

به نام یزدان پاک



سیستم های چندرسانه ای
گزارش پروژه نهایی

نگارنده :

مهیار عنصری ۹۶۳۲۰۹۳

تاریخ نگارش : بهمن ۱۳۹۹

توضیحات تابع `embed_proj` :

ابتدا ابعاد تصویر حامل را به دست آوردم و سپس باقی مانده تقسیم ابعاد آن بر اندازه بلوک ها را از آن کم کردم تا به نزدیکترین مضرب ابعاد بلوک ها برسم. بر فرض مثال برای تصویر 450×450 و بلوک های 8×8 حاصل این کار عدد ۴۴۸ که نزدیکترین مضرب ۸ به ۴۵۰ است می باشد.

سپس تعداد بلوک هایی که در سطر و ستون داریم را از تقسیم این مقدار جدید بر بلوک ها به دست آوردم که در مثال فوق $448/8=56$ می باشد. باید تصویر لوگو در نهایت تعداد بیتی به اندازه حاصل ضرب این بلوک ها داشته باشد تا بتوانیم هر بیتش را در یک بلوک جاسازی کنیم.

در گام بعدی با دستور `imresize` لوگوی داده شده را به اندازه لازم تغییر دادم (که بتوانیم هر بیت از تصویر لوگو را در هر بلوک آن جاسازی کنیم) و سپس به صورت سطری ماتریس حاصله را مرتب کردم یعنی سطرهای آن را پشت سر هم چیدم تا از ماتریسی دو بعدی به برداری سطری تبدیل شود. سپس به کمک توضیحات داده شده در صورت سوال لوگو را رندوم کردم.

در گام بعدی دو حلقه تو در تو گذاشتم و هر بار یک بلوک $B \times B$ را از تصویر اصلی انتخاب کردم و از آن `dct` گرفتم، حال با توجه به بیتی از رشته بیت رندوم شده قرار است جاسازی شود شرط را روی درایه های $(a+1, a)$ و $(a, a+1)$ انجام دادم. یعنی مثلاً اگر بیت k ام از رشته بیت رندوم شده برابر یک بود و درایه $(a+1, a)$ از $(a, a+1)$ بزرگتر بود نیازی به جابجایی نیست و تنها $\alpha/2$ به درایه بزرگتر اضافه و از درایه کمتر کم می کردم اما اگر درایه $(a+1, a)$ از درایه $(a, a+1)$ کوچکتر بود آنگاه در ضرایب `dct` نیاز به جابجایی داشتیم و بعد از جابجایی دوباره مثل حالت قبل $\alpha/2$ به درایه بزرگتر اضافه و از درایه کمتر کم میکردم. مانند همین کار برای حالتی که بیت k ام برابر صفر بود نیز انجام می شود.

در نهایت از ضرایب `dct` جدید عکس `dct` میگیرم و در بلوک نظیر در تصویر `W_image` جایگذاری می‌کند. همچنین سطر و ستون هایی از تصویر I که بزرگتر از بزرگترین مضرب ابعاد بلوک بودند (مثلاً در مثال فوق سطرها و ستون های ۴۴۹ و ۴۵۰) را عیناً از تصویر I به تصویر `W_image` منتقل کردم و در آخر `psnr` بین این دو را حساب کردم.

توضیحات تابع `attack_proj` :

در اینجا همان کارهایی که در تابع `embed_proj` برای ابعاد انجام دادیم را روی `W_image` تکرار می کنیم. سپس با دستور `imwrite` تصویر را با سه کیفیت ۶۰ و ۸۰ و ۱۰۰ به صورت `jpg` ذخیره می کنیم و دوباره آن ها را می خوانیم.

حال شبیه کاری که در تابع قبلی انجام دادم را انجام می دهیم یعنی با قرار دادن دو حلقه تودرتو، از هر سه تصویر بلوک های $B \times B$ استخراج می کنیم و `dct` می گیریم. حال اگر در ماتریس ضرایب `dct` به دست آمده درایه $(a+1, a)$ از درایه $(a, a+1)$ بزرگتر بود در بیت نظیر در رشته بیت ۱ قرار می دهیم و در غیر این صورت صفر می گذاریم و سه رشته بیت برای سه کیفیت می سازیم.

حال این سه رشته بیت ساخته شده را در فرآیندی شبیه به تابع قبلی اما اینبار `derandomize` می کنیم و رشته بیت های ساخته شده را با کمک یک `for` به ماتریس های دو بعدی تبدیل می کنیم که در حقیقت باید تصاویر لوگو باشند.

همچنین ورودی `W2D` را در عملی مثل تابع قبل به یک بردار سطری تبدیل می کنیم (با کمک دو حلقه `for` تو در تو) و سه بار به صورت مجزا آن را با بردارهای سطری `derandomize` شده به تابع `NC_calculator` که برای محاسبه `NC` نوشتم (در `MATLAB` دستوری برای `xnor` نبود) می دهیم و سه مقدار `NC60, NC80, NC100` را به دست می آوریم.

در نهایت تصویرهای دو بعدی ساخته شده را با تصویر دریافتی نمایش می دهیم.

توضیحات تابع `NC_calculator` :

این تابع دو رشته بیت میگیرد و اگر هم طول باشند بیت های نظیر آن ها را با هم مقایسه می کنیم. اگر بیت های نظیرشان یکسان بود (حالتی که `xnor` یک می شود) یکی به شمارنده اضافه می شود. در نهایت شمارنده را بر حاصل ضرب ابعاد تقسیم می کنیم تا به `NC` برسیم.

راه به دست آوردن بهترین Alpha

مشاهده کردیم که با افزایش Alpha مقدار PSNR کم می شود اما میانگین سه NC زیاد می شود. می دانیم که هرچه NC زیاد شود لوگوی بهتری را خواهیم دید و هرچه PSNR زیادتر باشد تصویر واترمارک شده با تصویر اصلی همخوانی بهتری دارد. (یا اگر با MSE کار می کنیم MSE کم شود)

میزان NC نشان دهنده Robustness و میزان PSNR نشان دهنده Transparency است یا به عبارتی هرچه MSE کمتر باشد Transparency بیشتر است.

```

4 - Logo=imread('iut5.bmp');
5 - Logo=imbinarize(Logo);
6 - Lena=imread('lena.bmp');
7 - Alpha_Iteration=150;
8 - Psnr=zeros(1,Alpha_Iteration);
9 - MSE=zeros(1,Alpha_Iteration);
10 - NC60=zeros(1,Alpha_Iteration);
11 - NC80=zeros(1,Alpha_Iteration);
12 - NC100=zeros(1,Alpha_Iteration);
13 - Mean_NC=zeros(1,Alpha_Iteration);
14 - QMSENC=zeros(1,Alpha_Iteration);
15 - QPSNRNC=zeros(1,Alpha_Iteration);
16 - for i=1:Alpha_Iteration
17 -     [WATERmark, Psnr(1,i), MSE(1,i)] = embed_proj(Lena, 8, 4, Logo, 19, i);
18 -     [NC60(1,i), NC80(1,i), NC100(1,i)] = attack_proj(WATERmark, 8, 4, 19, Logo);
19 -     Mean_NC(1,i) = ( NC60(1,i) + NC80(1,i) + NC100(1,i) ) / 3 ;
20 -     QMSENC(1,i) = Mean_NC(1,i) / MSE(1,i);
21 -     QPSNRNC(1,i) = Mean_NC(1,i) / Psnr(1,i);
22 -     close all
23 - end

```

با اندکی تغییرات در ساختار توابع (اضافه کردن خروجی MSE به تابع embed_proj و قرار دادن خروجی های NC60, NC80, NC100 برای تابع attack_proj) دو تابع را پشت سر هم برای Alpha های مختلف صدا می زنیم و نتایج را نگاه می کنیم. همان گونه که در زیر آمده است بر فرض مثال برای B=8 و Alpha=44 هر سه NC برابر یک می شود و بهترین حالت را داریم. اما شاهد این هستیم که psnr روند نزولی دارد که خوب نیست پس برای Alpha های بزرگتر از ۴۴ در حالتی که بلوک های ۸ تایی دیگر افزایش Alpha مناسب نیست چون NC به ماکزیمم خود می رسد ولی psnr همچنان کم می شود که خوب نیست.

برای سه B مختلف امتحان می کنیم:

B=8

Mean_NC											
1x100 double											
	38	39	40	41	42	43	44	45	46	47	48
1	0.9770	0.9839	0.9885	0.9973	0.9989	0.9996	1	1	1	1	1

پس Alpha=44 برای B=8 بهترین Alpha است.

B=10

Mean_NC

1x100 double

	65	66	67	68	69	70	71	72	73	74	75	76	77	78
1	0.9998	0.9998	1	0.9998	1	1	1	1	1	1	1	1	1	1

پس Alpha=69 برای B=10 بهترین Alpha است.

B=12

Mean_NC														
1x100 double														
	64	65	66	67	68	69	70	71	72	73	74	75	76	77
1	0.9993	0.9995	0.9995	0.9995	0.9998	0.9998	0.9998	0.9998	1	1	1	1	1	1

پس Alpha=72 برای B=12 بهترین Alpha است.

در صفحات بعد دو سری جدول برای Alpha=44 و Alpha=70 آورده شده است و برای هر دو مقدار Alpha تصاویر بلوک ۸×۸ ضمیمه شده است.

Alpha_max=44**B=8**

	Alpha=0	Alpha=4.4	Alpha=22	Alpha=44
Quality=100	NC=0.9043	NC=1	NC=1	NC=1
Quality=80	NC=0.5855	NC=0.6416	NC=1	NC=1
Quality=60	NC=0.4707	NC=0.4834	NC=0.5666	NC=1
PSNR	47.3405	45.4678	39.4857	35.008
MSE	0.5998	0.9231	3.6574	10.2575

B=10

	Alpha=0	Alpha=4.4	Alpha=22	Alpha=44
Quality=100	NC=0.8953	NC=1	NC=1	NC=1
Quality=80	NC=0.6657	NC=0.7565	NC=0.9921	NC=1
Quality=60	NC=0.5956	NC=0.6296	NC=0.7970	NC=0.9714
PSNR	49.1989	47.2959	41.2971	36.8913
MSE	0.391	0.606	2.4118	6.6511

B=12

	Alpha=0	Alpha=4.4	Alpha=22	Alpha=44
Quality=100	NC=0.8656	NC=0.9993	NC=1	NC=1
Quality=80	NC=0.6698	NC=0.7341	NC=0.9854	NC=1
Quality=60	NC=0.5573	NC=0.5931	NC=0.7421	NC=0.9438
PSNR	50.2042	48.7149	42.9871	1.6343
MSE	0.3102	0.4371	38.453	4.6363

به نظر می رسد که همین Alpha=44 بهترین انتخاب باشد زیرا برای B=8 که هر سه NC در بهترین حالت هستند و برای دو B دیگر فقط برای کیفیت ۶۰ به NC=1 نرسیده ایم و اگر از کیفیت بالا مطمئن باشیم گزینه بسیار مناسبی است.

Alpha_max=70**B=8**

	Alpha=0	Alpha=7	Alpha=35	Alpha=70
Quality=100	NC=0.9043	NC=1	NC=1	NC=1
Quality=80	NC=0.5855	NC=0.6610	NC=1	NC=1
Quality=60	NC=0.4707	NC=0.4914	NC=0.8233	NC=1
PSNR	47.3405	44.5219	36.6131	31.4700
MSE	0.5998	1.1478	7.0836	22.9602

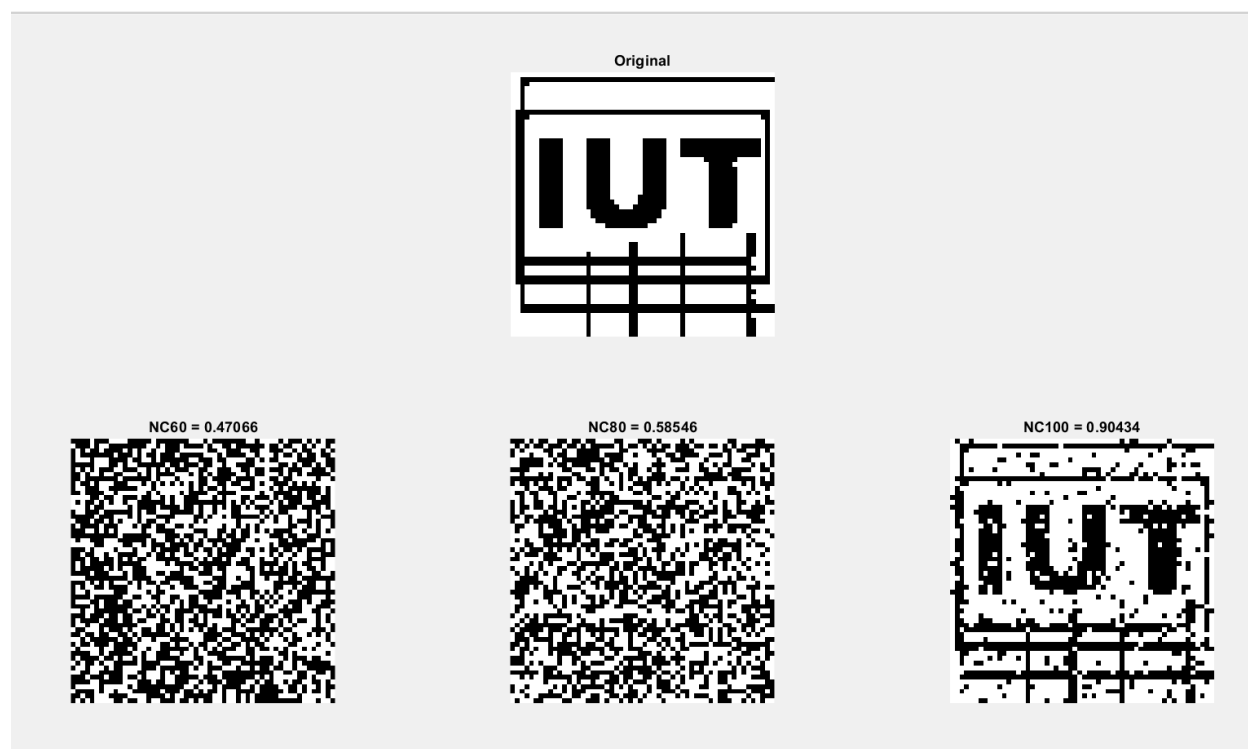
B=10

	Alpha=0	Alpha=7	Alpha=35	Alpha=70
Quality=100	NC=0.8953	NC=1	NC=1	NC=1
Quality=80	NC=0.6657	NC=-0.8296	NC=1	NC=1
Quality=60	NC=0.5956	NC=0.6637	NC=0.8968	NC=1
PSNR	49.1989	46.1634	38.5451	33.3945
MSE	0.391	0.7865	4.5451	14.8630

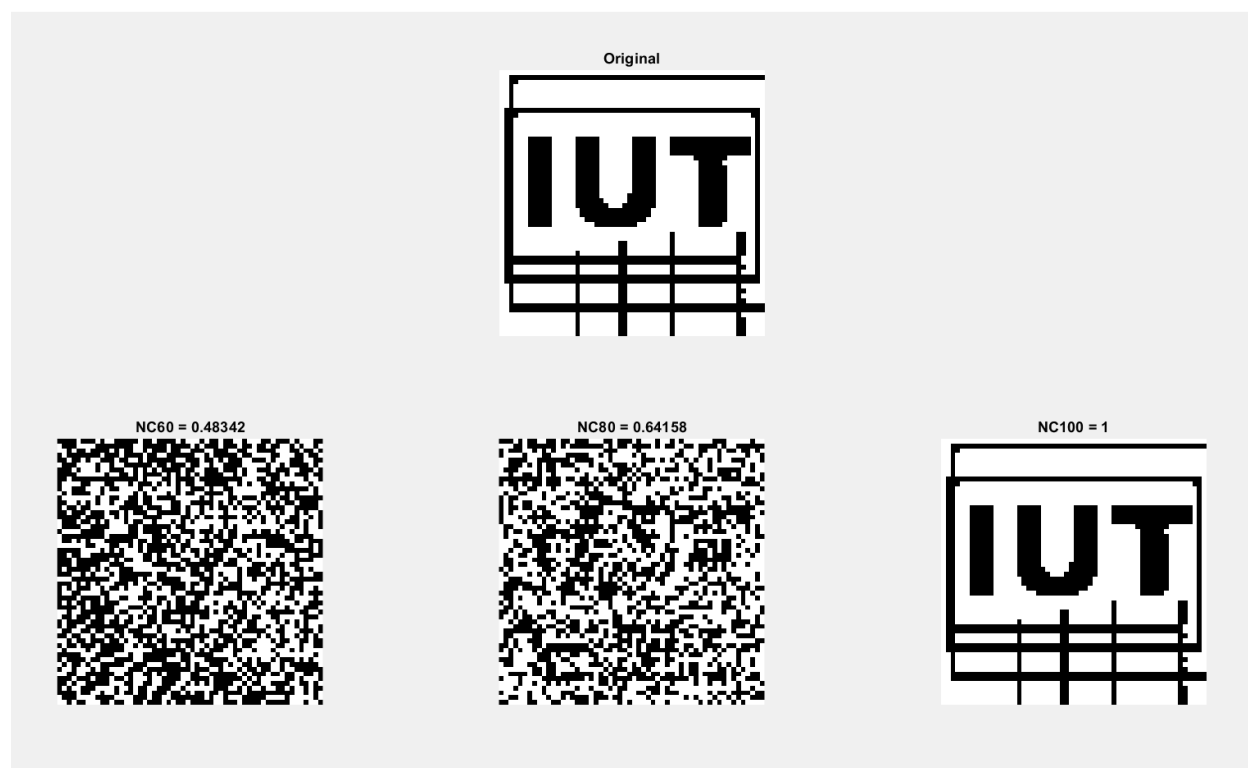
B=12

	Alpha=0	Alpha=7	Alpha=35	Alpha=70
Quality=100	NC=0.8656	NC=1	NC=1	NC=1
Quality=80	NC=0.6698	NC=0.6275	NC=1	NC=1
Quality=60	NC=0.5573	NC=0.8123	NC=0.8890	NC=0.9993
PSNR	50.2042	47.4613	40.0105	35.0009
MSE	0.3102	0.5833	3.2434	10.2714

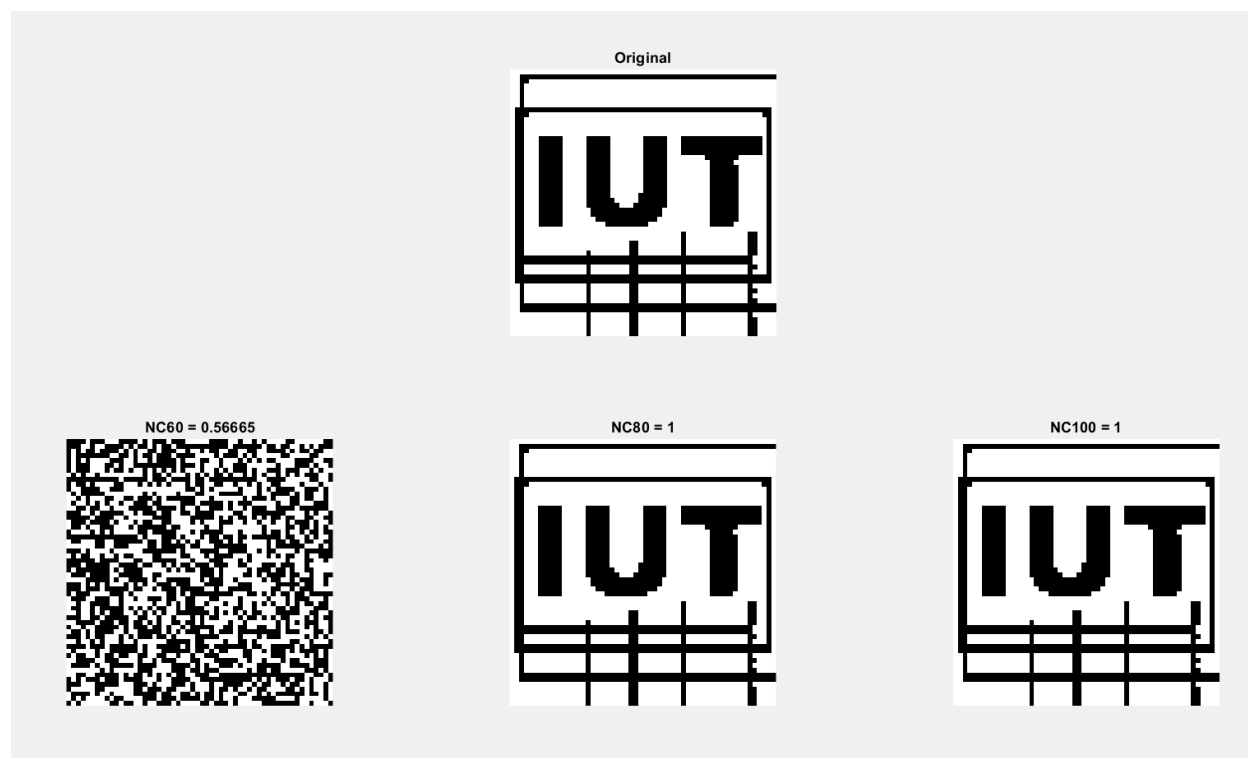
با مقایسه نتایج ستون سوم این جداول با جداول قبلی به این نتیجه می رسیم که Alpha=44 از Alpha=35 بهتر است ولی نیازی به Alpha=70 نیست و فقط PSNR کم می شود.



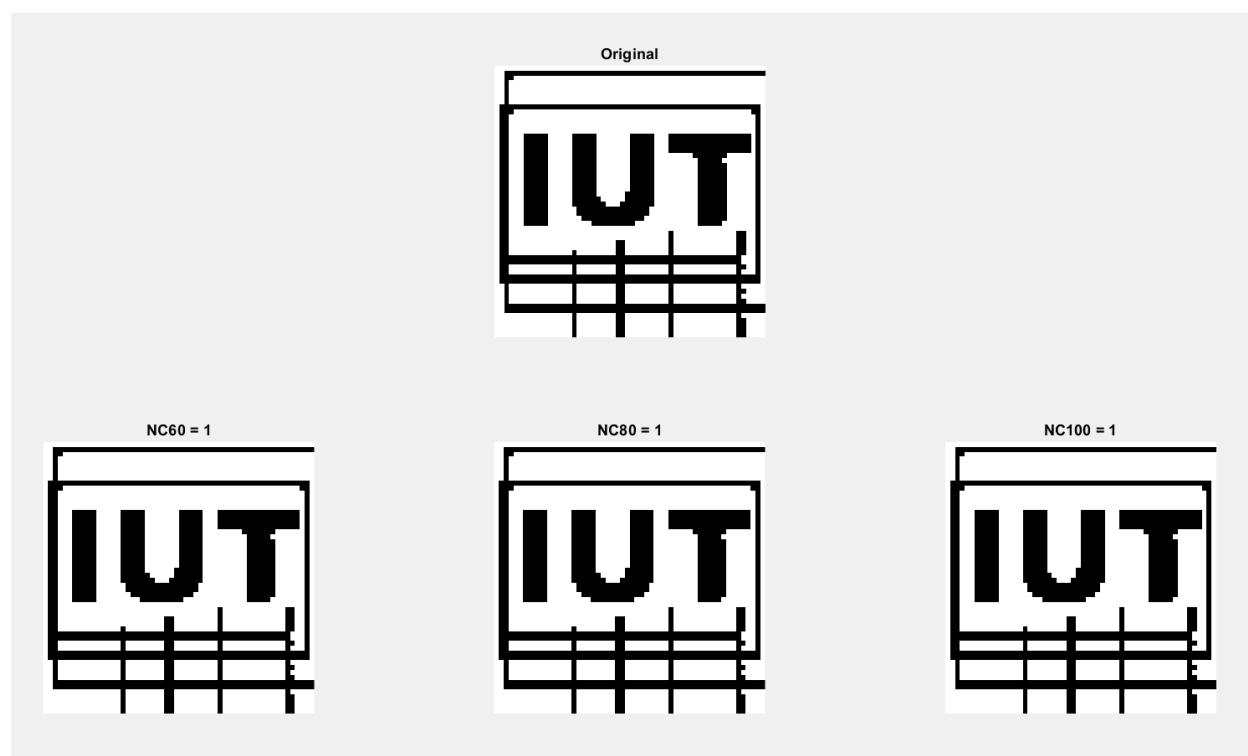
Alpha=0 , B=8 , a=4 , K=19



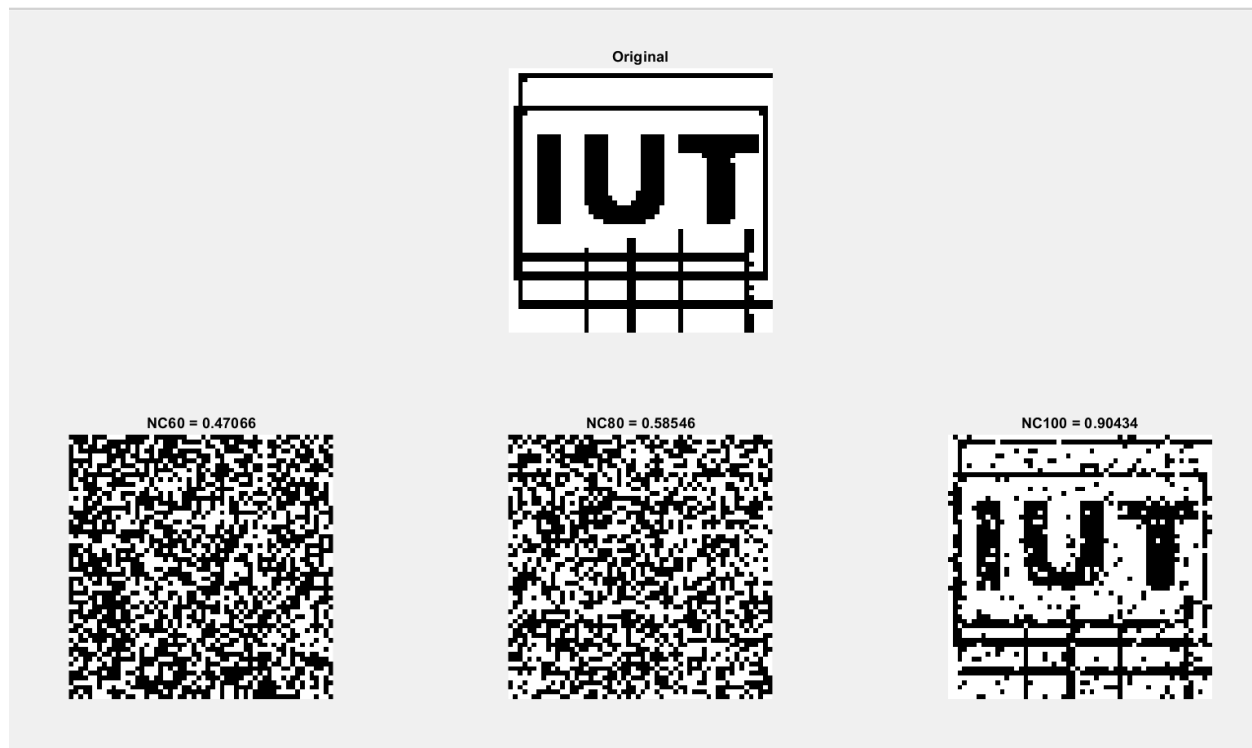
Alpha=4.4 , B=8 , a=4 , K=19



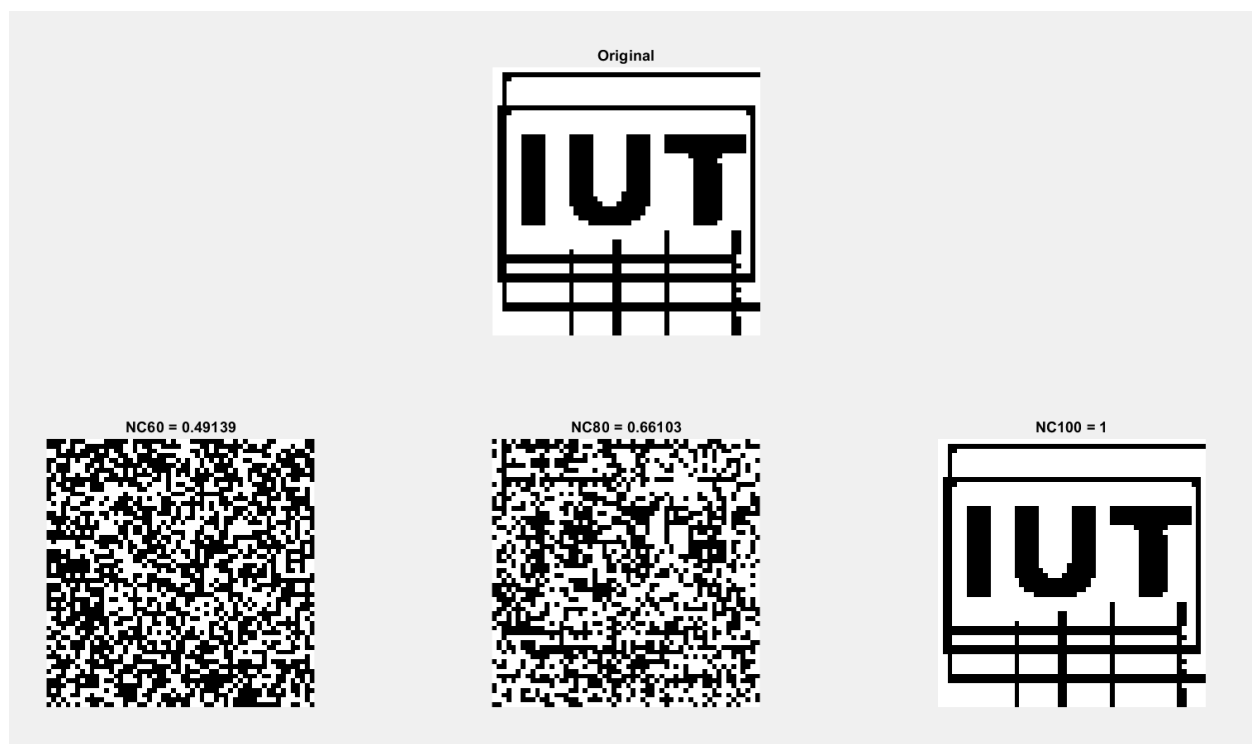
Alpha=22 , B=8 , a=4 , K=19



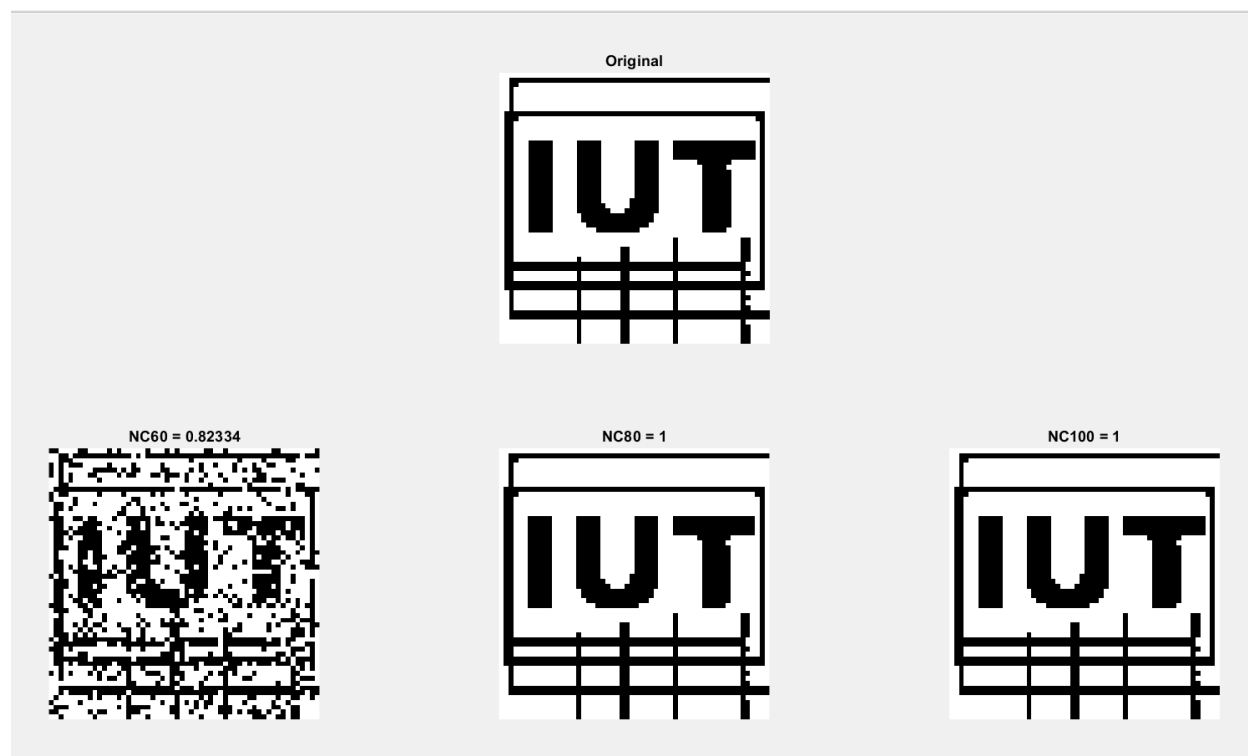
Alpha=44 , B=8 , a=4 , K=19



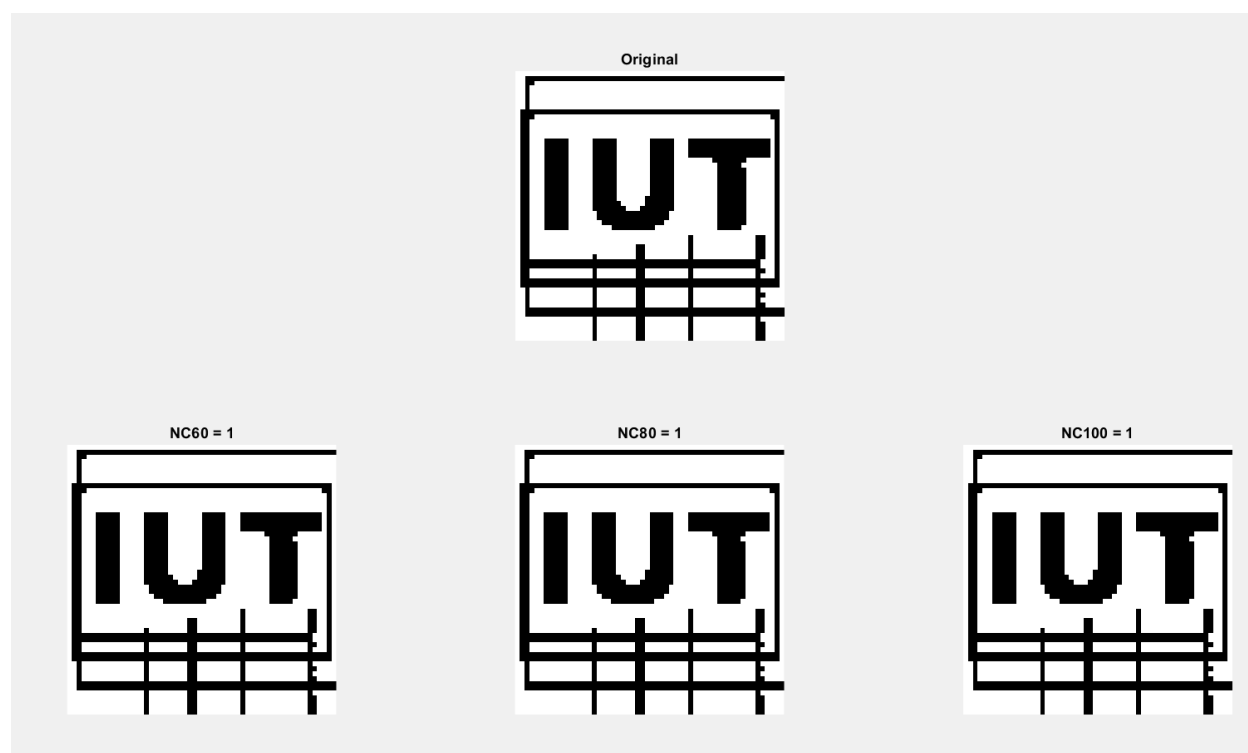
$\text{Alpha}=0$, $B=8$, $a=4$, $K=19$



$\text{Alpha}=7$, $B=8$, $a=4$, $K=19$



Alpha=35 , B=8 , a=4 , K=19



Alpha=70 , B=8 , a=4 , K=19