



## Tema 4 - DENOISING & INPAINTING

ANTRENAREA DICȚIONARELOR PENTRU REPREZENTĂRI RARE.  
APLICAȚII LA ELIMINAREA DE ZGOMOT ȘI RESTAURAREA ÎN IMAGINI

Lupașcu Marian | Concepție și Aplicații în Vederea Artificială | 19 – Dec - 2018

# Cuprins

Cuprins .....	1
Eliminarea de zgomot cu patch-uri distincte .....	2
Eliminarea de zgomot cu patch-uri suprapuse.....	11
Inpainting.....	22
Analiza statisticilor psnr si ssim.....	32
Influenta hiperparametrilor .....	36
Variatia dimensiunii unui patch (parametri.patchSize) .....	38
Variatia numarul de patch-uri folosite la invatarea dictionarului D (parametri.totalNumberOfPatches).....	40
variatia dimensiunii dictionarului D (parametri.dictionarySize) .....	42
Variatia deviatiei standard a zgomotului (parametri.noiseStandardDeviation).....	44
Variatia numarului de iteratii (parametri.DL).....	46

## Eliminarea de zgomot cu patch-uri distincte

În continuare voi prezenta câteva rezultate pentru funcția `test_denoising_distinct.m`. Menționez că am folosit următorii hiperparametrii pentru fiecare din exemplele pe care le voi enumera:

```
parametri.patchSize = 8;
parametri.sparsity = 6;
parametri.totalNumberOfPatches = 1000;
parametri.dictionarySize = 256;
parametri.DL = 50;
parametri.noiseStandardDeviation = 20;
```

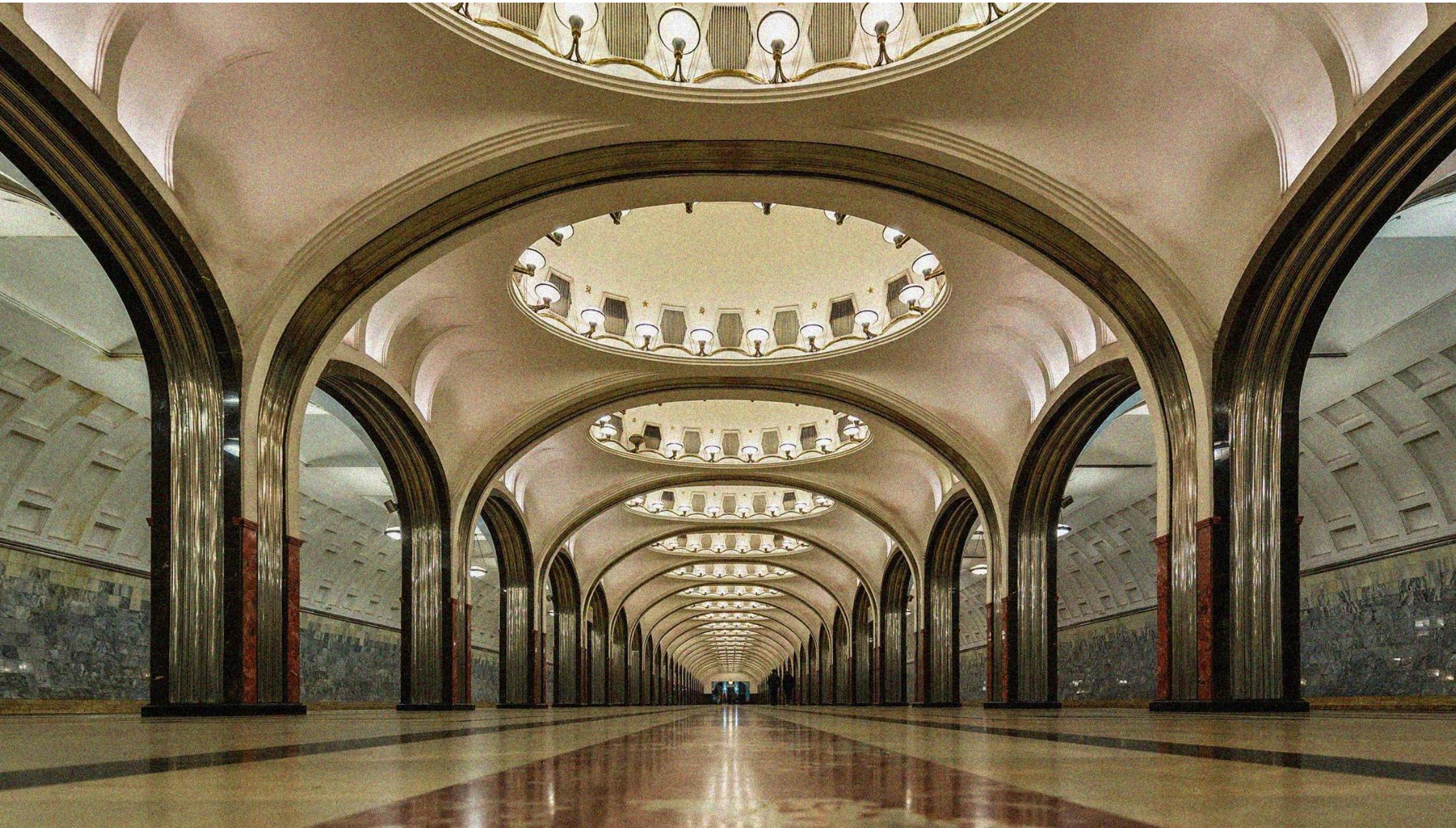
Fiecare exemplu se va întinde pe 3 pagini, prima pagina va reprezenta imaginea inițială, neperturbată, pagina a doua va reprezenta imaginea inițială perturbată cu zgomot alb gaussian, iar pagina a treia va reprezenta rezultatul scos de funcția `test_denoising_distinct.m` pe imaginea de pe pagina a doua.

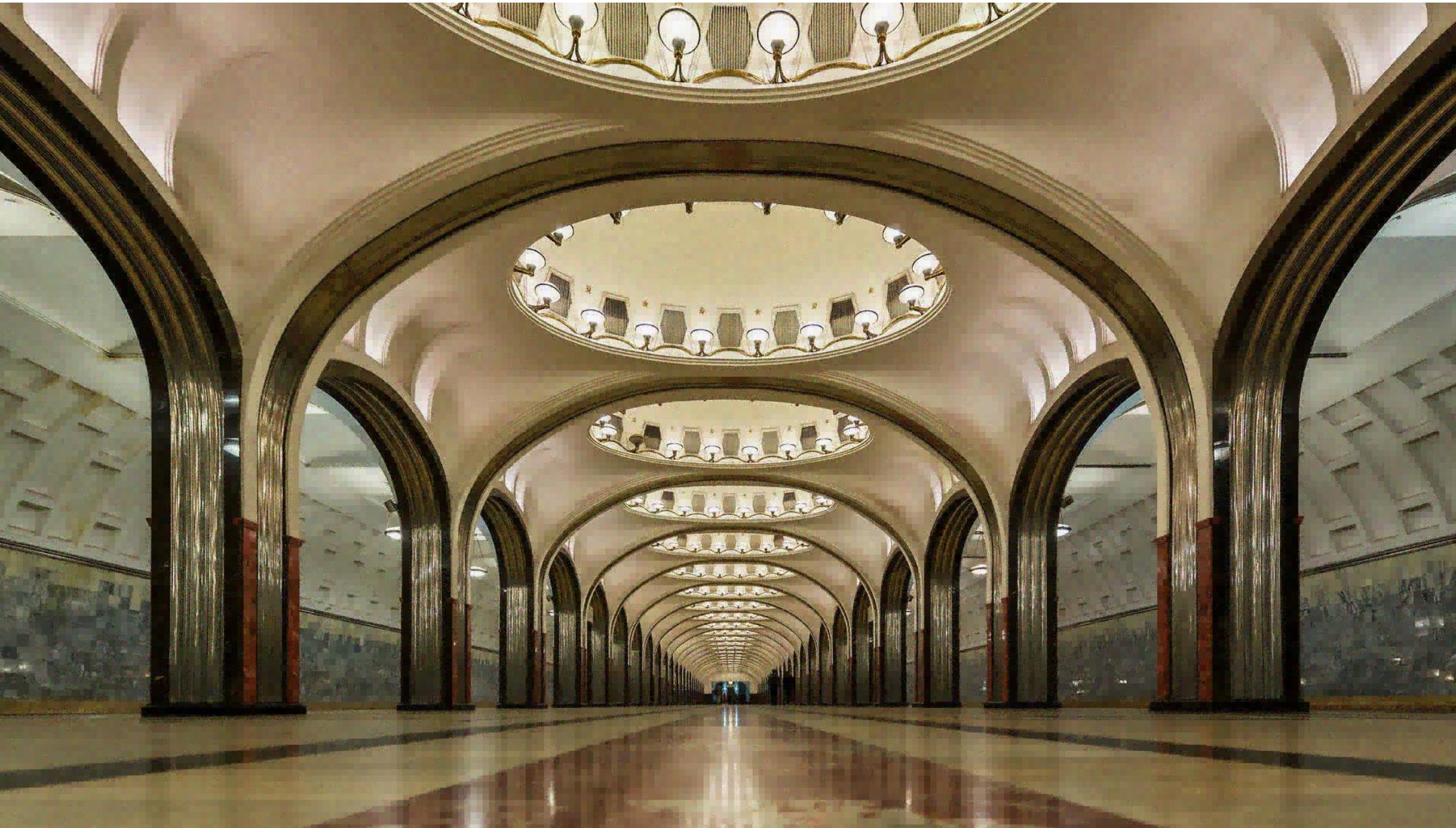


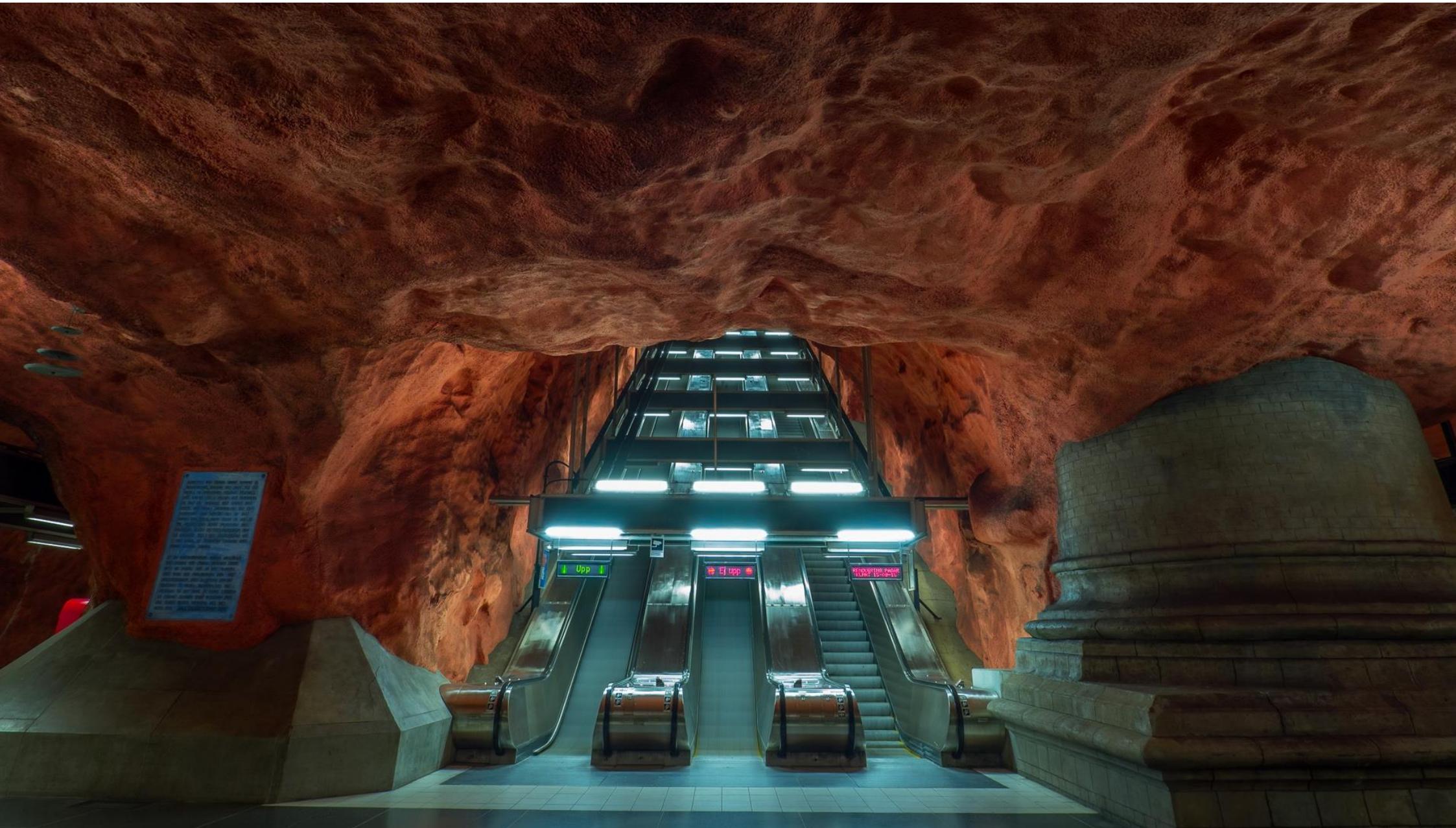


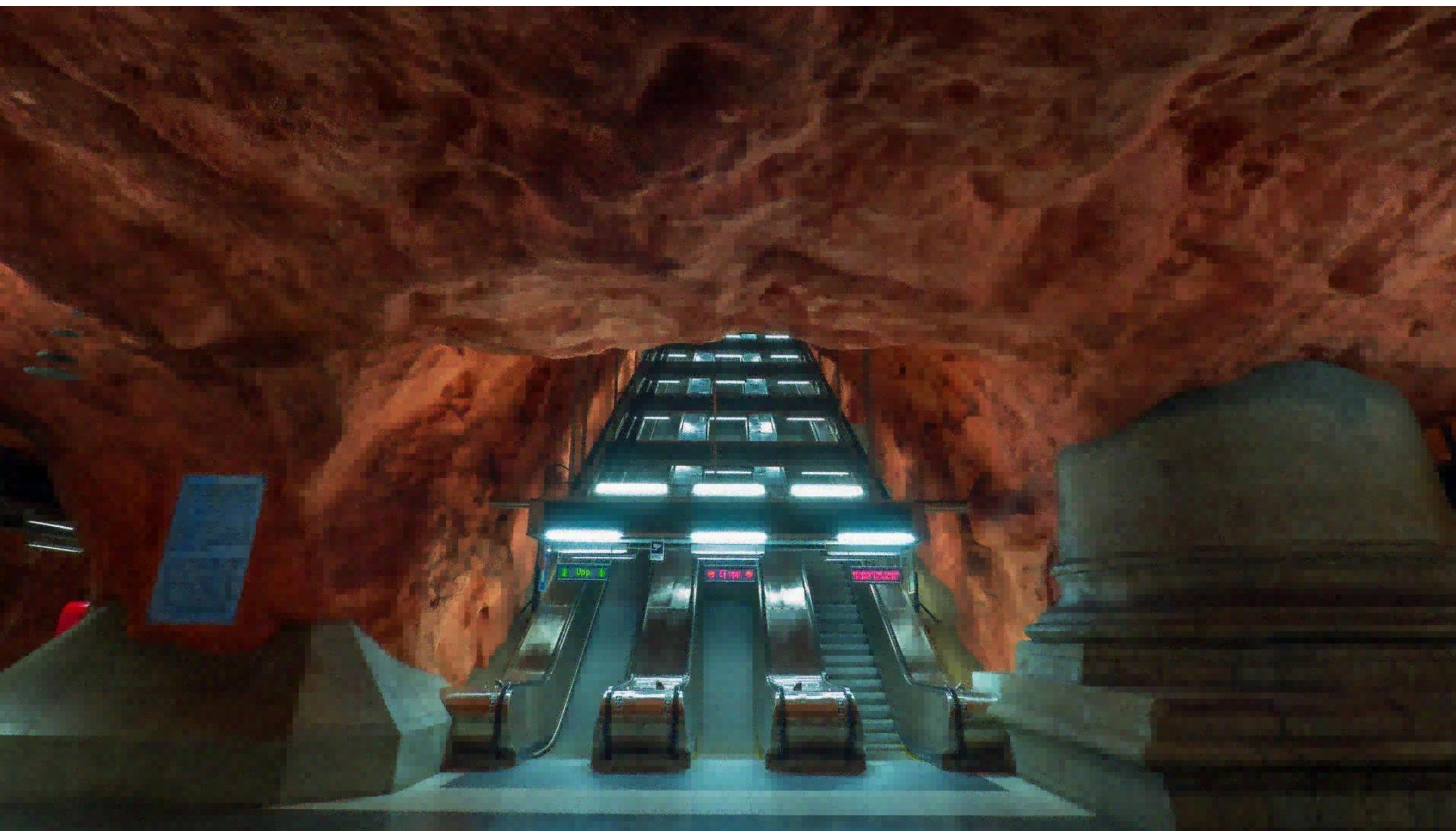


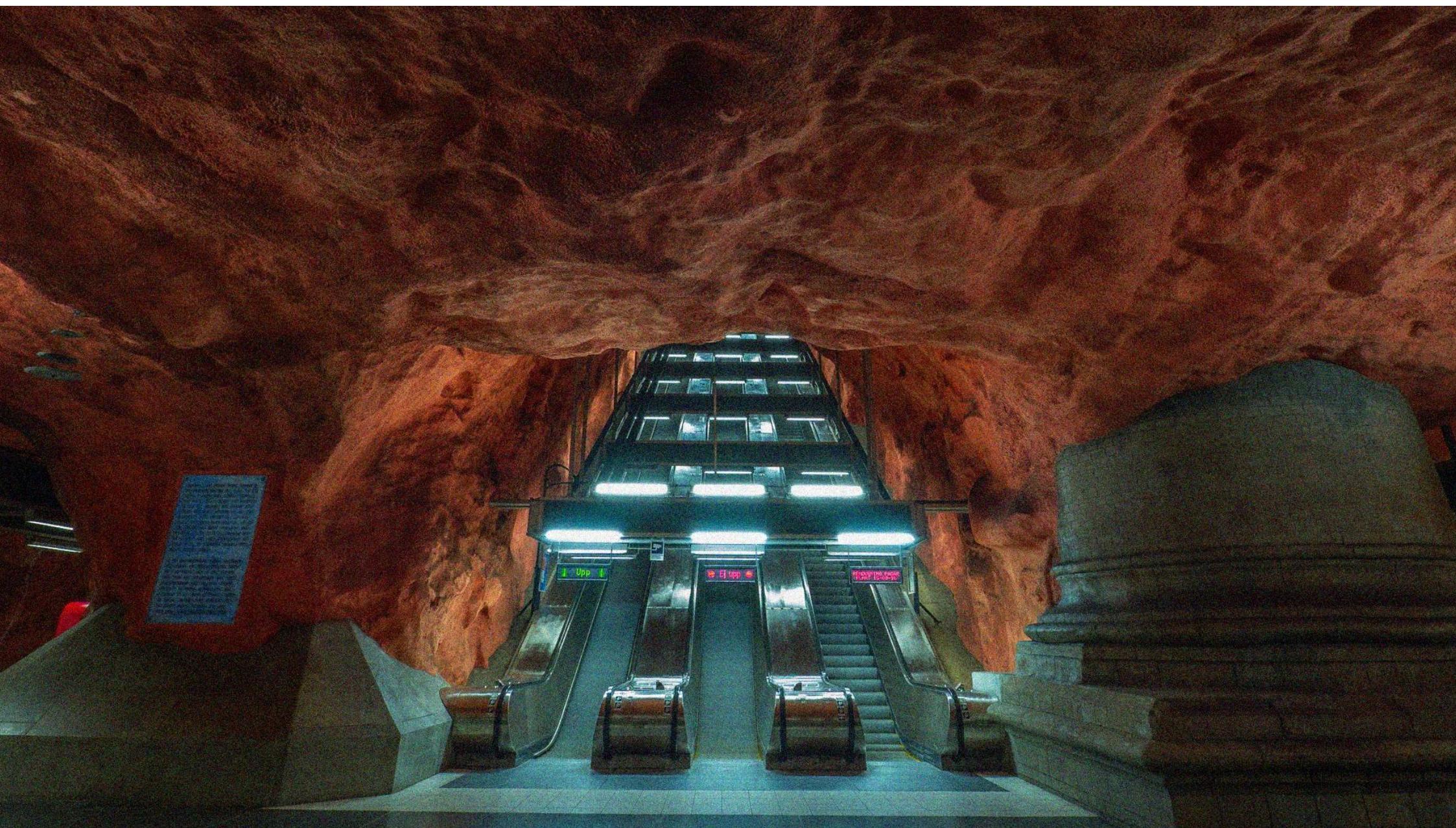












## Eliminarea de zgomot cu patch-uri suprapuse

În continuare voi prezenta câteva rezultate pentru funcția `test_denoising_overlapping.m`. Menționez că am folosit următorii hiperparametrii pentru fiecare din exemplele pe care le voi enumera:

```
parametri.patchSize = 8;
parametri.sparsity = 6;
parametri.totalNumberOfPatches = 1000;
parametri.dictionarySize = 256;
parametri.DL = 50;
parametri.noiseStandardDeviation = 20;
```

Fiecare exemplu se va întinde pe 3 pagini, prima pagina va reprezenta imaginea inițială, neperturbată, pagina a doua va reprezenta imaginea inițială perturbată cu zgomot alb gaussian, iar pagina a treia va reprezenta rezultatul scos de funcția `test_denoising_overlapping.m` pe imaginea de pe pagina a doua.



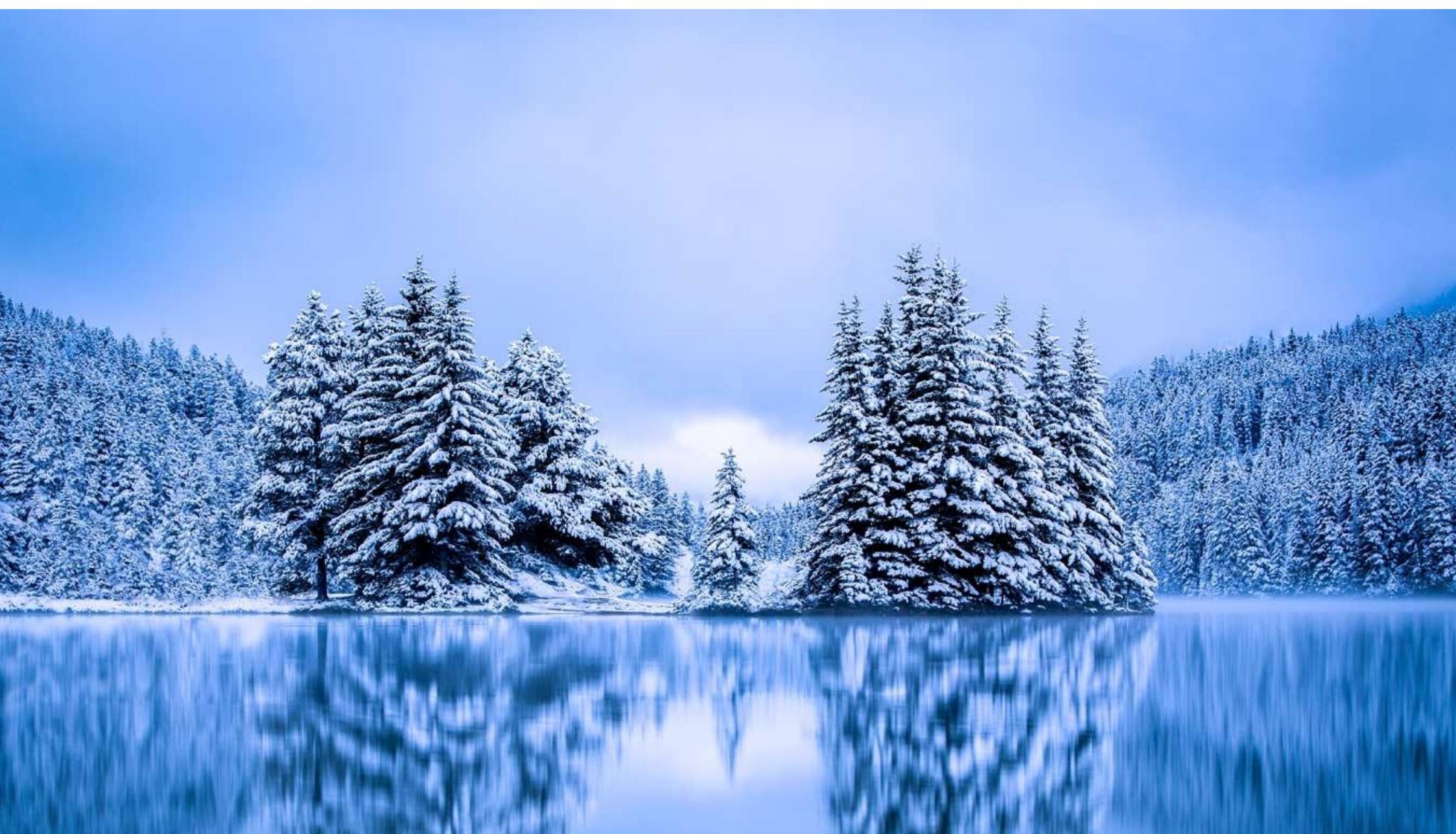


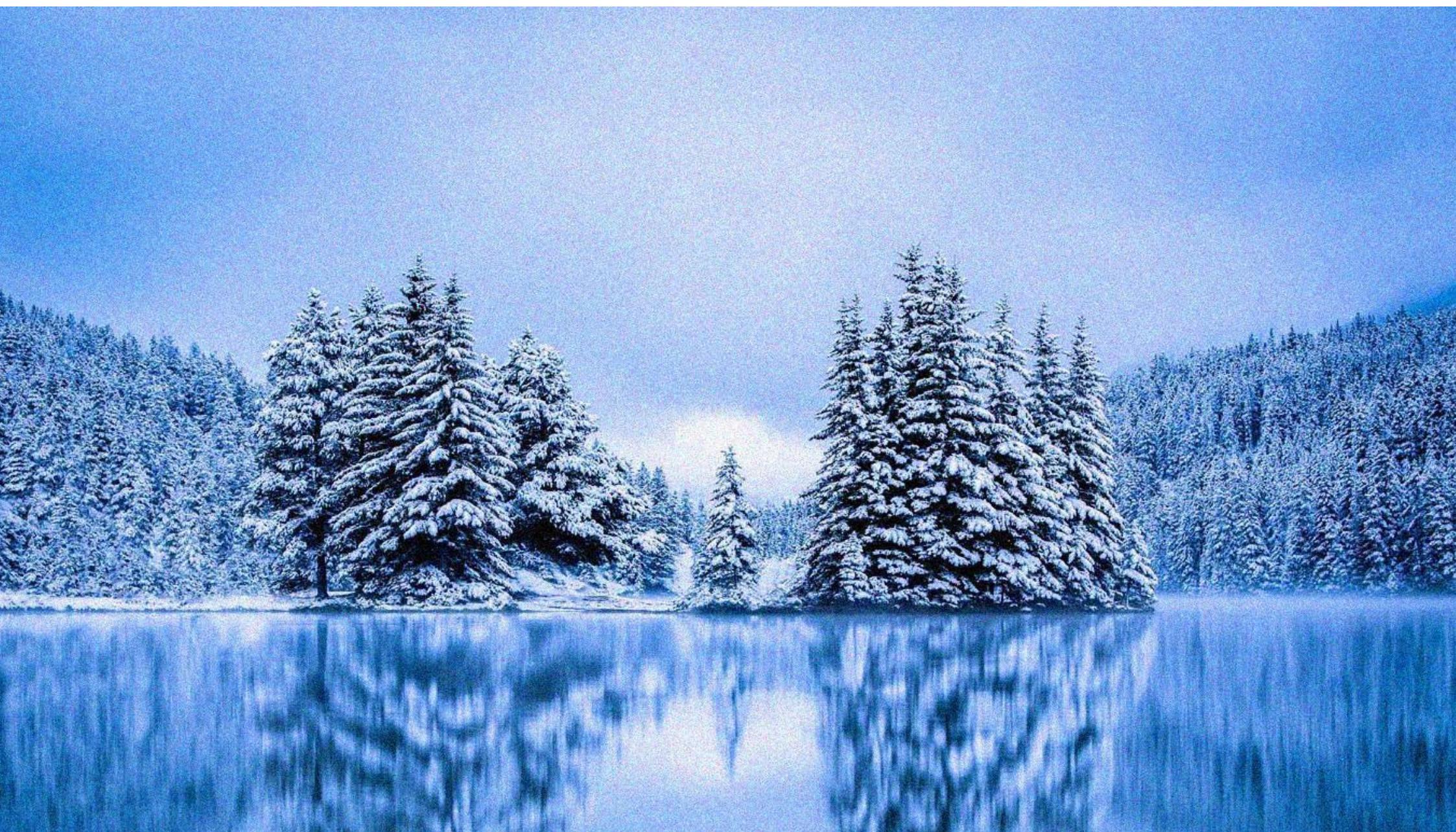


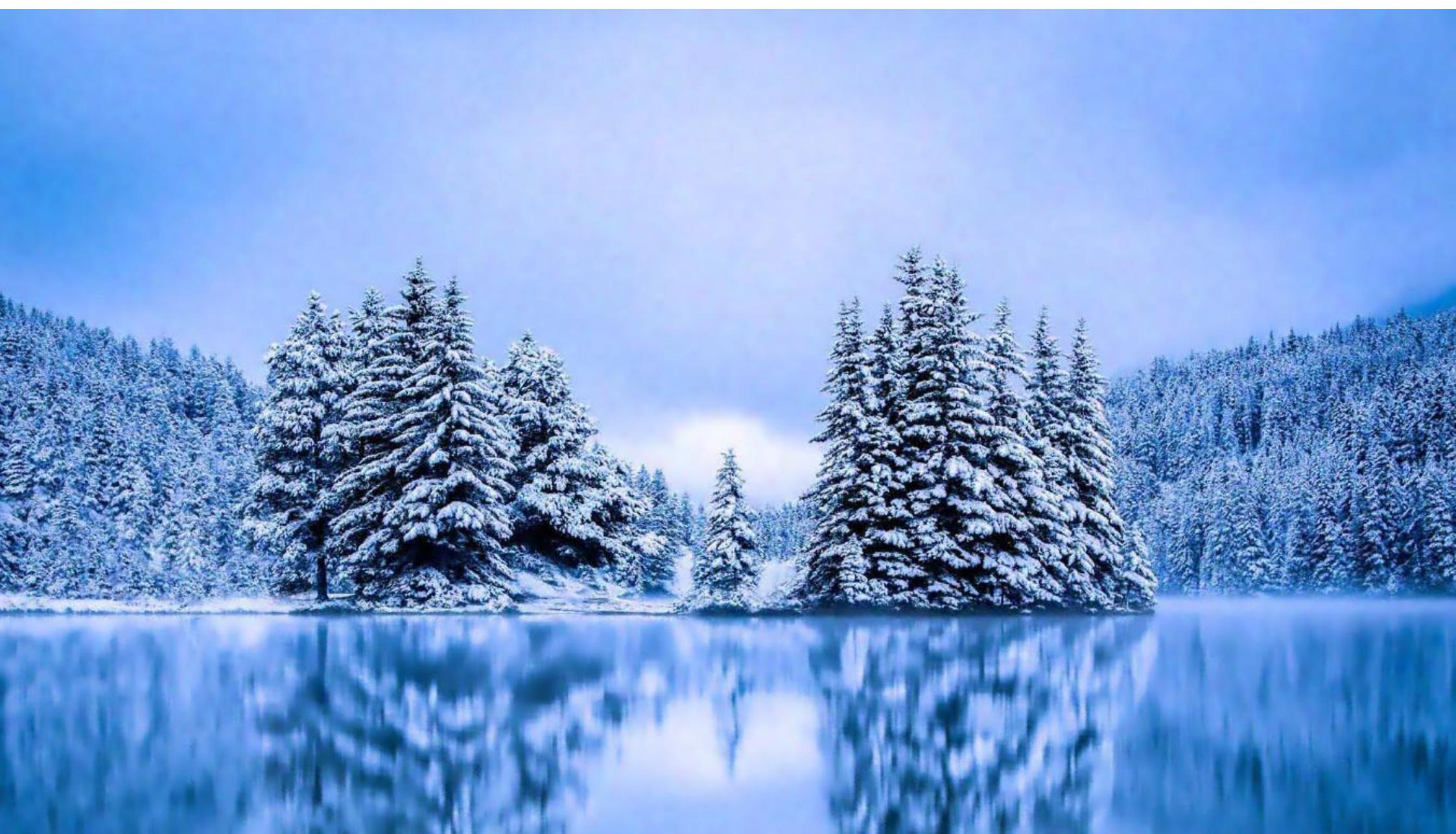












## Inpainting

În continuare voi prezenta câteva rezultate pentru funcția `test_inpainting.m`. Menționez că am folosit următorii hiperparametrii pentru fiecare din exemplele pe care le voi enumera:

```
parametri.patchSize = 8;
parametri.sparsity = 6;
parametri.totalNumberOfPatches = 1000;
parametri.dictionarySize = 256;
parametri.DL = 50;
parametri.missingDataRatio = 0.7;
```

Fiecare exemplu se va întinde pe 3 pagini, prima pagina va reprezenta imaginea inițială, neperturbată, pagina a doua va reprezenta imaginea inițială din care am eliminarea random a 70% din pixeli, iar pagina a treia va reprezenta rezultatul scos de funcția `test_inpainting.m` pe imaginea de pe pagina a două.



















## Analiza statisticilor psnr si ssim

Menționez ca cei mai populari indicatori sunt indicatori pentru cazul de fata sunt PSNR si SSIM. Ambii compara semnalul curățat cu originalul. PSNR indica diminuarea raportului dintre semnal si zgomot SSIM este mai apropiat de ce percep ochiul uman. In continuare voi face analiza pe următoarea imagine, si voi prezenta cei doi indicatori pentru cele 3 cazuri de rulare.



Pentru cazul 'test\_denoising\_distinct' se obțin  $\text{ipsnr} = 25.4237$  și  $\text{issim} = 0.6399$ . Pentru cazul 'test\_denoising\_overlapping' se obțin  $\text{ipsnr} = 28.6688$  și  $\text{issim} = 0.7683$ . Pentru cazul 'test\_inpainting' se obțin  $\text{ipsnr} = 29.5702$  și  $\text{issim} = 0.8893$  și următoarele 3 rezultate sub forma de imagine:



Pe următoarea imagine, să voi prezenta cei doi indicatori pentru cele 3 cazuri de rulare.



Pentru cazul 'test\_denoising\_distinct' se obțin ***ipsnr = 28.2699*** și ***issim = 0.7811***. Pentru cazul 'test\_denoising\_overlapping' se obțin ***ipsnr = 31.8699*** și ***issim = 0.8843***. Pentru cazul 'test\_inpainting' se obțin ***ipsnr = 34.2839*** și ***issim = 0.9533*** și următoarele 3 rezultate sub forma de imagine:



## Influenta hiperparametrilor

In continuare voi prezenta cateva rezultate pentru diferite valori ale hiperparametrilor pentru functia 'test\_denoising\_distinct.m' pentru următoarea poza:

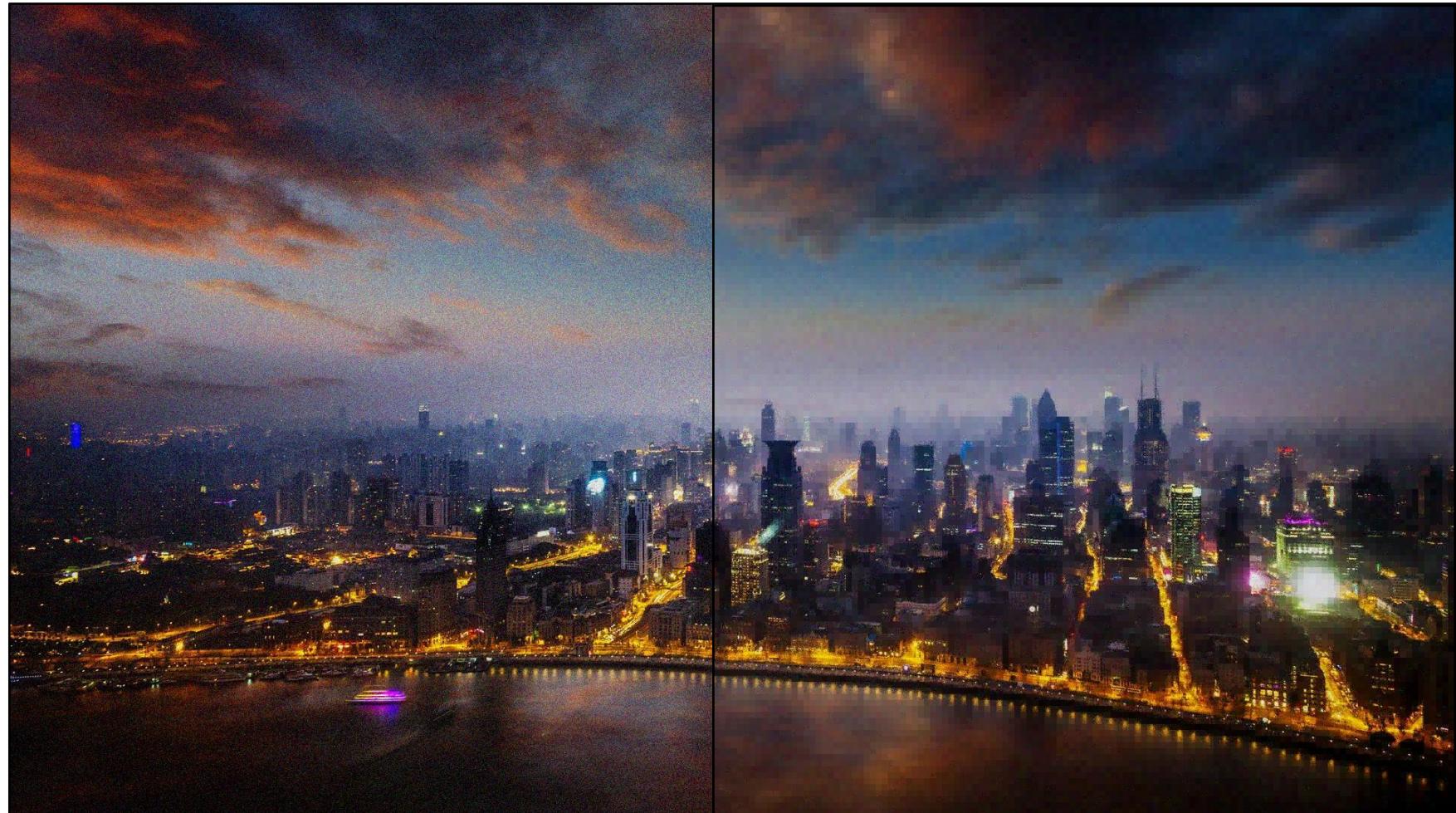


Parametrii de referinta sunt:

```
parametri.patchSize = 8;  
parametri.sparsity = 6;  
parametri.totalNumberOfPatches = 1000;  
parametri.dictionarySize = 256;  
parametri.DL = 50;  
parametri.noiseStandardDeviation = 20;
```

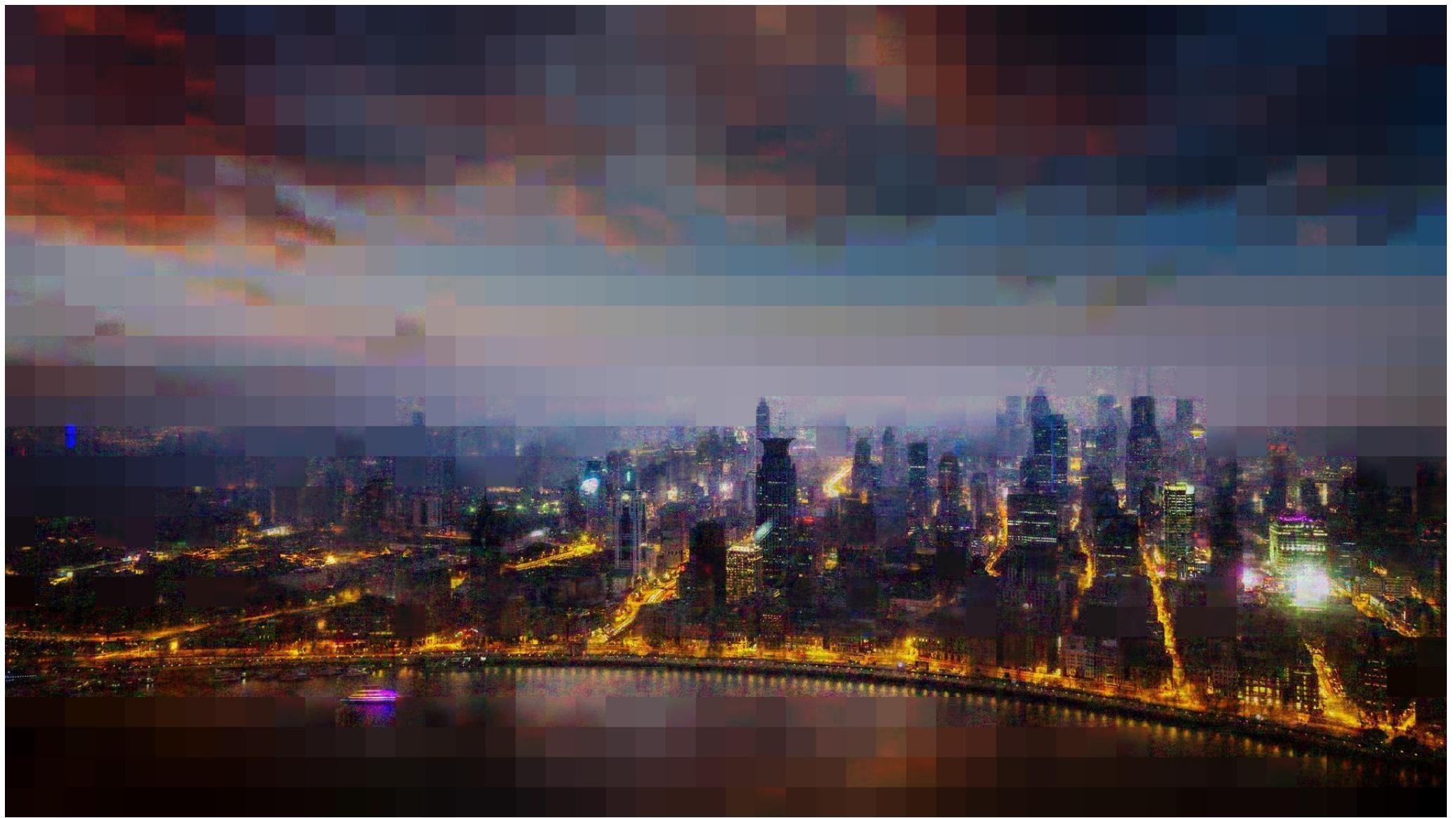
(ipsnr = 28.0045 issim = 0.7739)

Rezultatul pentru aceste parametrii este (a se observa diferenta dintre jumatarea stanga, care este perturbata si cea dreapta, rezultat al functiei studiate):



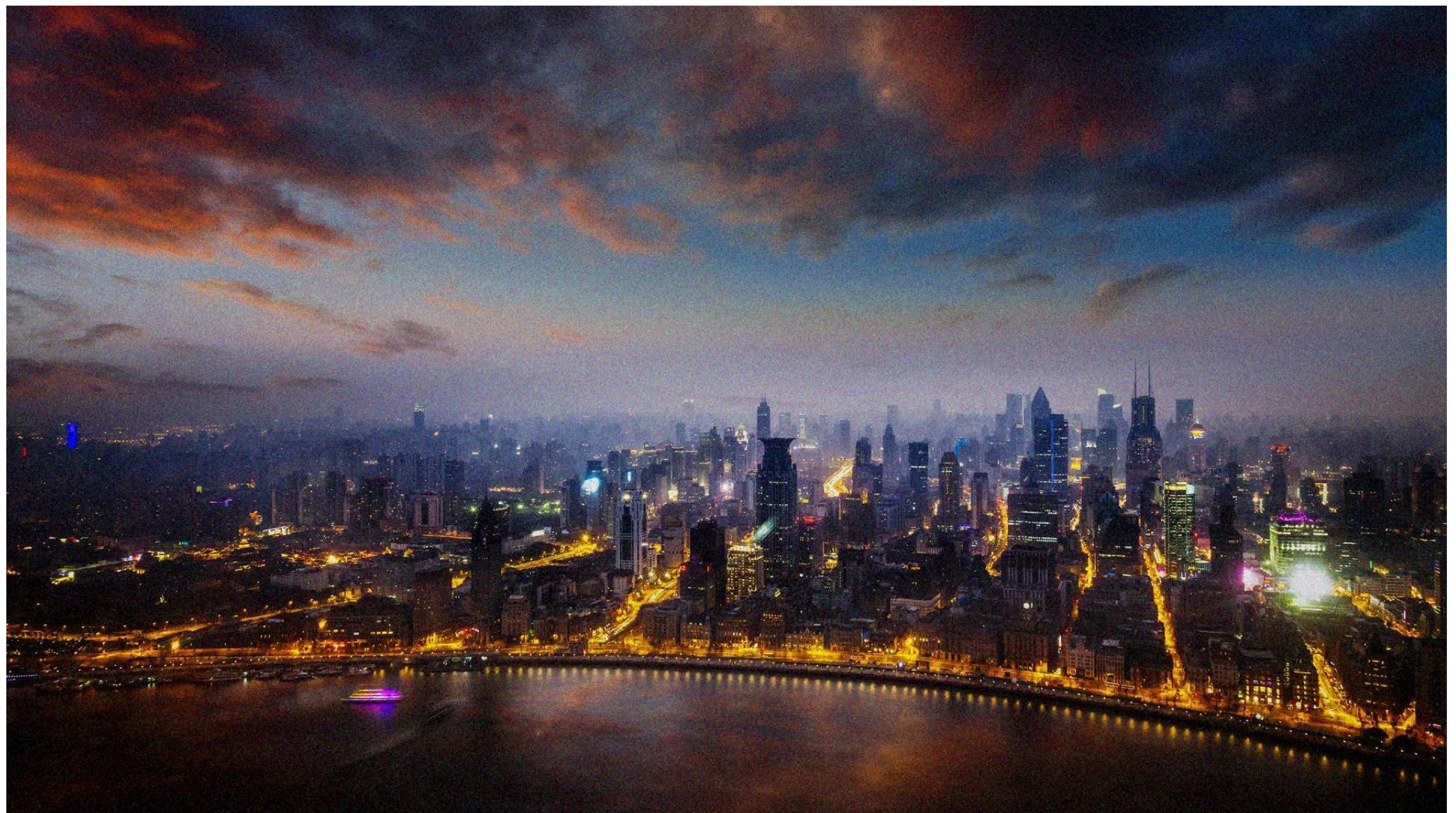
## VARIATIA DIMENSIUNII UNUI PATCH (PARAMETRI.PATCHSIZE)

parametri.patchSize = 40; (ipsnr = 25.8330 issim = 0.7356) vs initial (ipsnr = 28.0045 issim = 0.7739)



Se poate observa ca pentru un patchsize mare, efectul vizibil de blocking este foarte accentuat, iar zgomotul a fost redus considerabil. O alta observație vizibila este scăderea indicatorilor ipsnr si issim ceea ce indica o scădere a calității imaginii comparate cu cea in care se folosește un patchSize = 8.

```
parametri.patchSize = 2; (ipsnr = 24.4382 issim = 0.3749) vs initial (ipsnr = 28.0045 issim = 0.7739)
```



Se poate observa ca pentru un patchsize mic, efectul vizibil de blocking este redus, iar zgomotul încă există în imaginea rezultată în urma rulării algoritmului. O alta observație vizibila este scăderea dramatică a indicatorilor ipsnr și issim ceea ce indică o scădere a calității imaginii comparate cu cea în care se folosește un patchSize = 8. Acest lucru se întâmplă deoarece elementele din baza D vor avea dimensiunea de 2x2, deci foarte puțin context din imagine.

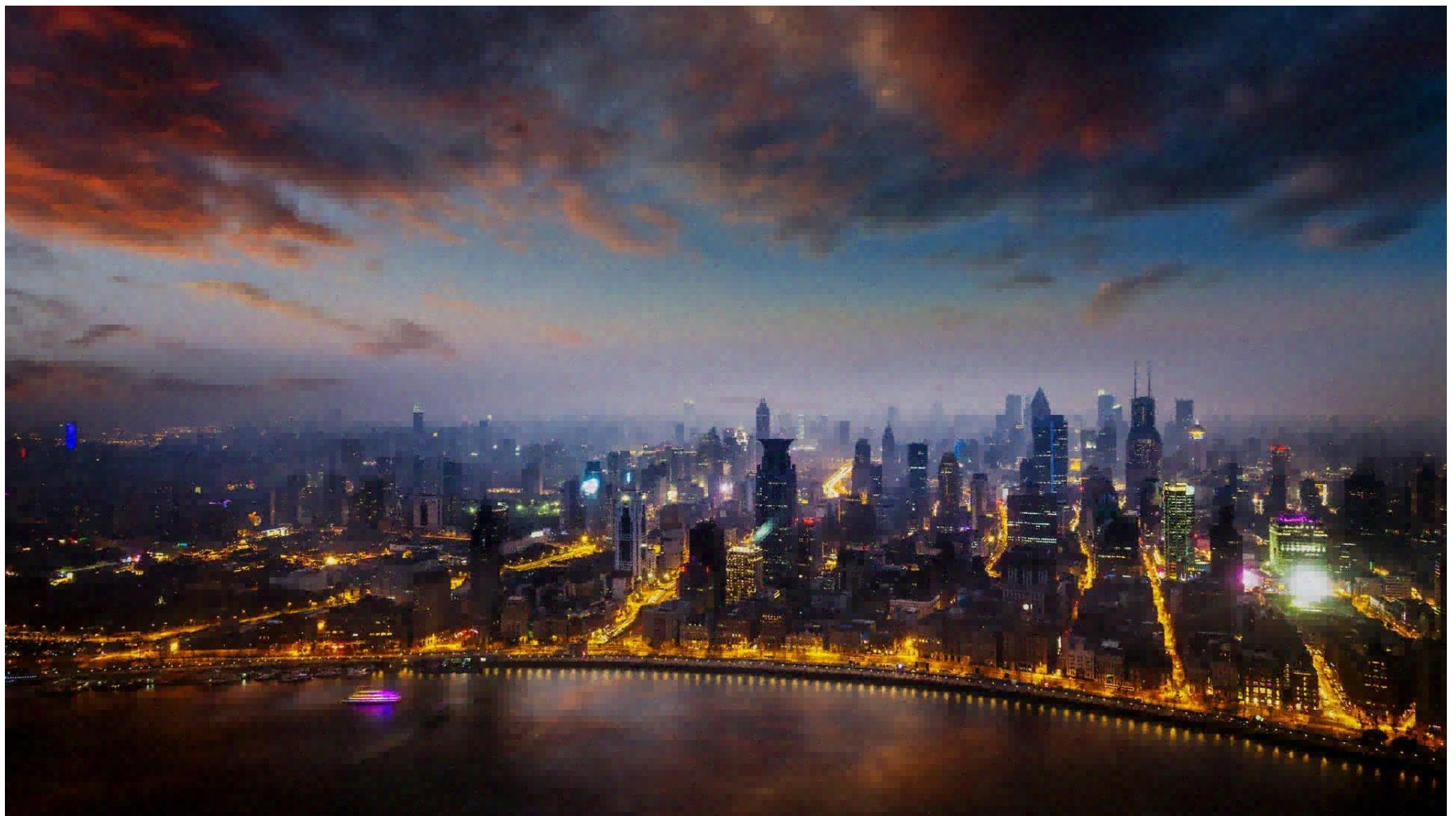
## VARIATIA NUMARUL DE PATCH-URI FOLOSITE LA INVATAREA DICTIONARULUI D (PARAMETRI.TOTALNUMBEROFPATCHES)

parametri.totalNumberOfPatches = 100; (ipsnr = 27.9910 issim = 0.7747) vs inițial (ipsnr = 28.0045 issim = 0.7739)



Pentru acest caz nu se poate observa o îmbunătățire majoră fata de cazul de referință pe motivul ca dicționarul D poate învăță din 100 de pachuri ceea ce învață pe cazul de referință din 1000 de pachuri, deci cele 900 de pachuri redundante se pot scrie ca o combinație liniară de cele 100 de pachuri pe care le folosim în acest caz.

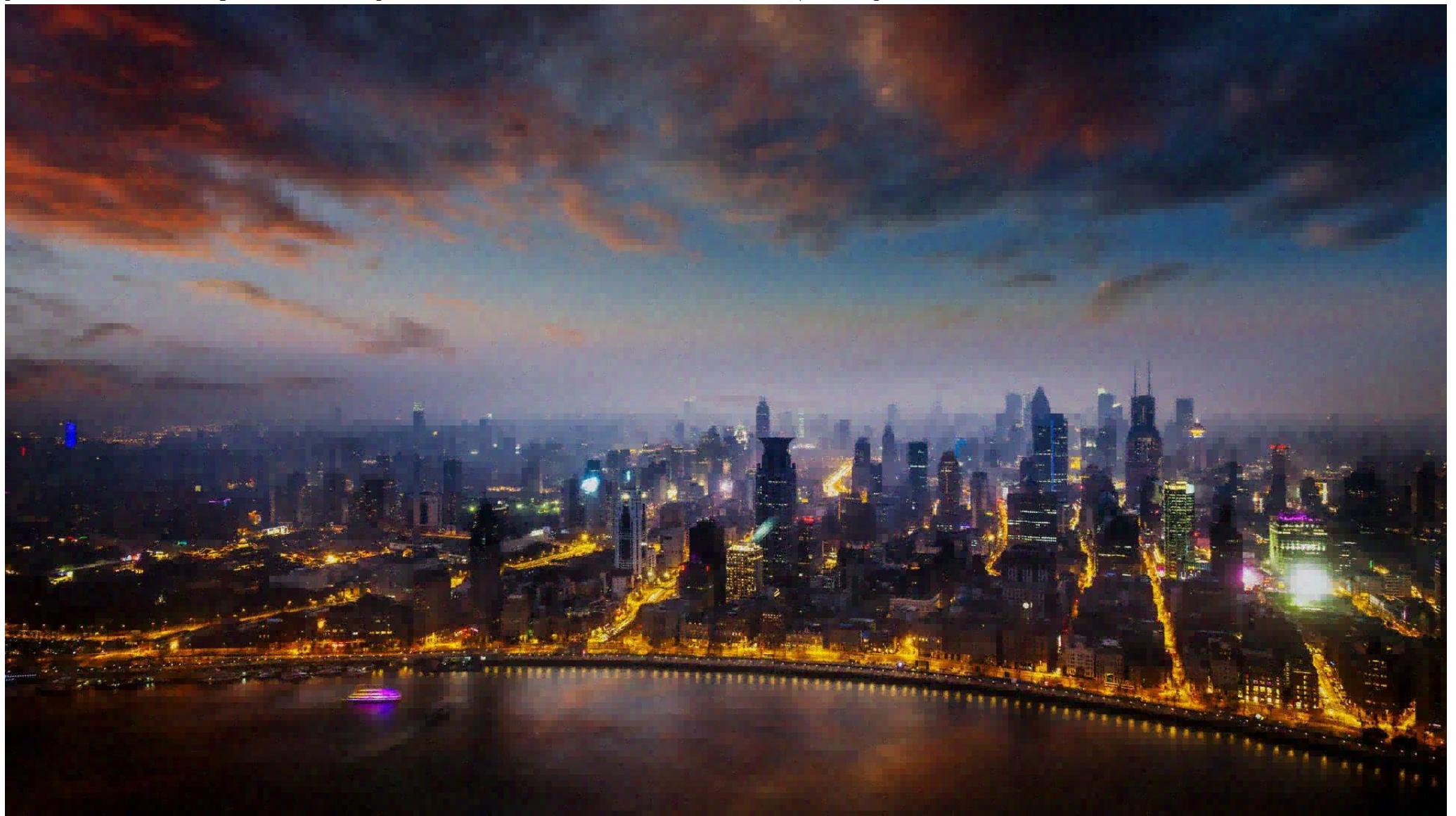
```
parametri.totalNumberOfPatches = 10000; (ipsnr = 28.6418 issim = 0.7868) vs initial (ipsnr = 28.0045 issim = 0.7739)
```



Pentru acest caz se poate observa o îmbunătățire minoră fata de cazul de referință. Acest lucru este perfect natural deoarece o învățare pe mai multe exemple (10000 pe cazul de fata in comparație cu 1000 pe cazul de referință) aduce o îmbunătățire in rezultat.

## VARIATIA DIMENSIUNII DICTIONARULUI D (PARAMETRI.DICTIONARYSIZE)

parametri.dictionarySize = 75; (ipsnr = 28.4103 issim = 0.7829) vs initial (ipsnr = 28.0045 issim = 0.7739)



Pentru acest caz se poate observa o scădere minoră a calității fata de cazul de referință. Acest lucru este perfect normal deoarece o învățare pe un dicționar mai sărac (cu mai puțini vectori) (75 pe cazul de fata în comparație cu 256 pe cazul de referință) aduce o scădere a calității.

```
parametri.dictionarySize = 1000; (ipsnr = 27.8568 issim = 0.7698) vs initial (ipsnr = 28.0045 issim = 0.7739)
```



Pentru acest caz se poate observa o scădere minoră a calității fata de cazul de referință în ciuda faptului că dicționarul pe care are loc învățarea este mai bogat. Acest lucru se întâmplă deoarece dicționarul în momentul de fata conține mulți vectori redundanți și este foarte departe de o baza.

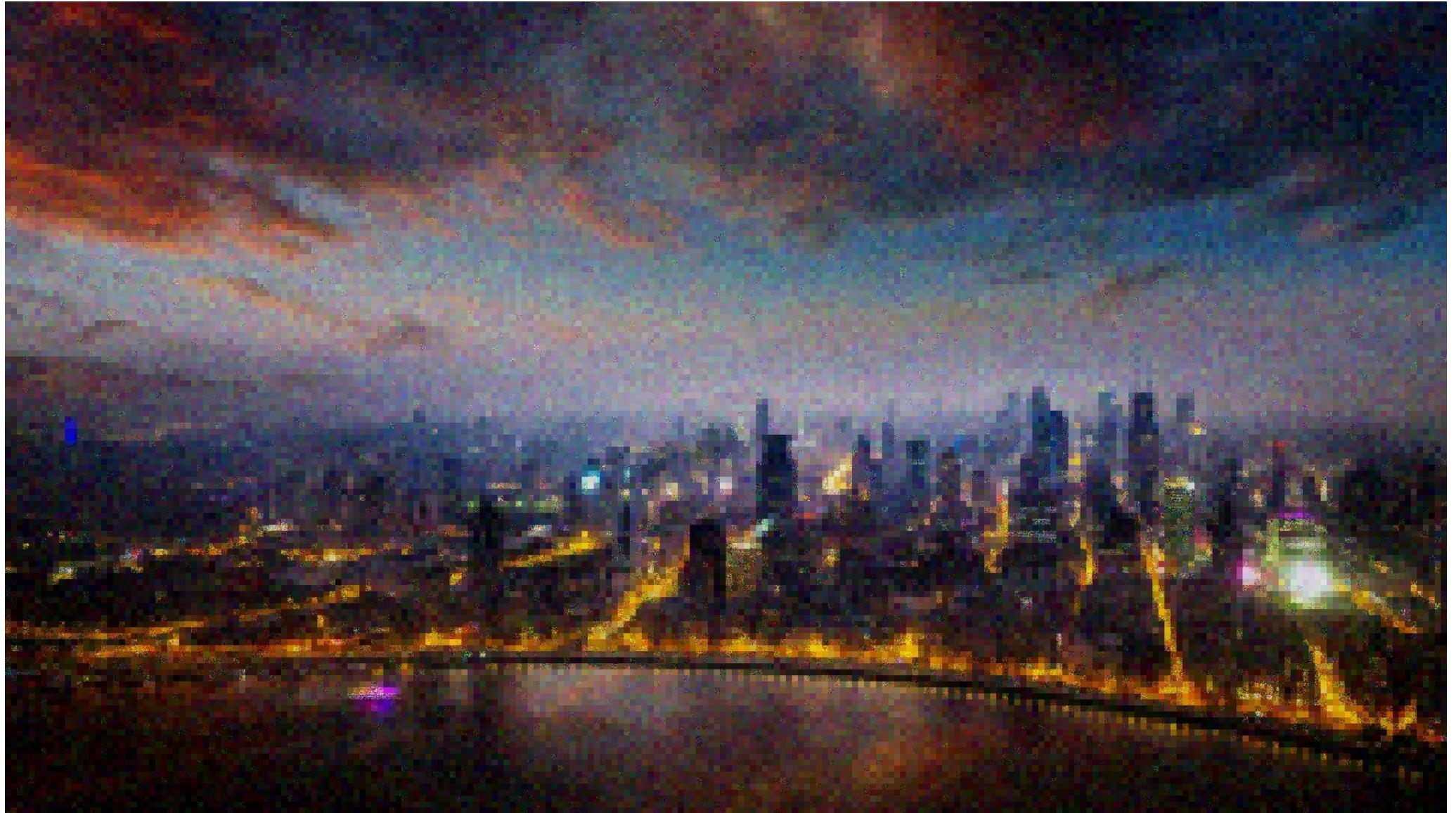
## VARIATIA DEVIATIEI STANDARD A ZGOMOTULUI (PARAMETRI.NOISESTANDARDDEVIATION)

parametri.noiseStandardDeviation = 5; (ipsnr = 36.0823 issim = 0.9186) vs inițial (ipsnr = 28.0045 issim = 0.7739)



Aceasta imagine seamănă foarte mult cu imaginea inițială. Acest lucru este de așteptat deoarece un zgomot cu deviație standard mică produce puțin damage în imaginea data, deci algoritmul are foarte puțin de corectat în imagine.

```
parametri.noiseStandardDeviation = 75; (ipsnr = 22.6823 issim = 0.5314) vs initial (ipsnr = 28.0045 issim = 0.7739)
```



Aceasta imagine este foarte perturbata. Acest lucru este de așteptat deoarece un zgomot cu deviație standard mare produce foarte mult damage în imaginea dată, deci algoritmul are foarte mult de corectat în imagine, iar imaginea pe care lucrează fiind destul de alterată este de așteptat să nu o poată reproduce la perfecție.

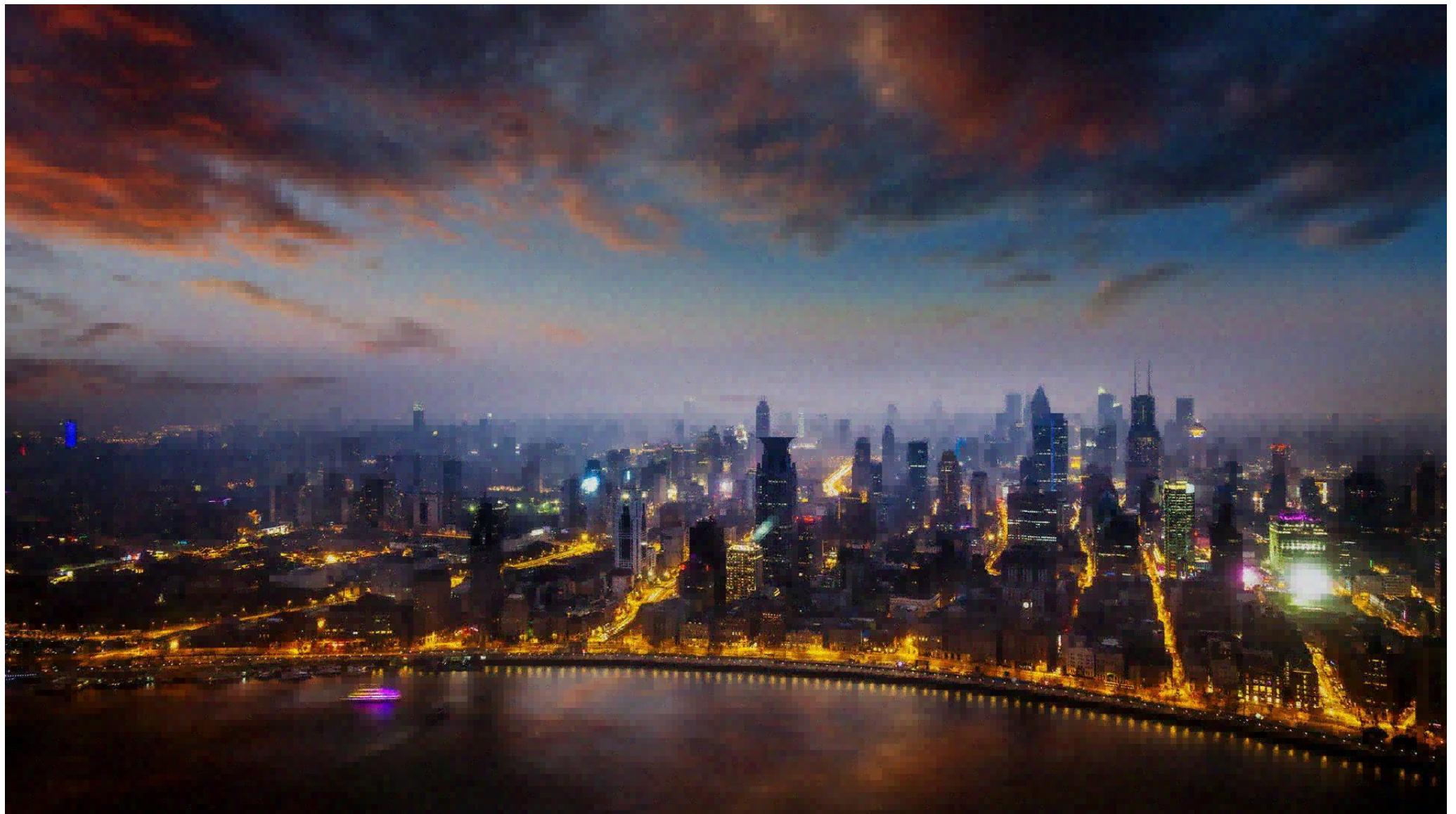
## VARIATIA NUMARULUI DE ITERATII (PARAMETRI.DL)

parametri.DL = 10; (ipsnr = 27.8560 issim = 0.7704) vs inițial (ipsnr = 28.0045 issim = 0.7739)



In acest caz avem o scădere ușoara a ciliații rezultatului fata de cazul de referință. Acest lucru este natural deoarece numărul de iterații nu este suficient pentru a sigura convergența algoritmului.

parametri.DL = 250; (ipsnr = 27.8609 issim = 0.7693) vs initial (ipsnr = 28.0045 issim = 0.7739)



In acest caz avem o scădere ușoara a ciliații rezultatului fata de cazul de referință în ciuda faptului că numărul de iterații este foarte mare. Acest lucru se întâmplă deoarece convergența este atinsă mult mai devreme.