

توضیح تابع :dilation

در این قسمت از تابع `convole2d` استفاده کرده ایم که عملیات convolution را بین تصویر و عنصر ساختاری انجام داده. تصویر خروجی هم هم اندازه با ورودی است. دقت کنید که فرض کرده ایم تصویر ورودی دوستخی و 255 یا صفر است. پس اگر حاصل بیشتر از 0 باشد یعنی اشتراکی با عنصر ساختاری وجود داشته و آن را 255 میکنیم.

توضیح تابع :morphological_connected_component

این تابع را به کمک ایده گفته شده پیاده سازی کرده ایم. به این شکل که در ابتدای یک پیکسل از کامپوننت را در نظر گرفته و آن را با عنصر ساختاری dilate کرده و سپس اشتراک آن با تصویر را حساب کرده. اگر عنظر ساختاری 3×3 باشد و همه خانه ها یک، باعث میشود تمام همسایه های 8 تایی پیکسل در شرایطی که 255 هستند به شکل اضافه شوند. اگر عنصر ساختاری را حالت علامت جمع تعریف کنیم، همسایه های 4 تایی را در نظر گرفته. این کار را تا جایی انجام میدهیم که پیکسل جدیدی به شکل اضافه نشده یا $X_{k+1} = X_k$ شود. این قسمت در bfs انجام شده. کد به این شکل است که پیکسل هایی که میتوانند باعث تشکیل یک کامپوننت شوند به عنوان نقطه شروع به تابع bfs داده شده تا کامپوننت را تشکیل دهند. همانطور که میبینید در خروجی، کامپوننت ها به دست آمدند.

توضیح :erosion

در این قسمت از تابع `convole2d` استفاده کرده ایم اما چون در erosion ما چرخش نداریم، ابتدا آن را 180 درجه چرخانده تا تاثیری در نتیجه نداشته باشد.

توضیح :morphological_skeleton

در اینجا تا زمانی که هنوز پیکسلی روشن وجود داشته باشد، عملیات Sk را انجام داده. همانطور که در خروجی میبینید تصویر به دست آمده درست است.

