

مدل‌های احتمالی گرافی

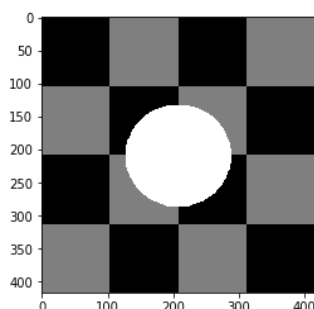
Markov Random Field

علی مرتضوی

بخش اول

در این بخش می‌خواهیم با استفاده از مدل Markov Random Field، تصویری که دارای نویز است را ترمیم کنیم.

تصویر درست به صورت زیر است:



الف

در این قسمت با استفاده از نویز گوسی مقادیر هر پیکسل را به طور مستقل تغییر می‌دهیم.

$\text{Pixel}[i][j] += \text{Normal}(0, \text{var})$

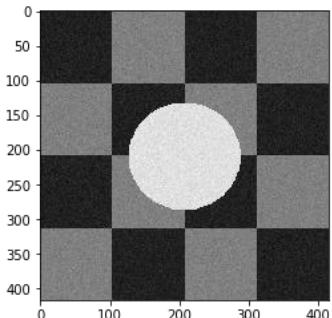
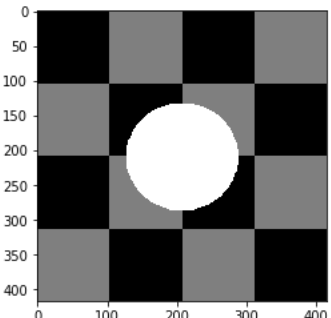
سپس احتمالات $p(\text{pixel}|\text{label})$ را با توجه به تصویر اصلی بدست می‌آوریم. (میانگین و واریانس را بدست می‌آوریم.)

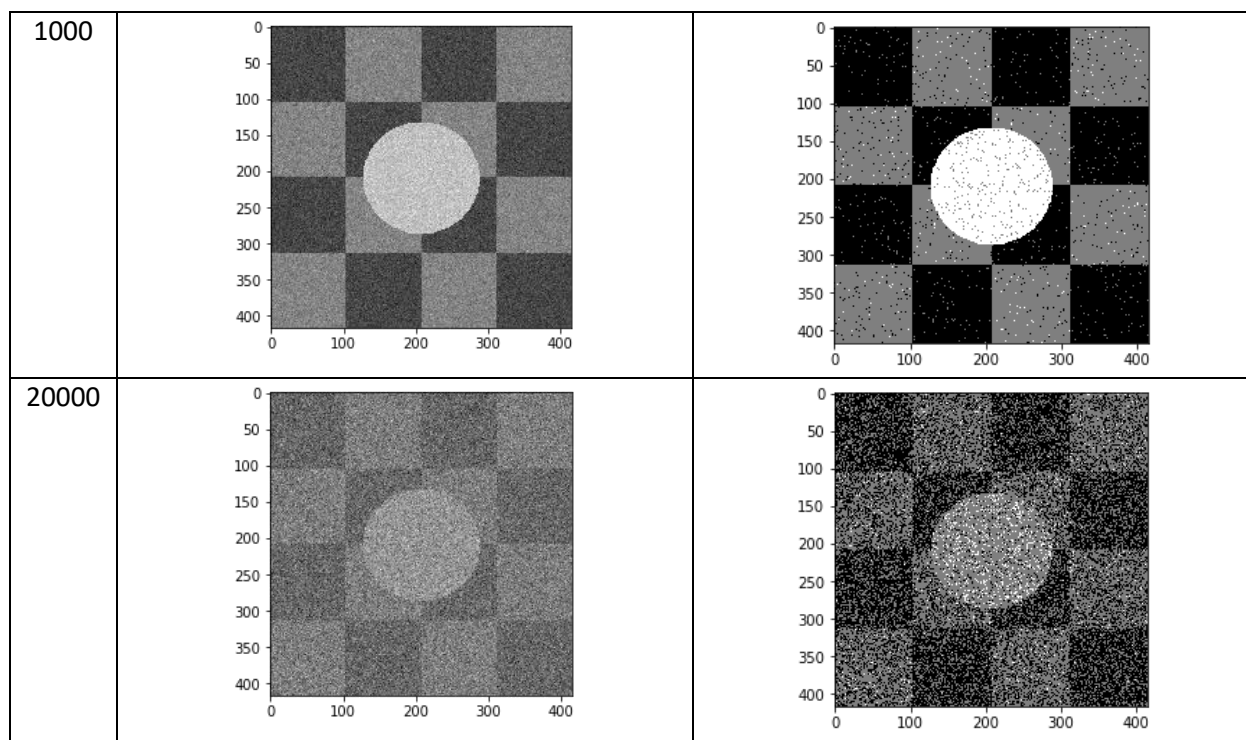
ب

در این قسمت به بررسی عملکرد برچسب زن ساده می‌پردازیم. ما تصویر اصلی را با سه مقدار واریانس مختلف [۱۰۰, ۱۰۰۰, ۱۰۰۰۰] نویزی کردیم.

سپس به ازای هر کدام از این تصاویر نویزی، با استفاده از دسته بند ساده، هر کدام از پیکسل ها را برچسب می‌زنیم.

نتایج به صورت زیر است:

Var	Noisy input	Predicted output
100		



همانطور که میبینید دسته بند ساده برای تصاویر با نویز کم بسیار خوب عمل میکند و تقریباً ۱۰۰ درصد تصویر را به درستی بازیابی میکند. اما زمانی که نویز زیاد باشد، عملکرد دسته بند ساده به شدت کاهش می‌یابد.

ج

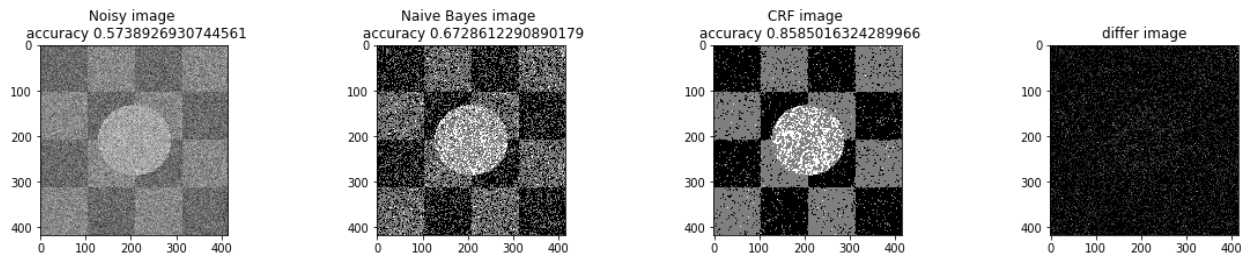
در این قسمت از مدل MRF استفاده می‌کنیم و سعی در بهینه سازی تابع پتانسیل خواهیم داشت. برای بهینه سازی از سردکردن تدریجی استفاده کردیم.

رابطه پتانسیل به صورت زیر بوده است:

$$U(w) = \sum_s (\lg(\sigma_{\omega_s} \sqrt{2\pi}) + \frac{(f_s - \mu_{\omega_s})^2}{2(\sigma_{\omega_s})^2}) + \sum_{s,r} \beta \delta(s, r)$$

د

در این قسمت نتیجه حاصل از مدل MRF را با مدل دسته بند ساده مقایسه می‌کنیم.



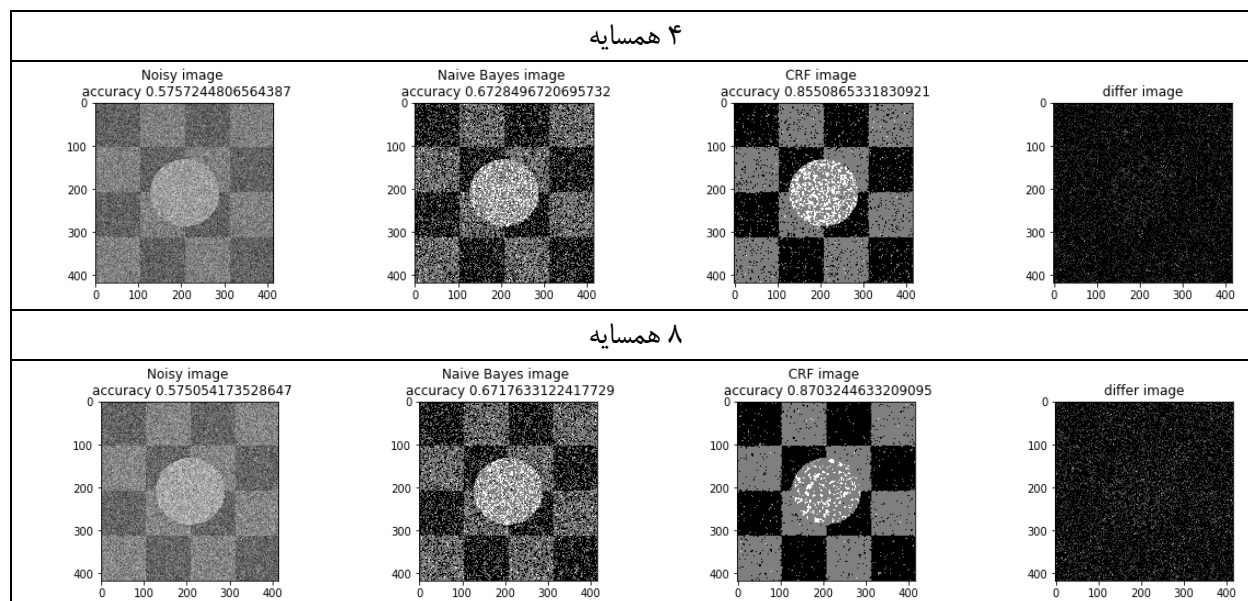
شکل ۱ در این شکل، تصاویر از سمت چپ به ترتیب عبارتند از: تصویر نویزی، نتیجه دسته بند ساده، نتیجه مدل CRF و پیکسل های تغییر داده شده در مدل CRF.

هـ

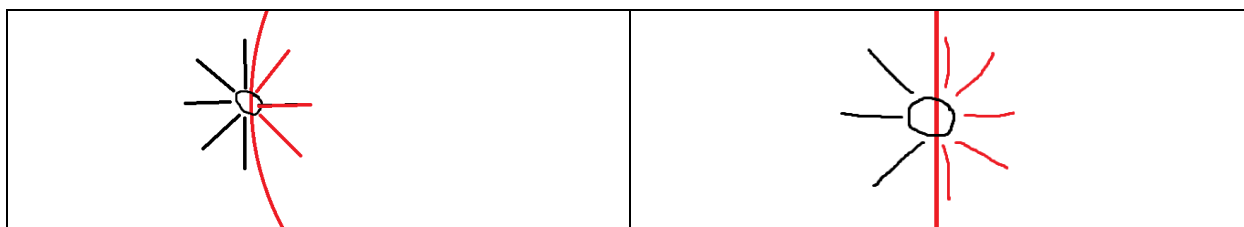
در این قسمت به بررسی تاثیر نوع همسایگی بر عملکرد مدل می پردازیم.

مدل اول با استفاده تنها از ۴ همسایه استفاده کرده است.

مدل دوم از ۸ همسایه استفاده کرده است.



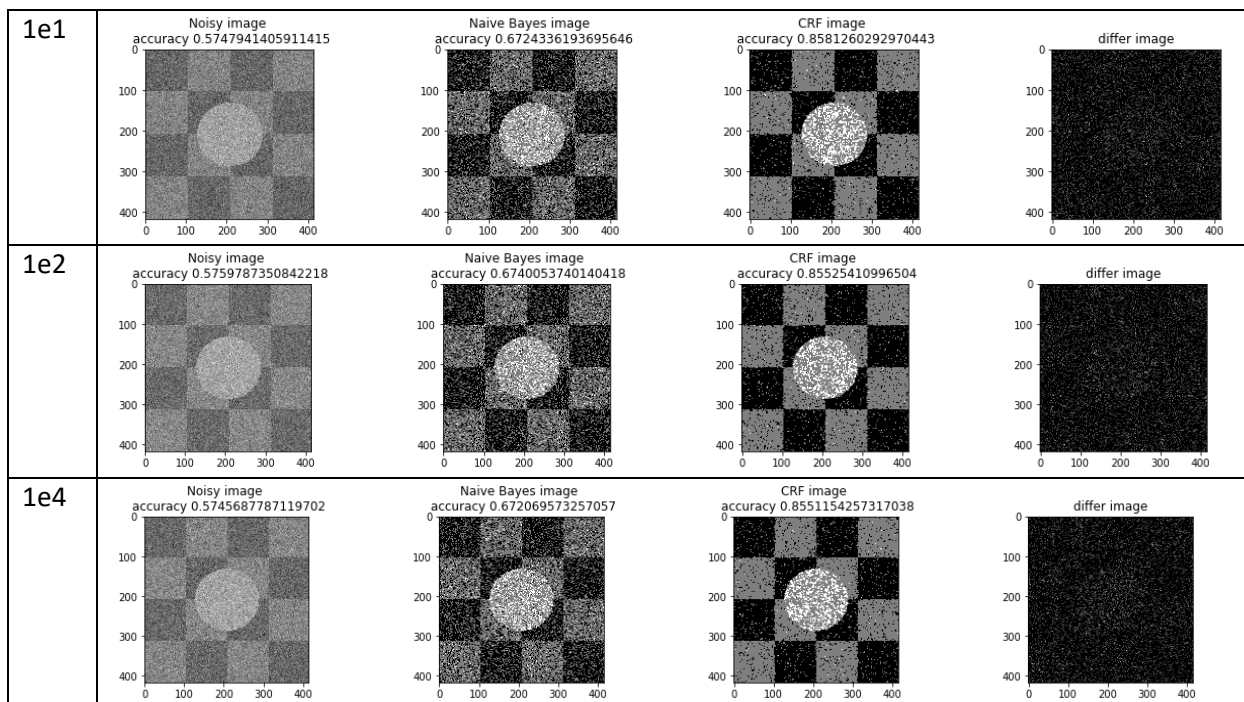
عملکرد ۸ همسایه در تشخیص مربع ها خوب بوده است اما در تشخیص دایره عملکرد خوبی نداشته است. دلیل این امر این است که همسایگی در نقاط حاشیه ای برای دایره با همسایگی در نقاط حاشیه ای مربع فرق دارد. با توجه به اینکه در ۸ همسایه ای که در نقاط گوشه ای دایره هستند، تنها ۳ تا از آنها هم رنگ هستند و ۵ تای آنها غیر هم رنگ هستند، در صورتی که ضریب بتا در مدل زیاد باشد، مدل نمیتواند به خوبی این نقاط را به عنوان نقاط گوشه ای تخمین بزند. اما در مربع با توجه به اکثر ۵ نقطه هم رنگ و ۳ نقطه غیر هم رنگ هستند، نتیجه بسیار خوب می شود.



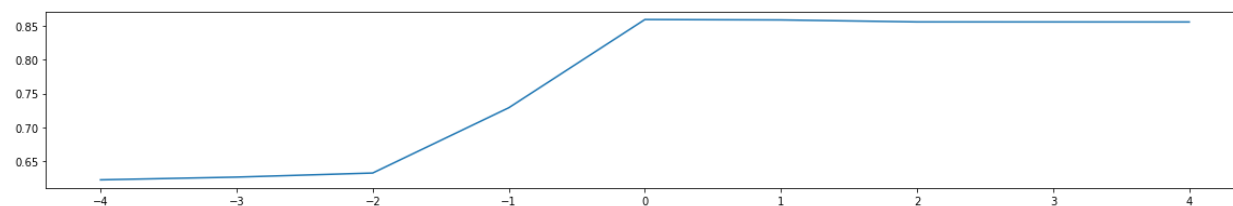
9

در این قسمت به تاثیر پارامتر β در عملکرد مدل می‌پردازیم.

β	output			
1e-4	Noisy image accuracy 0.5747247984744734 	Naive Bayes image accuracy 0.6716708560862154 	CRF image accuracy 0.6232064950449279 	differ image
1e-3	Noisy image accuracy 0.5748288116494756 	Naive Bayes image accuracy 0.6728034439917945 	CRF image accuracy 0.627372800554737 	differ image
1e-2	Noisy image accuracy 0.5747594695328075 	Naive Bayes image accuracy 0.6722198145098379 	CRF image accuracy 0.6332553234520817 	differ image
1e-1	Noisy image accuracy 0.5738117939383433 	Naive Bayes image accuracy 0.6713761520903759 	CRF image accuracy 0.7291034642165786 	differ image
1	Noisy image accuracy 0.5742162896189072 	Naive Bayes image accuracy 0.6717633122417729 	CRF image accuracy 0.8587096587790009 	differ image



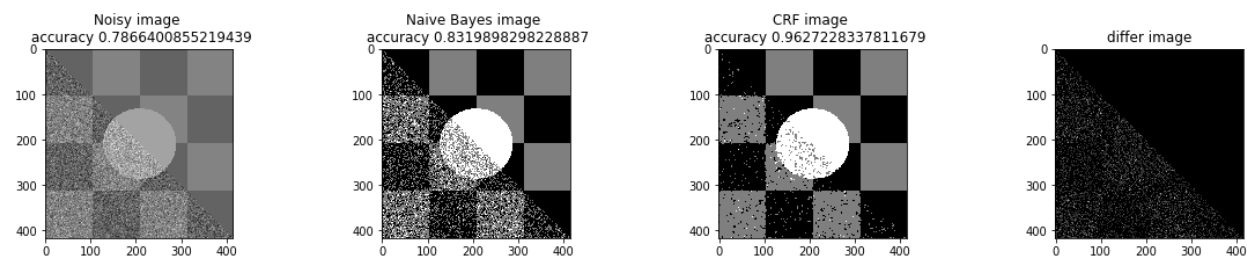
نمودار دقت به ازای بتا، به صورت زیر می شود:



ز

در این قسمت به بررسی تاثیر مقدار دهی اولیه برخی پیکسل ها میپردازیم.

در حالتی که نصف داده ها برچسب گذاری شده باشد، نتیجه به صورت زیر شد:



ح

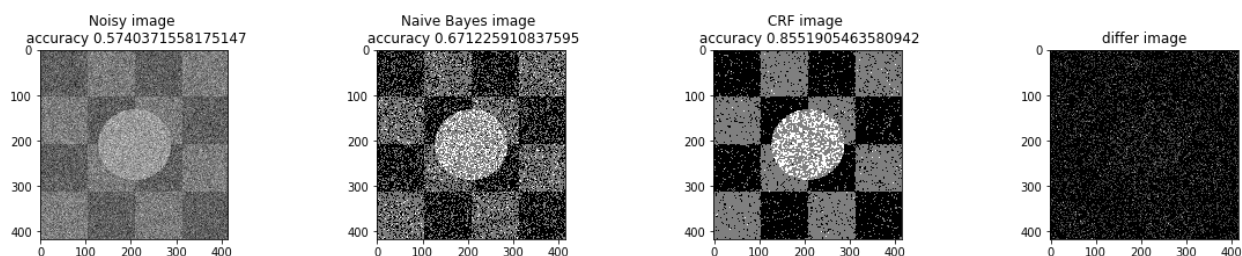
در این قسمت به بررسی تاثیر روش تنظیم پارامتر دما می پردازیم.

حالت اول

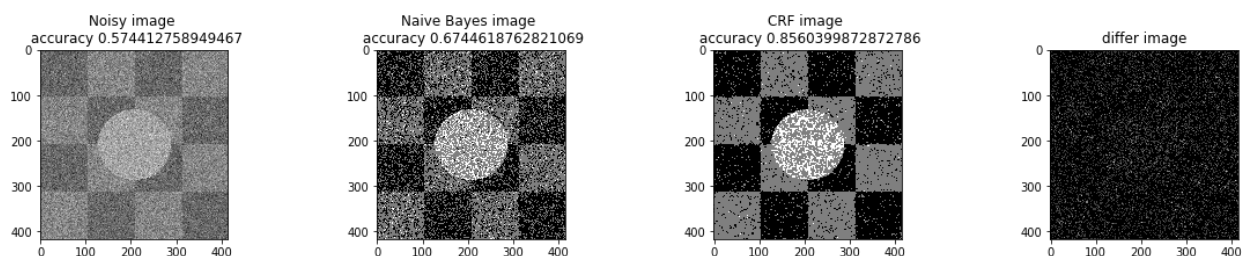
Exponential Schedule

$$t_{n+1} = t_n * constant$$

- Constant=0.8



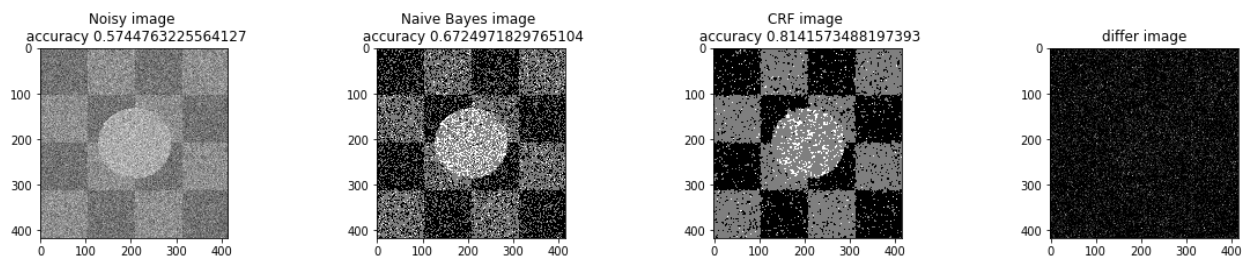
- Constant=0.9



Logarithmical Multiplicative Cooling Schedule

$$t_{n+1} = \frac{t_0}{1 + c * \lg(n + 1)}$$

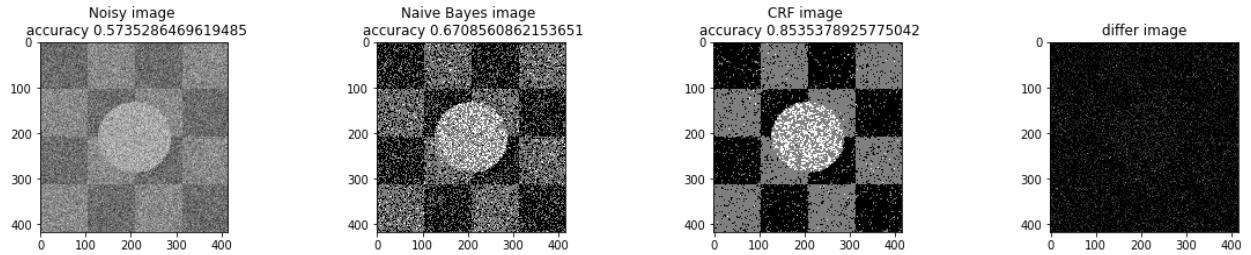
- c = 1



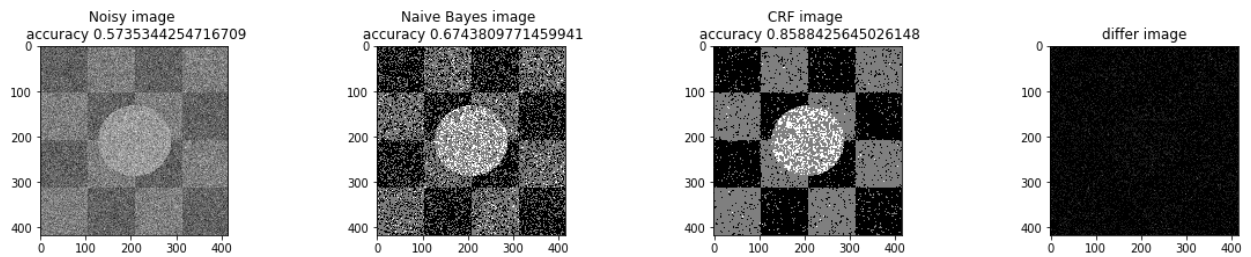
Linear Multiplicative Cooling Schedule

$$t_{n+1} = \frac{t_0}{1 + c * n}$$

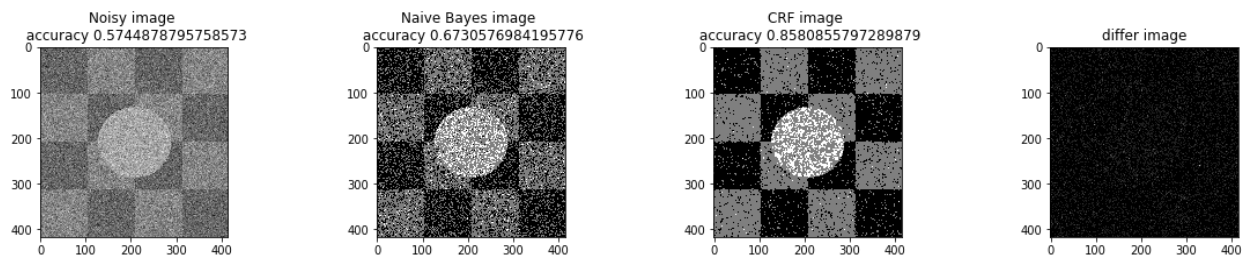
- C = 1



- C = 0.5



- C = 1.5



بخش دوم

در این بخش به قطعه بندی یک عکس رنگی به سمت قسمت آسمان، جاده، حاشیه جاده می پردازیم.

تصویر اولیه به شکل زیر است.

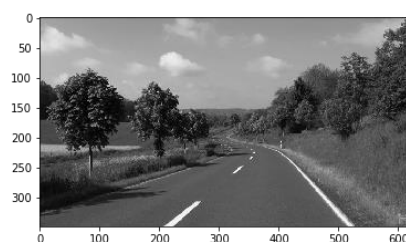


الف

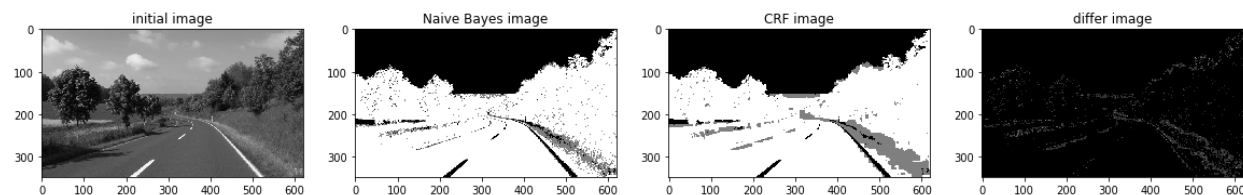
در این قسمت به قطعه بندی عکس با مدل MRF می پردازیم. برای یادگیری، ما با توجه به پیکسل های داخل این عکس، سه فایل با نام های Road.jpg، Sky.jpg، و Grass.jpg ساختیم و میانگین و واریانس کلاس ها را با استفاده از این سه فایل بدست آوردیم.

حالت خاکستری

در این حالت، عکس را به حالت خاکستری تبدیل کرده و سپس مدل را آموزش می دهیم.



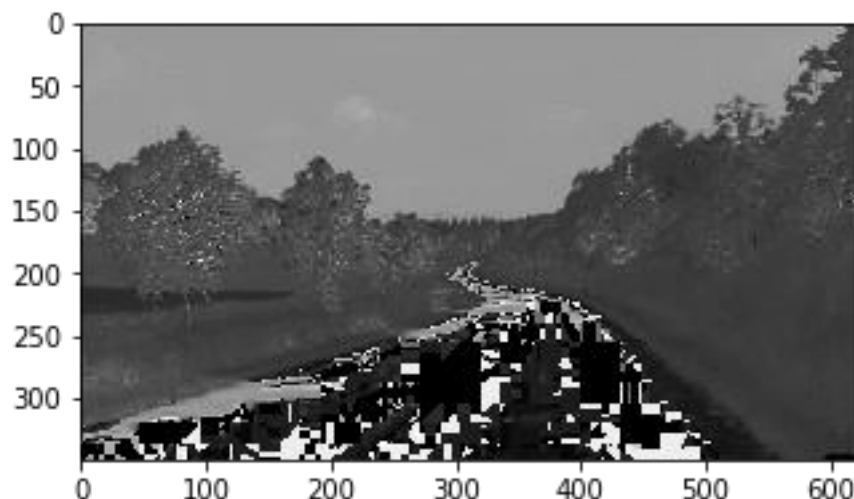
پس از آموزش نتیجه به صورت زیر شد.



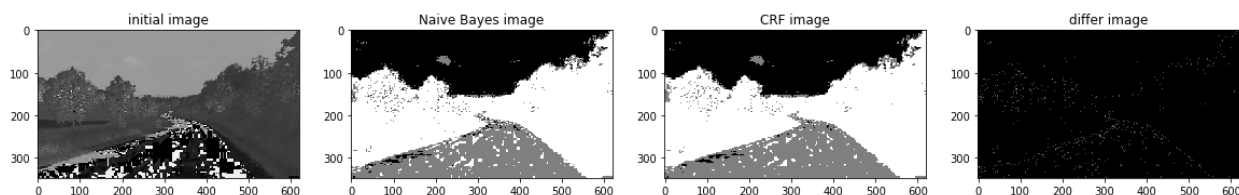
همانطور که مشخص است، این مدل در تشخیص حاشیه تصویر دچار مشکل است. اما آسمان را با دقت خوبی تشخیص داده است. البته بخشی از جاده که رنگ سفید تری دارند را نیز آسمان تشخیص داده است.

حالت HSV

در این قسمت از حالت HSV استفاده می‌کنیم. و از مقدار H برای یادگیری مدل استفاده می‌کنیم. مقادیر H در تصویر به صورت زیر هستند.



نتیجه مدل به صورت زیر شد.



عملکرد در این مدل از بازنمایی عکس بهتر از مدل خاکستری بوده است. دلیل این موضوع این است که اطلاعاتی که مقدار H در تصویر به ما می‌دهد برای این مساله موثر تر است.

البته می‌بینیم قطعه ای از ابر را "حاشیه جاده" و همچنین قسمتی از جاده را نیز حاشیه جاده در نظر گرفته است.

ب

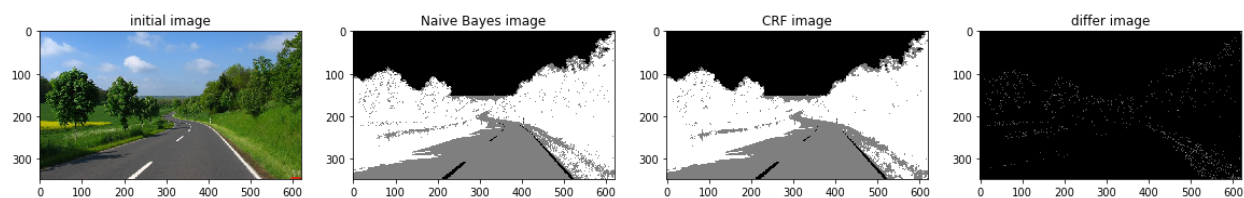
در این قسمت از ما خواسته شده است که با استفاده از ویژگی های دیگر این مدل را آموزش دهیم. در این قسمت ما حدس زدیم توجه به رنگ می‌تواند در عملکرد مدل موثر باشد. در نتیجه از عکس به صورت RGB استفاده کردیم. بدین ترتیب در مدل MRF، در تابع پتانسیل برای ویژگی های به ازای یک پیکسل، هر سه رنگ را در نظر گرفته و هرچه این ۳ رنگ از میانگین رنگ های اصلی فاصله داشته باشد، هزینه تابع پتانسیل افزایش می‌یابد.

در واقع در مدل ما استقلال سه رنگ در تابع پتانسیل مشاهده می‌شود.

$$p(R = r, G = g, B = b | mean_{rgb}, var_{rgb}) \\ = p(R = r | mean_r, var_r) * p(G = g | mean_g, var_g) * p(B = b | mean_b, var_b)$$

در تابع پتانسیل برای انرژی دو نقطه همسایه نیز دقیقاً از مدل قبلی استفاده کردیم.

نتیجه به صورت زیر شد.



همانطور که میبیند نتیجه بسیار خوب بوده است. و تنها قسمت سایه درخت‌ها در جاده و خطوط جاده به صورت اشتباه دسته بندی شده است. به نظر میرسد در صورت اضافه کردن ویژگی H از قسمت قبل به این مدل، شاهد افزایش چشمگیر عملکرد مدل خواهیم بود.