یکسان، مناسب‌تر است. زیرا در فیلتر میانگین وزن همه درایه‌های زیر کرنل یکسان است ولی در فیلتر گوسی وزن درایه مرکزی بیشتر از بقیه درایه‌ها است و پیکسل‌های مجاور تاثیر کمتری در خروجی تصویر دارند.

حال میتوان دید که اگر همین کرنل روی لبه بیوفتد، وزن بیشتری به پیکسل شامل لبه تصویر و قسمت zero padded میدهد و در نتیجه، آن را بیشتر حفظ میکند. میتوان گفت با اعمال zero padding در واقع در حال اعمال لبه‌هایی تیز روی تصویر جدید بزرگتر هستیم. به عبارت دیگر، فیلتر گوسی یک پیکسل را به پیکسل‌های همسایه کمتر میکشد و سهم کمتری را به آنها میدهد و تا جای ممکن این سهم روی پیکسل مرکزی باقی میماند. این نتیجه در نمودار داخل اسلایدها به وضوح قابل مشاهده است:



همینطور برای اینکه شهود خوبی بتوان از این واقعیت داشت، بنده در تصویر cameraman سطر و ستون‌های زیادی را pad کردم و پس از اعمال کرنل‌های میانگین و گوسی نتیجه به صورت زیر می‌باشد.



همانطور که مشهود است، فیلتر میانگین در smooth تر کردن تمامی لبه‌ها (اعم از لبه‌های گوشه‌ای pad شده) قوی تر عمل کرده است. کد این آزمایش در فایل Q1_b.m موجود می‌باشد.

مستحق طموی

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} + \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \equiv \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$
 Prewitt
 Sobel

۳. محوریت برای معکوس فرضی زیر: محوریت و خواص دایره:

\mathcal{K}_1	\mathcal{K}_2	\mathcal{K}_3
\mathcal{K}_4	\mathcal{K}_5	\mathcal{K}_6
\mathcal{K}_7	\mathcal{K}_8	\mathcal{K}_9

با جاسازی کردن فیلتر Prewitt، عمودی، مقدار جدید پیکسل مرکزی ۲۵ خواهد بود

$$K_{5\text{ new}} = -K_1 - K_2 - K_3 + K_7 + K_8 + K_9 \quad *$$

اصال فیلتر میانی روی K_2, K_5, K_8 (چون فقط این مقادیر را هنگام مستقر نمودن

$$x_{2 \text{ new}} = \frac{x_1 + x_2 + x_3}{3} \quad \text{میانگین داری}$$

$$K_5 \text{ new} = \frac{K_4 + K_5 + K_1}{3}$$

$$K_8 \text{ new} = \frac{K_7 + K_8 + K_9}{3}$$

حال: اعلان فیلتر کشی عمومی، و تقاریر بدست آمده جدید مقدار ۵۵۸ تعداد محاسبات

$$\kappa_{5, \text{new}}^2 = \frac{1}{3}(-\kappa_1 - \kappa_2 - \kappa_3 + \kappa_7 + \kappa_8 + \kappa_9) \quad : \quad \underline{n} \in \mathbb{C} \subset \mathbb{W}_P \subset \mathbb{I}, \mathbb{W} \subset \mathbb{I}^n$$

همانطور که مشاهده می شود رابطه با رابطه * معادل هستند و ضریب $\frac{1}{3}$ است. برکنار می آید.

از برابری دوابع روشتای عمودی به دست آمده. از دو روش نتیجه می گیریم که اعمال توانی فیلترهای یک بعدی هموارکننده (در اینجا میانگین) و بعد مشتق گیری به صورت عمودی معادل با اعمال فیلتر Prewitt در جهت عمودی خواهد بود.

فیلتر گوسی	$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$	+	مشتق افقی	$\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$	=	Sobel افقی	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$
---------------	---	---	-----------	--	---	------------	--

x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9

ماتریس حالت قبلی حل به صورت پارامتری:

اعمال فیلتر Sobel افقی روی تصویر:

$$x_{5new} = -x_1 - 2x_4 - x_7 + x_3 + 2x_6 + x_9$$

حال ابتدا فیلتر گوسی را روی x_4 و x_5 محاسبه می کنیم:

$$x_{4new} = \frac{x_1 + 2x_4 + x_7}{4}$$

$$x_{5new} = \frac{x_2 + 2x_5 + x_8}{4}$$

$$x_{6new} = \frac{x_3 + 2x_6 + x_9}{4}$$

سپس مشتق افقی را اعمال می کنیم:

$$x_{5new}^2 = \frac{1}{4} (-x_1 - 2x_4 - x_7 + x_3 + 2x_6 + x_9)$$

ملاحظه شود که مشتق دوم می شود رابطه افقی با رابطه عمودی است تنها در ضریب مثبت

گرنل گوسی تفاوت وجود دارد. (می توان این ضریب را به مشتک گرنل Sobel

منتقل کرد)

در اینجا هم مانند حالت قبل نتیم بگیریم اعمال توانی فیلترهای یک
بعدی هموارکننده (در اینجا گرس) و به مشتق گیر افقی معادل به اعمال
فیلتر با طول در جهت افقی است.

ب) به دلیل اینکه در مشتق دوم جهت تعبر تابع مورد بحث میباشد که به عبارت دیگر مشتق گرفتن از مشتق تابع میباشد که میتواند هم مثبت هم منفی باشد. پس برای گذار از منفی به مثبت یا مثبت به منفی نیازمند عبور از صفر در لبه تصویر هستیم که این امر با zero crossing مشخص میشود.

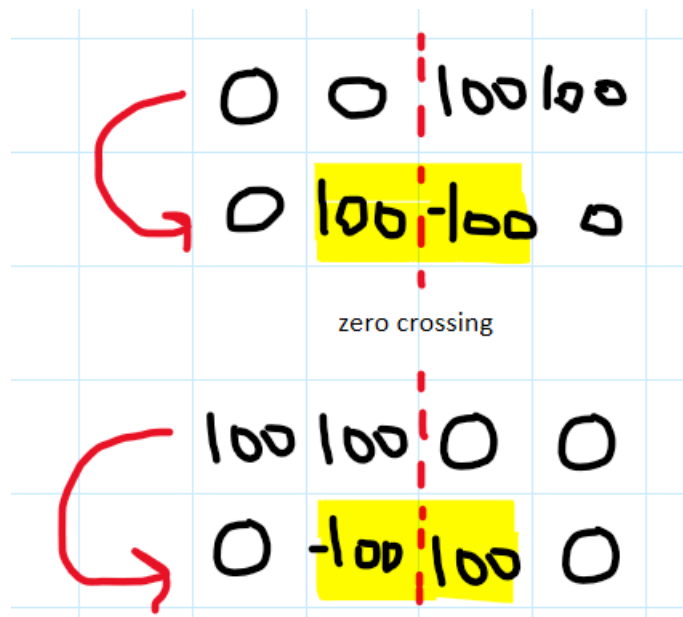
باید اضافه کرد که در مشتق اول به دلیل اینکه مقادیر اخذ شده توسط هر پیکسل همیشه عددی نامنفی (صفر تا ۲۵۵) است، zero crossing رخ نمیدهد.

حالت‌هایی که منجر به رخدادن zero crossing میشود عبارتند از:

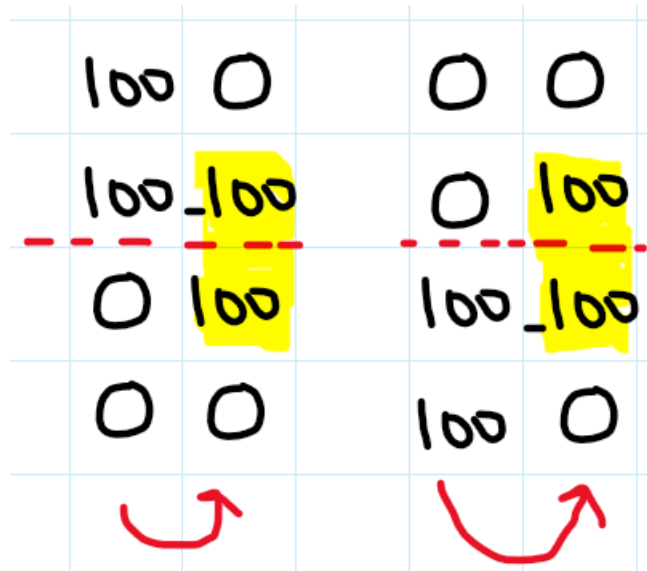
کرنل مشتق مرتبه دوم

1 -2 1

در راستای افقی:



در راستای عمودی:



ج) باتوجه به اینکه مشتق دوم مشتق اول تابع است، به کوچکترین تغییراتی حساس است و آن را نمایش میدهد. و در حالت کلی لبه‌های تیز تری را تولید میکند.

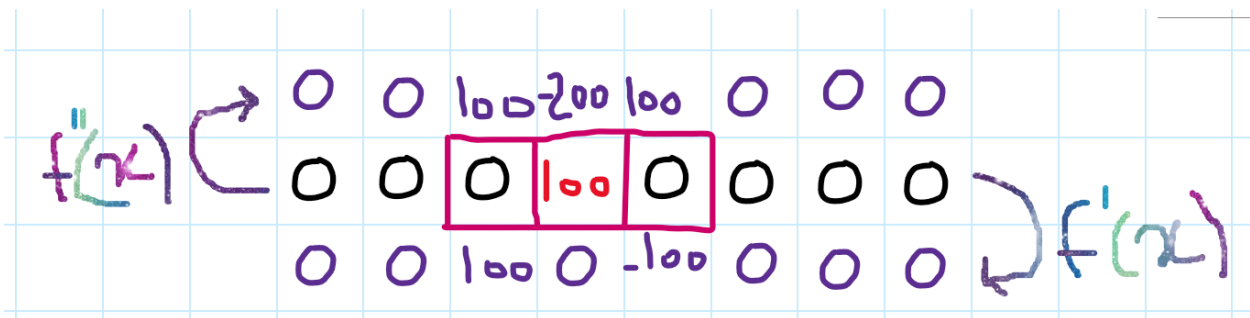
علت وجود نویز بیشتر در آن، وزن زیاد پیکسل مرکزی در زیر کرنل متناظر آن میباشد. مثلاً برای راستای افقی در مشتق مرتبه اول و دوم داریم:

$$f'(x) \approx f(x+1) - f(x-1)$$

$$f''(x) \approx f(x+1) - 2f(x) + f(x-1)$$

واضح است که اگر پیکسل حال حاضر x یک پیکسل نویزی باشد، وزن آن در محاسبه پیکسل جدید در مشتق اول صفر و در مشتق دوم، ۲ در نظر گرفته میشود که همین امر باعث حساسیت بالای این روش نسبت به نویز می‌باشد.

به عنوان نمونه فرض میکنیم پیکسل با سطح روشنایی ۱۰۰ یک پیکسل نویزی باشد. با اعمال کرنل‌های مشتق اول و دوم در راستای افقی خواهیم داشت: (برای مشتق اول از central difference استفاده شده است).



همانطور که واضح است، فیلتر مشتق دوم تغییرات را با شدت بیشتری نشان میدهد لذا میتوان گفت حساسیت بیشتری نسبت به کرنل مشتق اول به نویز وجود دارد.

سوال ۳) همانطوری که مشخص است، در dx تاکید بر روی لبه‌های عمودی و در dy تاکید بر روی لبه‌های افقی می‌باشد.



اما همانگونه که در خروجی تصویر leaves مشهود است، گزینه C نتوانسته به خوبی تمامی لبه‌های تصویر را مشخص نماید اما گزینه آخر به نظر میرسد بهترین سنجش برای تمامی لبه‌های تصویر باشد.



تا به اینجا از نظر شهودی میتوان گفت که گزینه آخر روی لبه‌های قطری به خوبی جواب میدهد. اما برای مقایسه بهتر، آن را با کرنل Sobel که برای پیدا کردن لبه‌های قطری است مقایسه میکنیم.



همانطوری که انتظار میرفت، در پیدا کردن لبه‌های قطری تصویر نیز به خوبی عمل کرده و آن‌ها را پیدا میکند.

سوال ۴)

در این روش، با استفاده از تحلیل لبه‌های تصویر، به دنبال یافتن قسمت مناسب برای هر قطعه پازل هستیم. برای این کار، در ابتدا چهار گوشه‌ی تصویر را فیکس می‌کنیم و سپس با مقایسه شباهت لبه‌ها، قسمت مناسب مربوط به هر قطعه را پیدا می‌کنیم.

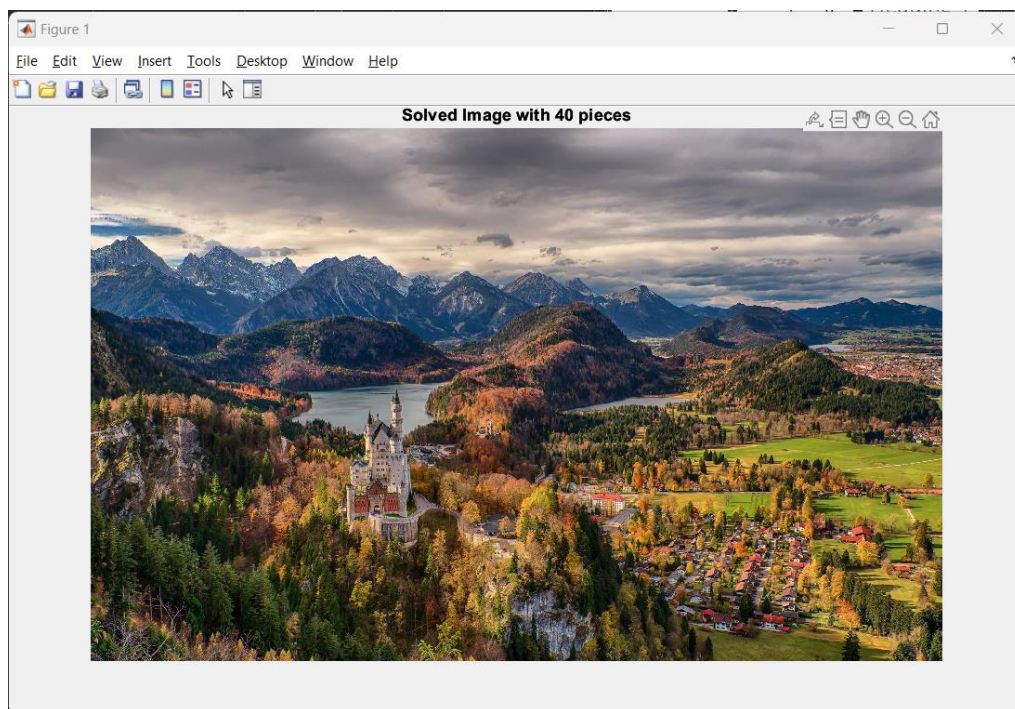
برای این منظور، از بردار ویژگی‌های تصویر خاکستری استفاده می‌کنیم. برای هر قطعه، دو بردار ویژگی تولید می‌شود: بردار ویژگی سمت چپ (بالا) و بردار ویژگی سمت راست (پایین). سپس با مقایسه سطح روشنایی پیکسل‌های لبه‌ی راست (پایین) با پیکسل‌های لبه‌ی چپ (بالا)، میزان شباهت بین دو بردار ویژگی محاسبه می‌شود.

سپس با پیمایش تصویر از گوشه‌ی بالا و چپ شروع کرده، با استفاده از فاصله ی اقلیدسی توان ۲، بین بردار ویژگی قطعات مختلف با تصویر خاکستری قطعه‌ی فیکس شده، مقداری را حساب می‌کنیم. سپس کوچکترین این مقادیر را به عنوان قسمت مناسب مربوط به همان موقعیت در نظر می‌گیریم، که بیشترین شباهت را با لبه‌ی تصویر فیکس شده دارد.

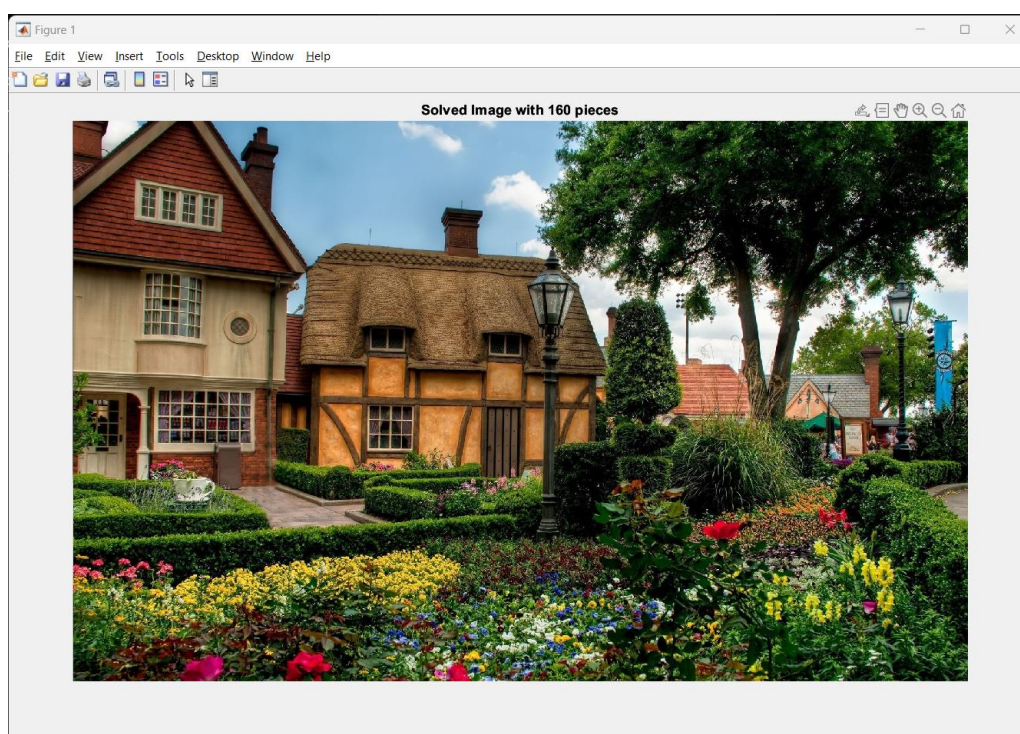
با استفاده از این روش، می‌توانیم قطعات پازل را به صورت دقیق و سریع با هم ترکیب کنیم و پازل را به شکل کامل حل کنیم

برای محاسبه همین امر برای تصاویر چرخانده شده، باید از ویژگی LBP استفاده کرد که مستقل از چرخش و دوران است. به دلیل کمبود زمان حل این سوال، به همین روش برای تصاویر چرخانده نشده اکتفا می‌کنیم.

نتیجه اجرا روی تصویر پازل اول-۴۰ تیکه



نتیجه اجرا روی تصویر دوم - ۱۶۰ تکه



نتیجه اجرا روی تصویر سوم - ۴۰ تکه

