

دانشکده مهندسی کامپیوتر و فن آوری اطلاعات
دانشگاه صنعتی امیرکبیر

گزارش تمرین دوم درس مدل‌های احتمالاتی گراف‌ی

استاد درس:

دکتر نیک‌آبادی

نام دانشجو:

احمد اسدی

۹۴۱۳۱۰۹۱

فروردین ۱۳۹۵

فهرست مطالب

۱	تقطیع با دسته‌بندی‌کننده بیز ساده	۱
۱	۱.۱ دسته‌بندی‌کننده بیز ساده به عنوان عملگر تقطیع	۱
۱	۲.۱ تاثیر نویزهای مختلف بر عملکرد دسته‌بندی‌کننده بیز ساده	۱
۲	۲ تقطیع با استفاده از مدل میدان تصادفی مارکف	۲
۲	۱.۲ بررسی مدل	۲
۳	۲.۲ تاثیر اهمیت یکپارچگی بر چسب‌ها نسبت به سطح خاکستری	۳
۳	۳.۲ نحوه تغییر دما	۳
۴	۴.۲ مقداردهی اولیه برخی پیکسل‌ها	۴
۴	۵.۲ تاثیر نوع همسایگی	۴
۴	۶.۲ بررسی تاثیر نویز	۴
۴	۷.۲ تقطیع داده واقعی در فضاهاى رنگى مختلف	۴
۵	۸.۲ تقطیع تصویر با بیش از یک ویژگی	۵
۵	۳ توضیحات	۵

۱ تقطیع با دسته‌بندی‌کننده بیز ساده

در این قسمت، ابتدا به بررسی دسته‌بندی‌کننده بیز به عنوان عملگر تقطیع پرداخته و سپس تاثیر نویزها مختلف بر نتیجه این دسته‌بندی‌کننده را بررسی می‌نماییم.

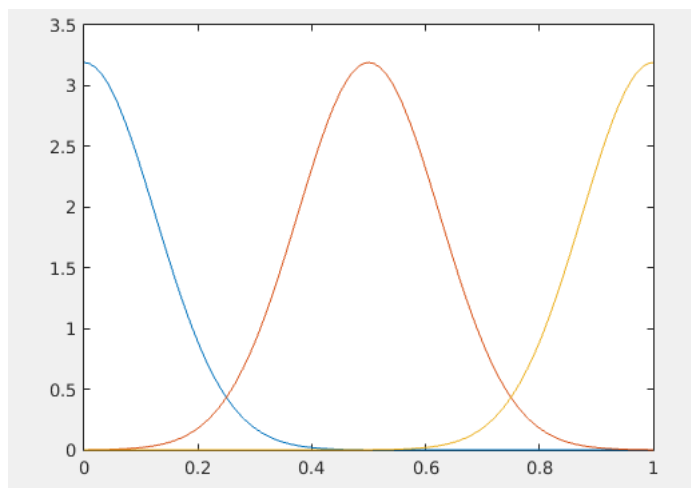
۱.۱ دسته‌بندی‌کننده بیز ساده به عنوان عملگر تقطیع

در این بخش قصد داریم، با استفاده از دسته‌بندی‌کننده بیز ساده، تصویر ورودی را به سه قطعه مختلف تقسیم نماییم. برای این منظور، تنها از مقدار سطح خاکستری نقاط تصویر به عنوان ویژگی استفاده می‌نماییم. رابطه ^۱، به عنوان رابطه پایه این دسته‌بندی‌کننده مورد استفاده قرار می‌گیرد. در این بخش ابتدا با استفاده از روش MAP ^۱، کلاس مربوط به هر پیکسل را تخمین می‌زنیم.

$$P(\Theta|X) = \frac{P(\Theta, X)}{P(\Theta)} \quad (۱)$$

به عنوان اولین گام در این فرایند، باید مدل بیز ساده را جهت تقطیع تصویر به سه قطعه مختلف، آموزش دهیم. برای آموزش این مدل، کفایت تعداد پیکسل‌ها از هر سطح خاکستری مختلف را شمارش کنیم. با توجه به این که تمام پیکسل‌های تصویر آموزشی ما فقط دارای سه سطح خاکستری مختلف هستند، هر سطح خاکستری دقیقاً بیانگر یک کلاس است. با این توضیح، نسبت تعداد پیکسل‌های هر سطح خاکستری به کل پیکسل‌ها برابر با $P(\Theta)$ است. برای مدل کردن احتمال توام $P(\Theta, X)$ از یک توزیع گاوسی استفاده می‌نماییم که میانگین آن برابر با سطح خاکستری کلاس مورد نظر باشد و واریانس هر سه توزیع توام (یک توزیع به ازای هر کلاس) را با هم برابر و برابر با ۰.۱۲۵ در نظر می‌گیریم. شکل ^۱ توزیع احتمالی توام اولیه را که در این بخش در نظر گرفته شده است، نمایش می‌دهد. با اعمال شرایط فوق به مدل، می‌توان مدل را طوری آموزش داد که برای تقطیع تصویر به سه سطح خاکستری مناسب باشد. این مدل

شکل ۱: توزیع احتمالی توام کلاس‌های مختلف



روی داده‌های بدون نویز به طور دقیق عمل می‌نماید. اما در مواردی که تصویر شامل نویز باشد، بسته به قدرت نویز در هر پیکسل می‌تواند دچار خطا شود. با توجه به نمودار ^۱، پیکسل‌هایی که در اثر نویز، تغییر زیادی در سطح خاکستری آن‌ها ایجاد شده باشد، اشتباه تشخیص داده خواهند شد.

۲.۱ تاثیر نویزهای مختلف بر عملکرد دسته‌بندی‌کننده بیز ساده

شکل ^۲ نشان‌دهنده نتیجه تقطیع تصاویر با استفاده از دسته‌بندی‌کننده بیز ساده است. همان‌طور که مشاهده می‌شود، در صورتی که قدرت نویز اعمال شده، خیلی زیاد نباشد، دسته‌بندی‌کننده قادر به دسته‌بندی صحیح تصویر است. اما در صورتی که نویز ورودی، قدرت زیادی داشته باشد، این دسته‌بندی‌کننده کمک

^۱ Maximum A Posteriori

زیادی به تقطیع تصویر نمی‌کند. در شکل ۲ به هیستوگرام تصاویر نویزی دقت نمایید. تا جاییکه شکل این هیستوگرام شبیه به شکل ۱ باشد، دسته‌بندی‌کننده به خوبی عمل می‌کند. در غیر این صورت، عمل تقطیع با مشکل روبرو می‌شود.

۲ تقطیع با استفاده از مدل میدان تصادفی مارکف

برای بهبود عملکرد تقطیع، می‌توان از مدل میدان تصادفی مارکف استفاده کرد. در این مدل علاوه بر فاکتور میزان سطح خاکستری نقاط، فاکتور دیگری تحت عنوان سازگاری با نقاط همسایه، تعریف نموده و از آن در فرایند تقطیع استفاده می‌نماییم. ایده اصلی در این بخش این است که معمولاً برچسب هر پیکسل، با برچسب همسایگان آن پیکسل، سازگار است. با در نظر گرفتن این فاکتور، از ایجاد نقاط یکه در تصویر که برچسب آن‌ها با برچسب همه یا بیشتر همسایگانش مغایر باشد جلوگیری می‌کنیم.

در ادامه، ابتدا به بیان مدل پرداخته و روش اجرای الگوریتم را توضیح خواهیم داد. سپس به بررسی پارامترهای مدل از جمله میزان اهمیت یکپارچگی برچسب‌ها نسبت به سطح خاکستری پیکسل‌ها، نحوه تغییر دما، میزان نویز موجود در تصویر، نوع همسایگی و مقداردهی دستی اولیه برخی پیکسل‌ها را مورد بررسی قرار خواهیم داد.

۱.۲ بررسی مدل

اگر فاکتور اول را برابر با توزیع احتمال توام برچسب هر نقطه و سطح خاکستری آن مطابق با رابطه ۲ در نظر بگیریم و فاکتور دوم را مطابق با رابطه ۳ تعریف کنیم، رابطه ۴ بیان‌کننده مدل مورد استفاده در این بخش خواهد بود.

$$\Phi_i(X_i, \Theta) = PDF(X_i; \mu_\Theta, \sigma_\Theta) \quad (2)$$

$$\Phi_{ij}(X_i, X_j) = \begin{cases} 1 & \Theta_{X_i} = \Theta_{X_j} \\ -1 & \Theta_{X_i} \neq \Theta_{X_j} \end{cases} \quad (3)$$

$$P(X_1, X_2, \dots, X_n, \Theta) = \frac{1}{Z} (\prod_{i=1}^n \prod_{j \in Neighbours(X_i)} \Phi_{ij}) \cdot \prod_{i=1}^n \Phi_i \quad (4)$$

برای تشخیص برچسب هر پیکسل در این حالت، باید Θ را طوری پیدا کنیم که رابطه ۴ بیشینه شود. برای بدست آوردن این مقدار، تابع آرا را به شکل رابطه ۵ تعریف می‌کنیم و با استفاده از روش شبیه‌سازی تابکاری^۱ پارامتر بهینه Θ^* را محاسبه می‌نماییم.

$$V = \sum_{i=1}^n ((1 - \beta) \Phi_i(X_i, \Theta) + \sum_{j \in Neighbours(X_i)} \beta \Phi_{ij}(X_i, X_j)) \quad (5)$$

با توجه به این که محاسبه تابع آرا برای کل تصویر زمان‌بر است، به جای محاسبه مقدار کل تابع آرا در هر مرحله از الگوریتم شبیه‌سازی تابکاری، مستقیماً میزان اختلاف ایجاد شده در تابع آرا به ازای تغییر برچسب پیکسل r ام را محاسبه می‌کنیم. رابطه ۶ میزان اختلاف تابع آرا به ازای تغییر برچسب پیکسل r ام، محاسبه می‌نماید. با اعمال این روش، سرعت اجرای الگوریتم به صورت چشم‌گیری افزایش یافت.

$$\Delta V = (1 - \beta) (\Phi_r(X_r, \Theta_r^{new}) - \Phi_r(X_r, \Theta_r^{old})) + \beta \sum_{j \in Neighbours(X_r)} (\Phi_{rj}^{new}(X_r, X_j) - \Phi_{rj}^{old}(X_r, X_j)) \quad (6)$$

در پیاده‌سازی الگوریتم شبیه‌سازی تابکاری، از پارامتر دیگری تحت عنوان نرخ تغییر در تکرار، علاوه بر پارامترهای ذکر شده استفاده شده است. این پارامتر مشخص می‌کند در هر تکرار الگوریتم چه نرخی از تصویر را باید تغییر دهیم. تغییرات به شکل نقطه به نقطه اعمال می‌شوند و تصمیم‌گیری برای پذیرش

^۱ Simulated Annealing

تغییرات بعد از اعمال هر تغییر صورت می‌گیرد. پس از اعمال تمام تغییرات مجاز در یک تکرار، تمامی پارامترها از جمله نرخ تغییر در تکرار به شکل خطی تغییر می‌کنند. رابطه ۷ نحوه کاهش نرخ تغییرات در هر تکرار را نمایش می‌دهد.

$$\xi = \xi_{init} + \alpha k \quad (۷)$$

شکل ۳ خروجی مدل مارکف را با خروجی مدل بیز ساده برای یک ورودی نویزی نمایش می‌دهد. همان‌طور که ملاحظه می‌شود، عملکرد مدل میدان تصادفی مارکف در مقایسه با نويز بسیار بهتر از عملکرد مدل بیز ساده است. همان‌طور که مشاهده می‌شود، در خروجی مدل میدان تصادفی مارکف خطاهایی وجود دارد که با ادامه دادن الگوریتم این خطاها هم از بین خواهند رفت اما به دلیل کم بودن دما در این مراحل، تعداد تکرار برای حذف این خطاها زیاد است. بنابراین، الگوریتم را در مرحله دهم متوقف کردیم.

۲.۲ تاثیر اهمیت یکپارچگی بر چسب‌ها نسبت به سطح خاکستری

در رابطه ۶، پارامتر β تنظیم‌کننده میزان اهمیت فاکتور یکپارچگی بر چسب‌ها نسبت به فاکتور سطح خاکستری است. هرچه این مقدار بزرگتر باشد، تاثیر این فاکتور از فاکتور سطح خاکستری بیشتر می‌شود. اگر مقدار این پارامتر به صفر برسد، مدل ارائه شده تبدیل به مدل بیز ساده که در قسمت قبل معرفی شد، می‌شود.

روش‌های مختلفی برای پیدا کردن مقدار بهینه برای این پارامتر وجود دارد. ما در این پروژه، مقدار این پارامتر را به طور خطی مطابق با رابطه ۸ تغییر می‌دهیم. در این رابطه، α شیب خط افزایشی، k شماره مرحله و β_{init} مقدار اولیه β است.

$$\beta = \beta_{init} + \alpha k \quad (۸)$$

مقدار اولیه β_{init} را صفر قرار می‌دهیم تا در مرحله اول، مدل شبیه مدل بیز عمل کند. به این طریق، الگوریتم به سرعت به حالتی نزدیک به حالت بهینه نزدیک می‌شود. اعمال این روش، تاثیر بسزایی در افزایش سرعت همگرایی الگوریتم ایجاد کرد. به این طریق در اولین تکرار، خروجی الگوریتم شبیه خروجی دسته‌بندی‌کننده بیز ساده می‌شود. با تکرار مراحل بعدی، گام به گام سعی در حذف نويز در تصویر نهایی می‌کنیم. با افزایش گام به گام این پارامتر، اهمیت بر چسب پیکسل‌های همسایه، بیشتر می‌شود و پیکسل‌هایی که به طور یکه در یک همسایگی دارای بر چسبی متفاوت نسبت به همسایگان خود هستند، تصحیح می‌شوند. این مراحل به بهبود عملکرد مدل کمک زیادی می‌کنند.

شکل ۴ تاثیر پارامتر β را بر عملکرد الگوریتم نمایش می‌دهد. نويز تصویر در همه مراحل این آزمایش به شکل گاوسی و با $\sigma = ۰/۱$ اعمال شده است. همان‌طور که در شکل مشاهده می‌شود، مقادیر پایین برای پارامتر β قدرت الگوریتم را در رفع نويز، کاهش می‌دهد. از طرفی، افزایش بیش از حد مقدار این پارامتر هم، با غلبه بر مقدار فاکتور اول (سطح خاکستری نقطه) منجر به افزایش خطا و کاهش کارایی الگوریتم می‌شود. بهترین روش ممکن برای کنترل الگوریتم و بالا بردن سرعت، افزایش تدریجی این پارامتر به صورت خطی است.

۳.۲ نحوه تغییر دما

در این قسمت، دو مرحله تغییر دمای مختلف را مورد بررسی قرار می‌دهیم. در روش اول، دمای سیستم را به طور خطی کاهش می‌دهیم تا به یک حد آستانه‌ای برسد. در روش دوم، دما را مطابق با یک نمودار نمایی تغییر می‌دهیم. در روش اول، دما مطابق با رابطه ۹ و در روش دوم، مطابق با رابطه ۱۰ تغییر می‌کند. در هر دو رابطه، k شماره مرحله و $\alpha > ۰$ و $\gamma > ۰$ مقادیر ثابت هستند.

$$T = T_{init} + \alpha k \quad (۹)$$

$$T = \frac{1}{\sqrt{2\pi}} \cdot e \cdot (-0.5k^2) + 10 \quad (10)$$

در رابطه ۹، دما به طور یکنواخت و خطی، کاهش می‌یابد. زمانی که دما زیاد است، به الگوریتم بیشتر اجازه می‌دهیم تا پارامتر غیر بهینه را انتخاب نماید و رفتار تصادفی الگوریتم شدید است. اما به مرور زمان و به طور یکنواخت، این اجازه کمتر می‌شود و در مراحل انتهایی الگوریتم، رفتار تصادفی به حداقل ممکن خود می‌رسد و الگوریتم، فقط در جهت بهبود نتیجه حرکت می‌کند.

در رابطه ۱۰، دما به طور نمایی کاهش می‌یابد به طوری که در تعداد زیادی از مراحل، دما زیاد است و رفتار تصادفی الگوریتم شدید است. در اواخر الگوریتم، دما با شیب زیادی کاهش می‌یابد و الگوریتم زمان کافی برای بهبود پاسخ نخواهد داشت.

شکل ۵ نتایج الگوریتم به ازای اجرای هر دو نحوه تغییر دما را نمایش می‌دهد. همان‌طور که انتظار می‌رود، بهترین عملکرد با تغییر دمای خطی بدست آمده است.

۴.۲ مقداردهی اولیه برخی پیکسل‌ها

در صورتی که برخی از پیکسل‌ها به طور دستی در ابتدای الگوریتم مقداردهی شوند و در فرایند اجرای الگوریتم، هیچ کدام از این پیکسل‌ها اجازه تغییر نداشته باشند، انتظار می‌رود نتیجه نهایی الگوریتم به طور قابل ملاحظه‌ای بهبود یابد. زیرا با داشتن چنین اطلاعاتی و با اعمال فاکتور یکپارچگی برچسب‌ها، نویز موجود در نقاط همسایه این پیکسل‌ها به طور کامل حذف شده و برچسب این پیکسل‌ها در تشخیص برچسب پیکسل‌های دیگر موثر خواهد بود. شکل ۶ تاثیر مقداردهی اولیه را بر عملکرد الگوریتم نمایش می‌دهد.

۵.۲ تاثیر نوع همسایگی

در صورتی که همسایگی نقاط را به شکل ۴ تایی در نظر بگیریم، برچسب هر پیکسل فقط با همسایگان عمودی و افقی خود مقایسه می‌شود. این مساله می‌تواند باعث کاهش دقت الگوریتم و باقی ماندن نویزهای مورب در تصویر شود. اگر همسایگی نقاط را به شکل ۸ تایی در نظر بگیریم، از آنجا که تعداد همسایگانی که برچسب هر پیکسل را با آن‌ها مقایسه می‌کنیم افزایش می‌یابد، دقت الگوریتم افزایش خواهد یافت. با در نظر داشتن بهبودی که در بخش «بررسی مدل» ارائه شد (محاسبه تغییرات تابع آرا به جای مقدار تابع در کل تصویر)، افزایش تعداد همسایگی‌های مورد بررسی، تاثیری در زمان اجرای الگوریتم نخواهد داشت. شکل ۷ تاثیر نوع همسایگی را بر عملکرد الگوریتم نمایش می‌دهد.

۶.۲ بررسی تاثیر نویز

در دسته‌بندی‌کننده بیز ساده دیدیم افزایش قدرت نویز در تصویر می‌تواند باعث کاهش چشم‌گیر دقت فرایند تقطیع شود. در مدل میدان تصادفی مارکف، با اضافه کردن فاکتور یکپارچگی برچسب نقاط همسایه، تا حد خوبی این مشکل حل شد. اما در این مدل، اگر قدرت نویز زیاد باشد، تعداد تکرار بسیار بیشتری باید انجام شود تا به پاسخ بهینه دست پیدا کنیم. با این توضیح، انتظار داریم با افزایش قدرت نویز تصویر، تعداد مراحل مورد نیاز برای رسیدن به پاسخ نهایی افزایش یابد.

شکل ۸ تاثیر نویز را در عملکرد الگوریتم به نمایش گذاشته است. همان‌طور که مشاهده می‌شود، افزایش نویز تصویر، دقت تقطیع را کاهش می‌دهد.

۷.۲ تقطیع داده واقعی در فضاهای رنگی مختلف

در این بخش از دو فضای رنگی مختلف برای تقطیع تصویر استفاده می‌نماییم. تصویر سطح خاکستری را مانند مثال‌های قبلی تقطیع می‌کنیم. برای پیدا کردن پارامترهای توزیع گاوسی برچسب‌ها در تصویر سطح خاکستری، ابتدا هیستوگرام تصویر بدون نویز را بدست می‌آوریم و سپس قله‌های هیستوگرام را به عنوان μ_i و نصف عرض هر قله را برابر با σ_i قرار می‌دهیم. با بدست آوردن این اطلاعات، همانند مثال‌های قبلی، عمل تقطیع را انجام می‌دهیم.

برای بررسی تاثیر فضاهای رنگی دیگر، فضای HSV را به عنوان نمونه مورد بررسی قرار می‌دهیم. در این قسمت، ابتدا تصویر را از فضای رنگی به فضای HSV

منتقل کرده و سپس فاکتورهای SV را فیلتر می‌کنیم. سپس تمام مراحل قبلی را روی تصویر بدست‌آمده اجرا می‌کنیم.

شکل ۹ تاثیر فضاهای رنگی در دقت تقطیع را نمایش می‌دهد. در این شکل، خروجی الگوریتم به ازای دو نویز مختلف با قدرت‌های مختلف به نمایش گذاشته شده است. همان‌طور که مشاهده می‌شود، تقطیع در فضای رنگی HSV دقت بیشتری دارد.

۸.۲ تقطیع تصویر با بیش از یک ویژگی

در این قسمت برای استفاده از بیش از یک ویژگی در مدل، از هر دو ویژگی سطح خاکستری و پارامتر رنگی H که در بخش قبلی مورد استفاده قرار گرفت، در تقطیع استفاده می‌کنیم. پارامترهای اولیه هر دو این ویژگی‌ها شامل μ و σ قبلاً محاسبه شده‌اند. فاکتور H را به دو فاکتور موجود قبلی اضافه می‌نماییم و با بازنویسی رابطه ۶ به بررسی نقش بیش از یک ویژگی در تصویر می‌پردازیم.

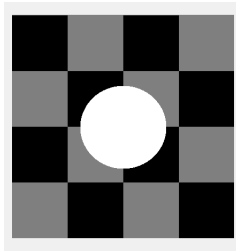
شکل ۱۰ خروجی الگوریتم را در حالت تک ویژگی با استفاده از سطح خاکستری و در حالت دو ویژگی نمایش می‌دهد. همان‌طور که مشاهده می‌شود، خروجی الگوریتم در حالت دو ویژگی بهتر از حالت یک ویژگی است.

۳ توضیحات

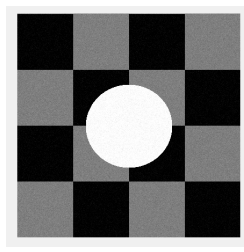
* با توجه به زمان‌بر بودن اجرای کامل الگوریتم، عموم آزمایشات نتیجه تکرار چهارم و پنجم الگوریتم هستند. بدیهی است ادامه اجرای الگوریتم بر بهبود

پاسخ موثر خواهد بود اما نتایج مقایسات تغییری نخواهند کرد.

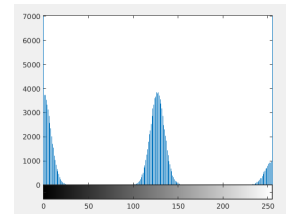
* سورتس کد مربوط به پروژه در ضمیمه این گزارش ارسال شده است. همین‌طور این کد از [این لینک](#)، قابل دریافت می‌باشد.



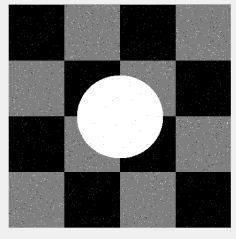
(ج) نتیجه تقطیع تصویر نویزی با $\sigma = 0.001$



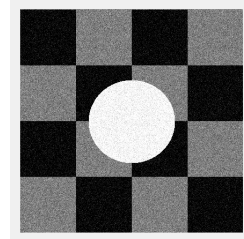
(ب) تصویر نویزی با $\sigma = 0.001$



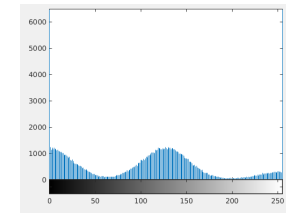
(آ) هیستوگرام تصویر نویزی با $\sigma = 0.001$



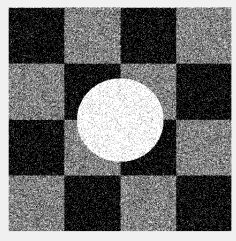
(و) نتیجه تقطیع تصویر نویزی با $\sigma = 0.01$



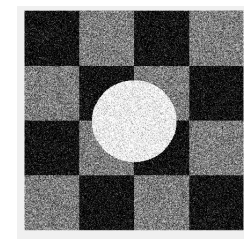
(ه) تصویر نویزی با $\sigma = 0.01$



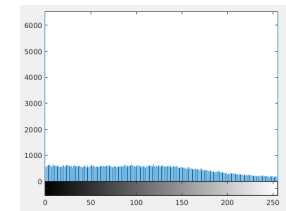
(د) هیستوگرام تصویر نویزی با $\sigma = 0.01$



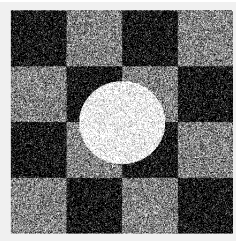
(ط) نتیجه تقطیع تصویر نویزی با $\sigma = 0.05$



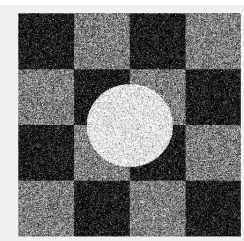
(ح) تصویر نویزی با $\sigma = 0.05$



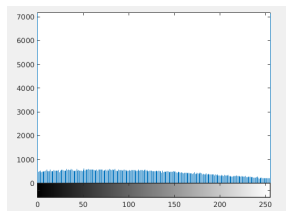
(ز) هیستوگرام تصویر نویزی با $\sigma = 0.05$



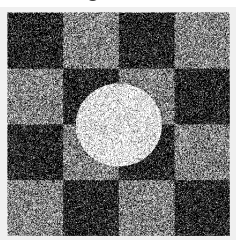
(یب) نتیجه تقطیع تصویر نویزی با $\sigma = 0.1$



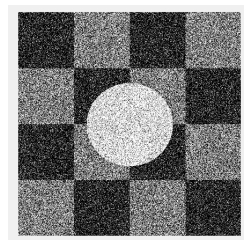
(یا) تصویر نویزی با $\sigma = 0.1$



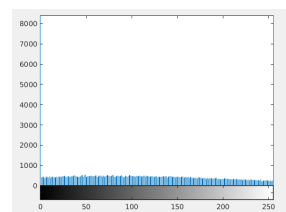
(ی) هیستوگرام تصویر نویزی با $\sigma = 0.1$



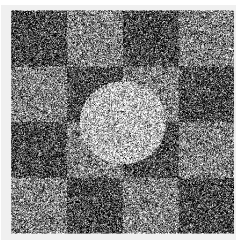
(یپ) نتیجه تقطیع تصویر نویزی با $\sigma = 0.2$



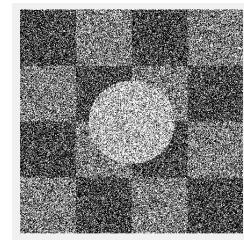
(ید) تصویر نویزی با $\sigma = 0.2$



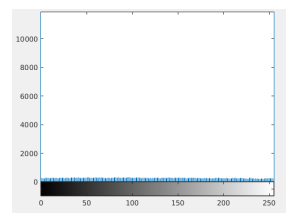
(یج) هیستوگرام تصویر نویزی با $\sigma = 0.2$



(یح) نتیجه تقطیع تصویر نویزی با $\sigma = 0.8$

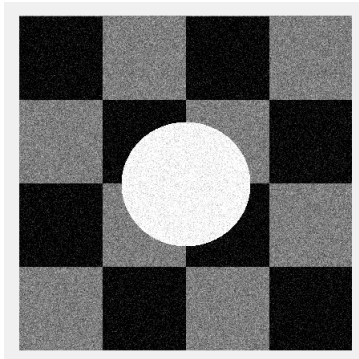


(یز) تصویر نویزی با $\sigma = 0.8$

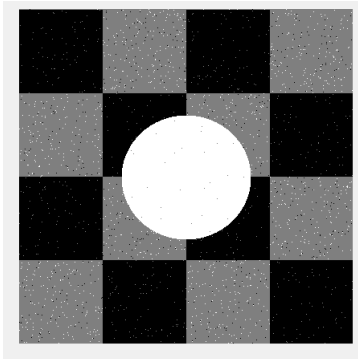


(یو) هیستوگرام تصویر نویزی با $\sigma = 0.8$

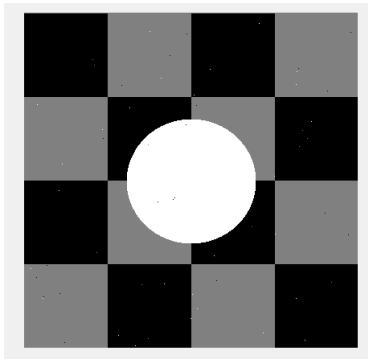
شکل ۲: نتایج عملکرد دسته‌بندی‌کننده بیز ساده در تقطیع تصاویر. در تمام تصاویر $\mu_{noise} = 0$



(ج) تصویر ورودی با نویز گاوسی و $\sigma = 0.01$

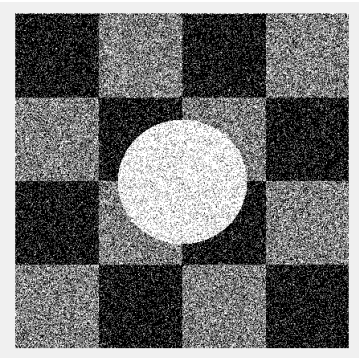


(ب) خروجی مدل بیس ساده

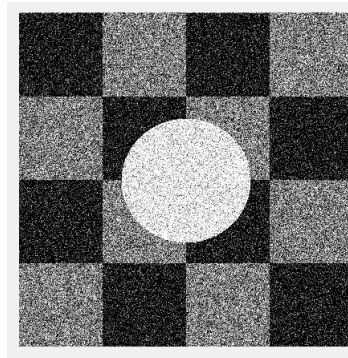


(آ) خروجی مدل مارکف در تکرار ۱۰

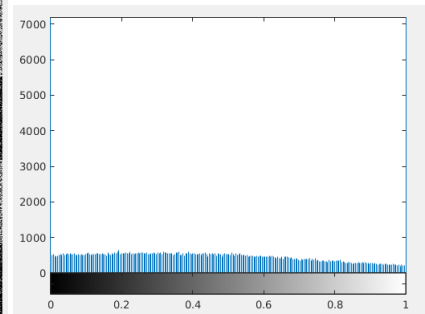
شکل ۳: مقایسه خروجی مدل های مارکف و بیس ساده



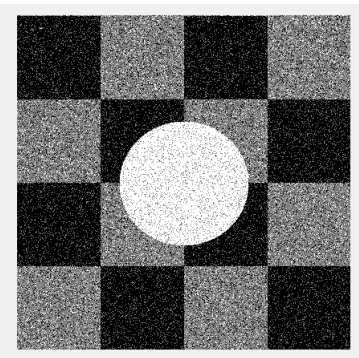
(ج) خروجی الگوریتم با $\beta = 0.05$



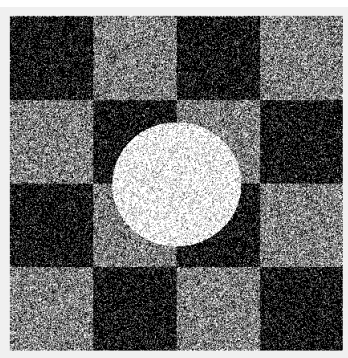
(ب) تصویر نویزی ورودی با $\sigma = 0.1$



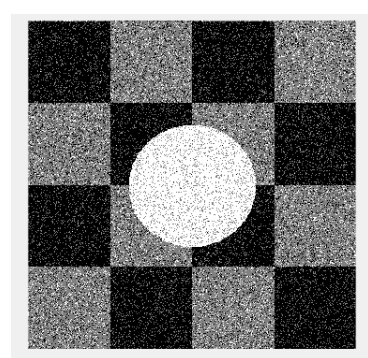
(آ) هیستوگرام تصویر نویزی با $\sigma = 0.1$



(و) خروجی الگوریتم با β متغیر

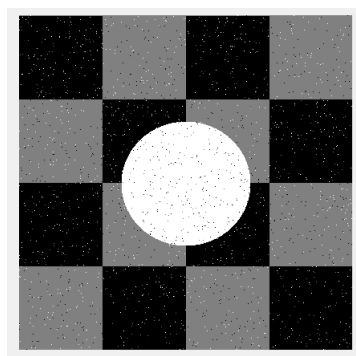


(ه) خروجی الگوریتم با $\beta = 0.8$

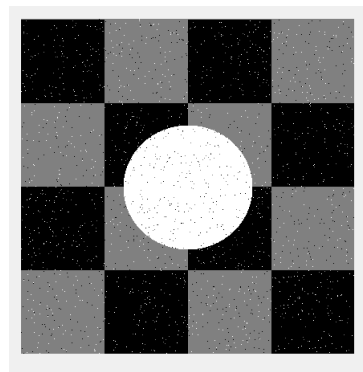


(د) خروجی الگوریتم با $\beta = 0.2$

شکل ۴: مقایسه تاثیر پارامتر β

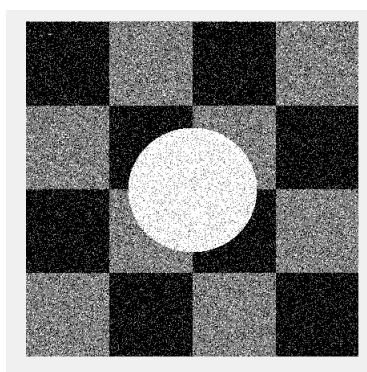


(ب) تغییر دما به صورت نمایی. نتیجه در تکرار چهارم

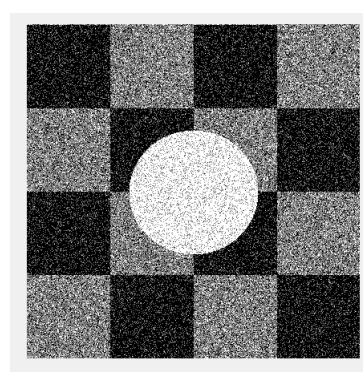


(آ) تغییر دما به صورت خطی. نتیجه در تکرار چهارم

شکل ۵: مقایسه تاثیر نحوه تغییر دما با تصویر ورودی بدون نویز

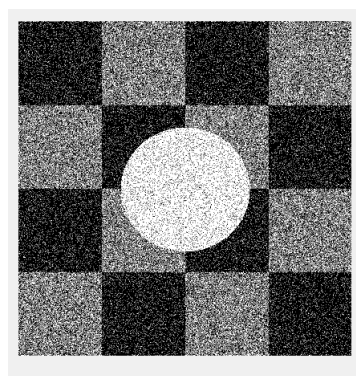


(ب) دسته‌بندی با مقداردهی اولیه ۵ درصد از نقاط

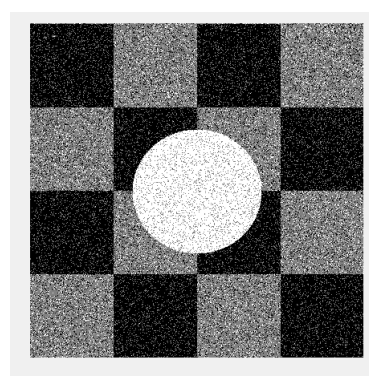


(آ) دسته‌بندی معمولی

شکل ۶: مقایسه تاثیر مقداردهی اولیه برخی نقاط. تصویر نویزی با $\sigma = 0.1$

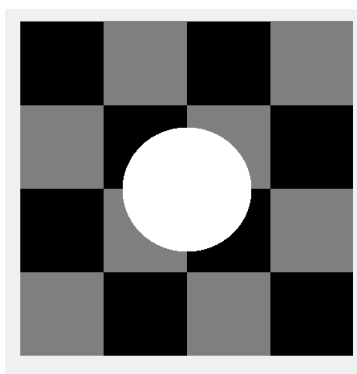


(ب) همسایگی ۴ تایی

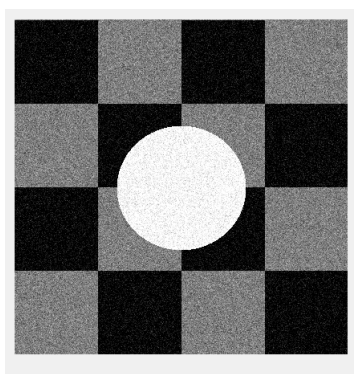


(آ) همسایگی ۸ تایی

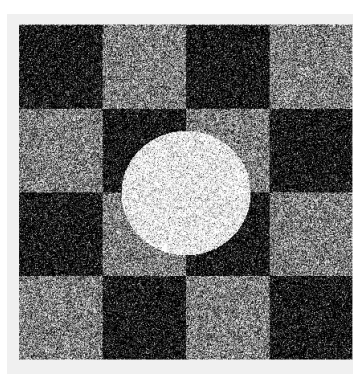
شکل ۷: مقایسه تاثیر نوع همسایگی. تصویر نویزی با $\sigma = 0.1$



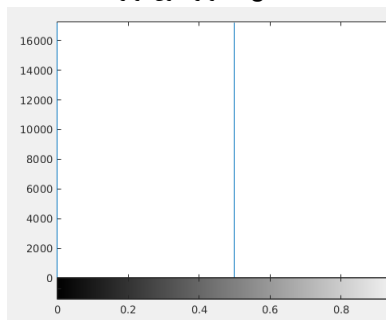
(ج) تصویر بدون نویز



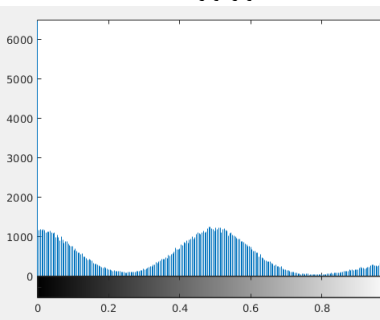
(ب) تصویر نویزی با $\sigma = 0.01$



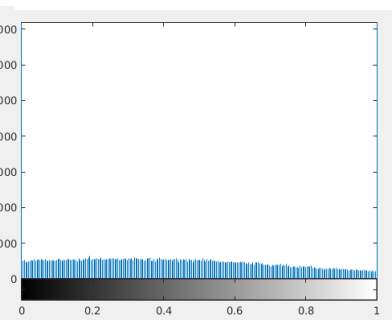
(آ) تصویر نویزی با $\sigma = 0.1$



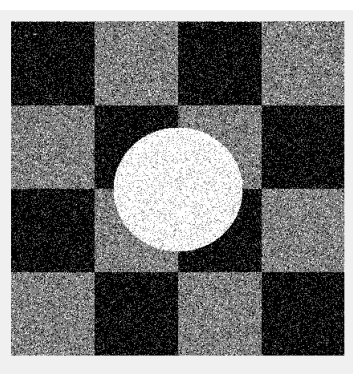
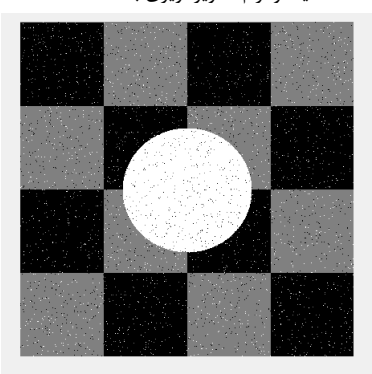
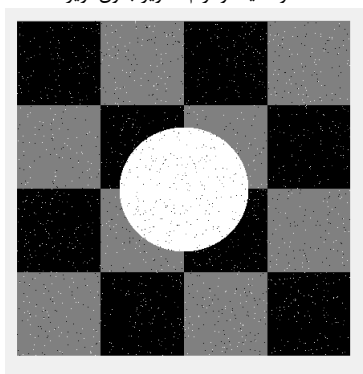
(و) هیستوگرام تصویر بدون نویز



(ه) هیستوگرام تصویر نویزی با $\sigma = 0.01$



(د) هیستوگرام تصویر نویزی با $\sigma = 0.1$



(ز) خروجی الگوریتم برای تصویر نویزی با $\sigma = 0.1$ (ح) خروجی الگوریتم، تصویر نویزی با $\sigma = 0.01$ (ط) خروجی الگوریتم برای تصویر نویزی با $\sigma = 0$

شکل ۸: مقایسه تاثیر نویز در عملکرد الگوریتم



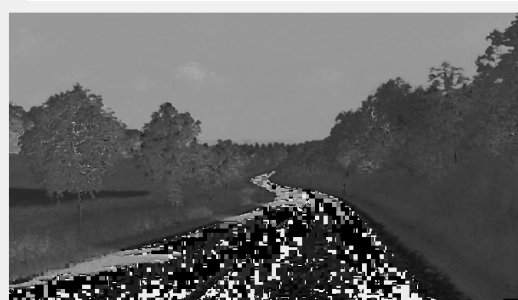
(ب) خروجی الگوریتم برای تصویر HSV با $\sigma = 0.1$



(آ) خروجی الگوریتم برای تصویر سطح خاکستری با $\sigma = 0$



(د) سطح S تصویر ورودی



(ج) سطح H تصویر ورودی

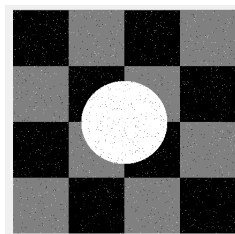


(ه) سطح V تصویر ورودی



(و) تصویر ورودی نویزی برای دسته‌بندی‌کننده در فضای رنگی HSV

شکل ۹: مقایسه عملکرد الگوریتم در فضاهای رنگی مختلف



(ب) خروجی حالت استفاده از بیش از یک ویژگی



(آ) خروجی حالت سطح خاکستری برای تصویر بدون نویز

شکل ۱۰: مقایسه تاثیر استفاده بیش از یک ویژگی