

Odstranjevanje šuma

T. Ploetz, S. Roth. Benchmarking Denoising Algorithms with Real Photographs

J. Anaya, A. Barbu. RENOIR - A Dataset for Real Low-Light Image Noise Reduction

Projektna naloga z predmeta *Napredna Obdelava slik*

Luka Bašek, E5029527

Ivan Gradečak, E5031111

Maj, 2018.

PREGLED ČLANKA (25 točk)

Glavni članek je narejen iz razloga ker danes imamo zelo veliko različnih algoritmov in pristopa za odstranjevanje šuma in so metode za primerjalne analize z teg področja večinoma slabe. Zaradi tega so avtorji članka naredili metodo z katerom lažje in boljše primerjamo različne algoritme. Glavni cilj je narediti metodo z uporabo pravih slik in je ideja da se naredijo slike fotografirane z enakim ogledom na način da imamo šumne in brezšumne slike. Z razloga ker taki način ni enostaven, rabimo postopke za naknadno obdelavo in posebne regresijske modele z katerima odstranimo linearne odvisnosti med intenzivnostmi slik. Algoritmi kateri se uporabljajo za primerjavo so WNNM, K-SVD, EPPL, FoE, NCSR, BM3D, MLP in TNRD.

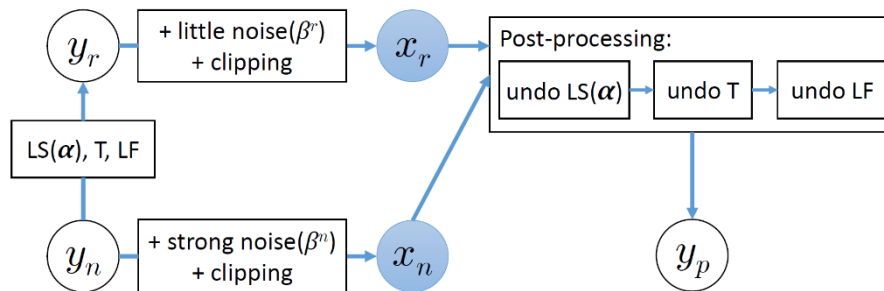
Nabor podatkov kateri se uporablja v členu se imenuje Darmstadt Noise Dataset (DND). Slike z tega nabora su narejene z štiri različne kamere v rasponu od navadnog pametnog telefona do profesionalne kamere z zamenljivim lečami. Kamere in telefoni z katerima su slike zajete su nasledni: Sony A7R, Olympus OMD E-M10 (Micro Four-Thirds), Sony RX100 IV in Nexus 6P. Nabor podatkov sestavljen je z 50 pari statičnih slik in je referenčna slika zajeta z osnovnom ISO stopnjom, a šumna slika je zajeta z večjim ISO in drugačim časom ekspozicije. Ostali parametri kamere su v obe slike enaki. Scene na slikami su tipične fotografije in izzivne strukture.

Po Wikipediji (Noise (signal processing), 2018), šum je v obdelavi signalov splošni izraz za neželene spremembe v signalu. Šum lahko pride v času zajemanja, shranjevanja, prenosa, obdelave ali konverzije signala in je v večinoma naključen in nima nobene koristne informacije. V členu je šumna slika x_n razložena kot dodajanje šuma brezšumni sliki in nato pa je odrezana intenzivnost katera upošteva nasičenost pikslov na senzorju kamere, a slika y_n je brezšumna. Šum lahko modeliramo kot Poisson-Gaussian šum katera moč je odvisna od brezšumnog intenziteta. V primeru njihovega nabora podatkov, imamo od vsake scene 2 slike, eno lahko prikazujemo kot šumno in je zapisana kot x_n , in druga bi rabila biti brezšumna pač takvo sliko ni mogoče narediti. Potem imamo in sliko z manjšim šumom kateru imenujemo x_r . Tisto sliko dobimo na način da zmanjšamo ISO vrednost.

Post-obdelava je narejena ker želimo odstraniti neželene transformacije med slikami. Glavni cilj je odstraniti tri stvari z obdelanih slik:

1. Majhne pod-pikselne premike katere se lahko zgodijo z vibracijami kamere (npr. zaradi zapiranja in otpiranja mehanskega zaklopa). Take majhne premike kamere lahko predstavimo kot globalne 2D translacije katere želimo odstraniti. V členu so ugotovili da je najboljša metoda Lucas-Kanade in da na tej način lahko dobe nazaj translacije in pri močnem šumu.

2. Napake katere se lahko zgodijo z majhnimi spremembami razsvetljenja prizora npr. zaradi premikajočih oblakov ali zaradi svetlobnega utripanja. Ker šum ima ničelno povprečje lahko ocenimo nizkofrekvenčni vzorec z nizkopasovnim filtriranjem.
3. Linearne spremembe intenziteta izhajajo iz dejstva, da se nit analogni dobiček niti čas izpostavljenosti ne morejo popolnoma nadzorovati.



Slika 1 - Proces oblikovanja slike, T. Ploetz, S. Roth (2017)

Na zgornjem diagramu lahko vidimo oblikovanje slike, z slike z nizkim ISO v sliko u visokim ISO vrednosti. Na začetku imamo brezšumne slike y_r in y_n katere so prestrukturirane z linearnimi spremembami slikovnih intenziteta (LS), majhnimi premikanjima kamere (T) in spreminjanjem svetlobe oz. preostalim nizko frekvenčnim vzorcima.

Pri izvajanju merjenja uspešnosti algoritmov slike su bile sesekljane na manjše dele zaradi tega ker so algoritmi za odklanjanje šuma zelo počasni da bi ih lahko izveli na velikim slikami. Vsaka slika je bila sesekljana na 20 manjših dela velikosti 512x512 pikslov in na koncu je ispalo 1000 sesekljanih slik. Rezultate merjenja uspešnosti algoritmov su prikazovali kot PSNR vrednosti katere so povprečane s pomočjo vseh sekljalnih slik in njihovih barvnih kanalov. Vsi rezultati su primerjani z post-obdelanim slikami v linearnem prostem (ang. raw) prostoru ali in sRGB prostoru. Tudi so merjene in SSIM vrednosti katere lahko pogledamo v dodatnim materijalima členka.

Na koncu se je iskazalo da algoritam BM3D je najboljši na odstranjanju šuma, nasledni pač je WNNM. Ostali algoritmi su ispali slabši kljub temu kaj je mjenje da so ostali pristopi odstranjevanja šuma prekosila BM3D algoritem. Taka primerjalna analiza je zelo podobna za primerjanje za uporabo v znanstvenoj. Slike z zmanjšanim šumom so podale slabše rezultate v sRGB prostoru v primerjavi z prosim (ang. raw) prostorom ker so sRGB slike v barvnem prostoru korelirane. Analiza ovega členka prikazuje da znanstvena praksa za ocenjevanje tehnik odstranjanja šuma ima precej omejen pomen za realistične nastavitve (T. Ploetz, S. Roth. 2017).

PREGLED PODROČJA (25 točk)

Članek *Benchmarking Denoising Algorithms with Real Photographs* (T. Ploetz, S. Roth. 2017) uporablja 8 algoritmov za odstranjevanje šuma kateri su narejeni od strane drugih avtorja kaj pomeni da je vsaki algoritem podporen z znanstvenim člankom. Uporabljeni algoritmi za odstranjevanje šuma su nasledni:

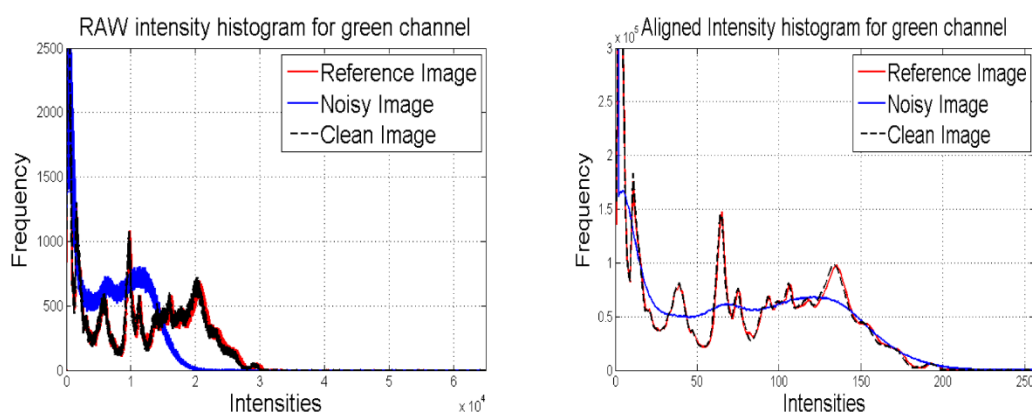
1. **WNNM (Weighted Nuclear Norm Minimization)** – Algoritem ki se temelji na minimizacija jedrskih norm (ang. nuclear norm minimization - NNM) kda imamo singularne vrednosti z različnim utežima. Ena od glavnih stvari algoritma je uporaba približka matrike nizkog ranka kateri se tudi uporablja in v računalniškem vidu in strojnem učenju. Prema členu z WNNM dobimo boljše PSNR rezultate in se sahrane zelo boljše lokalne strukture slike in imamo manjše vizualnih artefaktov (S. Gu, L. Zhang, W. Zuo, X. Feng, 2014).
2. **K-SVD** – Ta algoritem pospoševa K-Means postopek klasteriziranja (ang. clustering) zaradi prilagajanja slovarja za doseganje potrebne signalne predstavitve. Tudi K-Means algoritem je iterativni postopek kateri se večinoma uporablja z optimalnim šifrant (ang. codebook) za problem vektorske kvantizacije. K-Means uporablja K Mean izračune za ovrednotenje šifranta, K-SVD dobi posodobljeni slovar s postopki K SVD, pri čemer vsak proizvaja en stolpec. Ena od glavnih stvari zvezi K-SVD algoritma je učenje slovarja in možnost da se lahko prilagodi in deluje v konjugaciji z ostalim algoritmima. Taki dizajn slovarja lahko zelo uspešno zamenj popularne predstavitvene metode katere se uporabljajo za izboljšanje, stiskanje in ostale stvari v obdelavi slik (M. Aharon, M. Elad, A. Bruckstein, 2005).
3. **EPLL (Expected Patch Log Likelihood)** – Okolje ki omogoča celotno obnavljanje slike z uporabo obliža (ang. patch) na podlaki katerega lahko izračunamo MAP ali približno oceno MAP vrednosti. Modeli kateri uporabljajo obliž (ang. patch) lažje učijo in delajo neg modeli z celim slikami. EPLL izboljšava rezultate celotne slike v primerjavi z enostavnim povprečanjem obližka kateri se uporabljajo v večinoma metod. Tudi se uporablja in Gaussova mešanica, ki pred tem presenetljivo dobro opravlja slike katere rabijo odstranjevanje šuma, zameglitve ali nejasnosti (D. Zoran, Y. Weiss, 2011).
4. **FoE (Fields of Experts)** – Ovi algoritem se lahko uporablja za obdelavo statičnih slik in različne potrebe v računalniškem vidu. FoE omogoča praktično metodo za učenje Markovih modela naključnega polja visokega reda (ang. Markov random field - MRF) z funkcijami katere se razširijo nad velika področja pikslov. V tem algoritmu vsi parametri vključno z linearnimi filtrimi navčeni na učnoj množici. Članek prikazuje izvajanje algoritma v primeru odstranjevanja šuma in odstranjevanja dodatnih slik na drugoj sliki (S. Roth, M. J. Black, 2009).

5. **NCSR (Nonlocally Centralized Sparse Representation)** – Algoritem predstavlja standardni redki reprezentativni model kateri se prikazal kot dobra rešitev za obnovitev slik v slučaju šuma ali zamegljenosti. Bayesova interpretacija NCSR modela je poddana in to modelu daje iterativno ponderirano izvajanje. Tudi se uporablja in efikasna iterativna funkcija za zmanjšanje katera uspešno rešuje l_1 -reguliran NCSR minimizacijski problem. NCSR protop omogoča zelo konkurenčna uspešnost v usporedbi z ostalim algoritmima za odstranjenje šuma in je zelo dober po izvedbi (W. Donga, L. Zhangb, G. Shia, X. Li, 2012).
6. **BM3D** – Strategija katera se temelji na izboljšanju porazdelitve skupina 2D fragmenta slike v 3D podatkovnih nizo kateri se imenuje skupine. Ideja se temelji na tri zaporedna koraka: 3D transformacije nad skupinama, krčenje spektra transformacije, inverzne 3D transformacije. Z zmanjšanjem šuma lahko razkrimo in najmanjše detalje v skupinama in zaradi tega ker se skupine prekrivajo za vsaki piksel lahko dobimo več mogočih kombinacija. Algoritem uporablja in Wiener filter kateri je preobdelan specialno za težave BM3D algoritma. Članek v katerem je razložen BM3D algoritem je predstavljen kot zelo dobra rešitev za odkljanjanje šuma v smislu izvezbe. Tudi ima dobro razmerje med signalom in šumom in subjektivno vizualno kakovostjo katera se iskazala in v našoj poustvaritevi eksperimenta (K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, 2007).
7. **MLP (Multi Layer Perceptron)** – Ovi pristop je drugači kot ostali ker uporablja večplastne perceptrone oz. nevronske mreže in se algoritem predstavlja kot rešitev katera lahko nasprotuje BM3D algoritmu. Način na kateri deluje algoritem je da pospravlja šumne obliže (ang. patch) na brezšumne. Da bi taka nevronska mreža naredila potrebno delo rabimo dovolj skritih plasti in dovolj skritih enot. Tudi velikost šumnog obliža rabi biti kot brezšumni obliž. Učni primeri se generirajo u času učenja na način da kvarijo brezšumne obliže z šumom. Zaradi velike učne množice tista nevronska mreža se uči na grafični procesni enoti (ang. Graphics Processing Units - GPU) (H. C. Burger, C. J. Schuler, S. Harmeling, 2012).
8. **TNRD (Trainable Nonlinear Reaction Diffusion)** – TNDR algoritem se uporablja za različne postopke obnove slik z vključitvijo ustrezne reakcijske sile. Članek deluje na tri načina z uporabo Gaussove metode odstranjanja šuma, ene slike z super razločljivosti in JPEG deblokiranjem. Navčeni model sahranjuje sktrukturalno enostavnost difuzijskog modela z uporabo majhnog števila korakov in je potem pospotek zelo efikasen. Tudi algoritem lahko deluje in na način da imamo vzporedno izvajanje na več računalnika in z uporabo grafičnih procesnih enot (ang. Graphics Processing Units - GPU) hitrost algoritma zelo zboljšamo. (Y. Chen, T. Pock, 2016).

POUSTVARITEV EKSPERIMENTA (50 točk)

Nabor podatkov kateri smo uporabili v naši poustvaritvi eksperimenta je drugač kot v glavnem členu. Uporabljeni nabor imenuje se RENOIR in je javno dostopni za testiranje in primerjalne analize. RENOIR je skupina od 500 slik in 120 različnih scena katere su zajete z dvema kamerami in enim pametnim telefonom: Cannon T3i, Cannon S90 in Xiaomi MI3. Svaka scena se sastoji od tri slike, ena referenčna katera je brezšumna, ena šumna in ena tudi brezšumna katera je zajeta po zajemanju šumne. Na tej način imamo sendvič slik prema redu zajemanja kje je v sredi šumna slika kaj je zelo pomembno za računanje PSNR i ostalih vrednosti. Nabor podatkov je zelo velik ker su vse slike ostavljene v originalnoj ločljivosti kamere in je vsaka slika okoli 10 megapiksela. Tudi je nabor zelo dober za raziskanje šuma v slikama z nizkom svetlobom, učenje in testiranje algoritmov za odstranjevanje šuma in ostale uporabe. Ker slike imajo velik razpon šuma avtorji nabora priporočajo pri odstranjevanju šuma da se notarnji parametri algoritma pri vsaki sliki prilagodijo (Adrian Barbu's Research, 2018).

Ker je dostop do nabora podatkov glavnog članka omejen, zbrali smo drugi nabor kateri se imenuje RENOIR. V tem naboru imamo tri tipa slik prema tipu kamere in tipu formata slik. Za našo poustvaritev eksperimenta vzeli smo slike zajete z kamero Canon T3i in poravnane (ang. aligned) slike v formatu BMP. Razlog zakaj smo se odločila za poravnane slike nad RAW slikami je ker avtorji priporočajo uporabo tisteg nabora slik ker imajo poravnano svetlobo.



Slika 2 - Primer razlike zelene barve pri RAW in poravnani sliki, J. Anaya, A. Barbu (2014)

Ker su slike zelo velike in svaka ima okoli 50 MB, če bi delali odstranjevanje šuma nad celim slikami vzelo bi nam zelo veliko časa. Tudi ker so algoritmi počasni in nimamo dostop do nekih boljših računalnika testiranje smo omejili na manjše število sesekljanih slik. V členu T. Ploetz, S. Roth (2017) primerjalna analiza je narejena nad vse skup 1000 sesekljanih slik kje je vsaka velikosti 512 x 512 pikslov. Vsaka celotna slika je razrezana na 20 manjših delova na način da se prekrivajo okoli 10%.

V našem primeru imamo 40 serija slik kje je od vsake originalne slike vzeta sesekljanja slika velikosti 512 x 512. Na tej način imamo 40 scena in vsaka scena se sestoji od tri slike: referenčne, šumne in čiste slike.

Canon T3i			
	Referenčna slika	Šumna slika	Čista slika
ISO	100	3200 ali 6400	100
Trajanje zajemanja	auto	auto	auto

Tabela 1 - ISO in trajanje zajemanja vsakog tipa slike zajete z kamero Canon T3i, J. Anaya, A. Barbu (2014)

Slike su zajete po principu sendviča ker na tej način dobimo manjšo napako če je v času zajemanja prišlo do manjših sprememba svetlobe, ali gibanja kamere. Zaradi tega računanje PSNR in SSIM vrednosti je mal drugače neg običajno.

Kda računamo Peak Signal-to-Noise Ratio (PSNR) in Structural Similarity Index (SSIM) uporabljamo normalne formule kak in običajeno. Razlika je edino v tem kaj v našem primeru uporabljamo tri slike (referenčno, šumno in čisto) namesto dve kak je opisano v vsim opisima tistih formula. Tri slike uporabljamo zaradi principa sendviča ker hočemo zmanjšati majhne razlike med časom zajemanja. Zaradi tega združimo referenčno in čisto sliko in naredimo njihovo povprečje. Potem vzamemo to povprečno sliko in z šumno računamo PSNR in SSIM. Takav način izdelave je primeren zaradi značilnosti RENOIR nabora podatkov.

PSNR se računa po nasledni formuli:

$$PSNR = 10 * \log_{10}(peakval^2 / MSE)$$

Vrednost *peakval* se lahko postavi od strane uporabnika ali programske kode. Tista vrednost lahko bo razpon v katerem se vrednosti piksla v sliki predstavijo in v normalnem slučaju gre od 0 do 255. Običajemo *peakval* je maksimalna vrednost svih pikslov v sliki. *MSE* pomeni povprečno kvadratno napako (The MathWorks, 2018).

SSIM je metoda za merjenje kakovosti slik. Kot vhod v algoritem se pošiljajo dve slike velikosti N x N in se računa njihova podobnost po naslednji formuli:

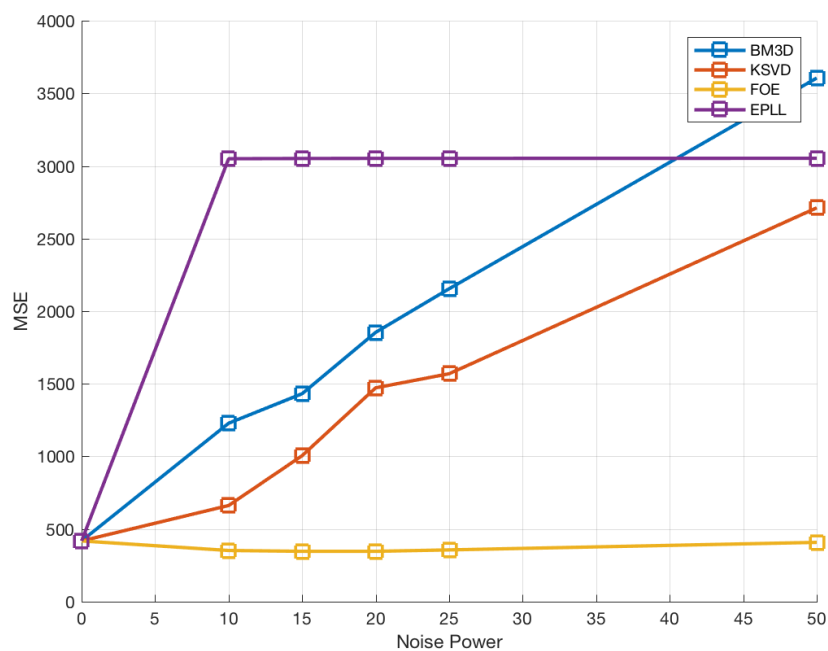
$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_y^2 + \mu_x^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Kje je: μ_x povprečje slike x , μ_y povprečje slike y , σ_x^2 varianca slike x , σ_y^2 varianca slike y , σ_{xy} kovarianca od x in y slike. SSIM se lahko naredi nad barvno ali kromatsko sliko. Vrednost katera se dobi je decimalna v razponu od -1 do 1. Vrednost 1 se edino zgodi če so slike identične. Tudi avtorji priporočajo da se algoritem uporablja nad manjšim oknom ker je kompleksnost računanja velika (Wikipedia, 2018).

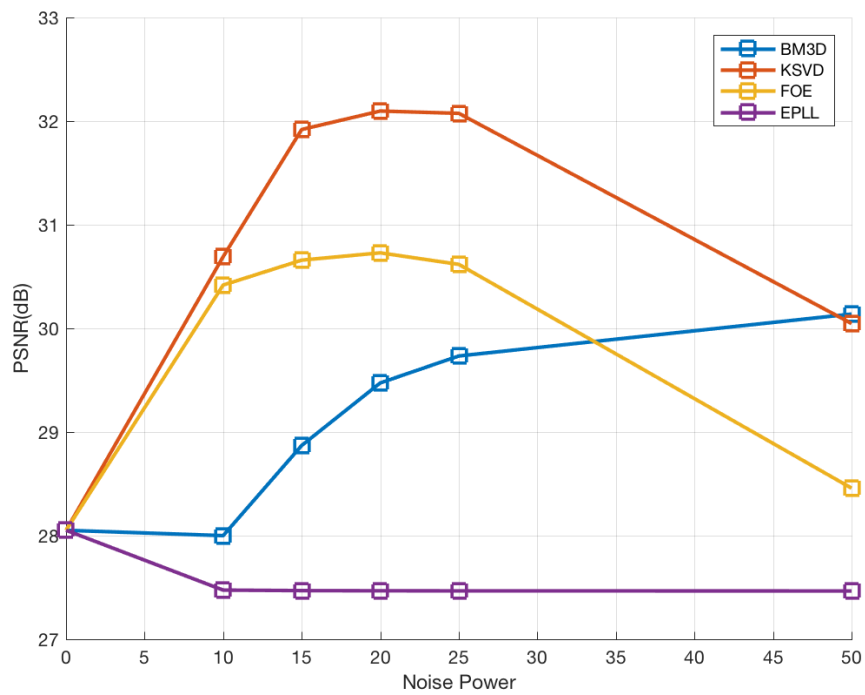
Vsa koda katero smo uporabljali za računanje vrednosti je napisana v MATLAB računalniškem okolju. Algoritmi katere smo uporabljali so prevzeti od členkov napisanih zaradi njihove podpore in smo jih spreminjali da bojo čim boljše postavljeni za našo poustvaritev eksperimenta.

Vrednotenje kakovosti slike pred i po obdelavi smo naredili pomočjo naslednjih metrik: PSNR, SSIM in MSE. Poleg zgornjih mertik smo dodal še faktor vrednotenja, čas.

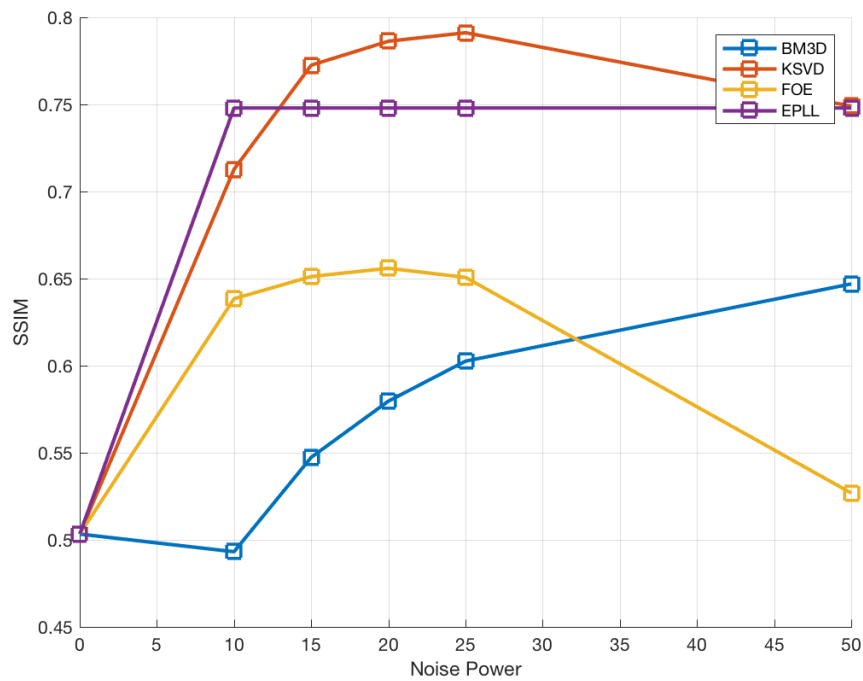
Pri merilo uspešnosti smo zagnali nabor od 20 slik, katere so bile velike 512x512 in smo jih testirali z štirimi algoritmi: BM3D, K-SVD, FOE, EPLL. Algoritme WNNM in NCSR smo rabili odstranit ker čas izvajanja bil zelo dolg. Algoritme smo testirali z vnaprej določenim vrednostima šuma σ : 10, 15, 20, 25 in 50.



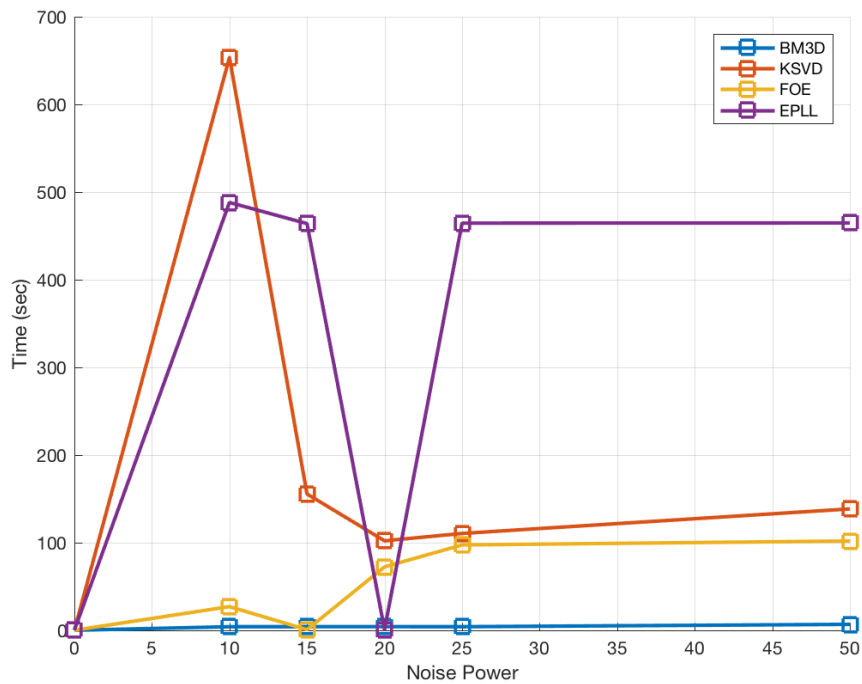
Slika 3 - Rezultati MSE metrike nad 20 slik za algoritme BM3D, KSVD, FOE in EPLL



Slika 4 - Rezultat PSNR metrike nad 20 slik za algoritme BM3D, KSVD, FOE in EPLL



Slika 5 - Rezultat SSIM metrike nad 20 slik za algoritme BM3D, KSVD, FOE in EPLL



Slika 6 – Poprečni delovni čas nad 20 slik za algoritme BM3D, KSVD, FOE in EPLL

MERA	ORIGINAL	BM ₃ D	KSVD	FOE	EPLL
MSE	417.799567286173	1228.31117992401	661.829725138346	352.073617172241	3049.91714998881
PSNR	28.05106489921	27.9992549689984	30.6917676369447	30.4178441563735	27.4736095924037
SSIM	0.503258650626292	0.493111697843747	0.712752615233474	0.638424256944697	0.747985197083297
TIME	-	4.03223262756157	653.550031363394	26.9738004735498	488.116793192663

Tabela 2 – Rezultati algoritmov BM3D, KSVD, FOE in EPLL za $\sigma(\sigma) = 10$

MERA	ORIGINAL	BM ₃ D	KSVD	FOE	EPLL
MSE	417.799567286173	1431.7840447108	1004.97239723206	346.351612726847	3051.63115291595
PSNR	28.05106489921	28.8696579961094	31.9195868844955	30.6585253741552	27.4687140254927
SSIM	0.503258650626292	0.547327169567773	0.772548209904691	0.651176824730899	0.74796936425649
TIME	-	4.08653422853552	155.512157765963	47.1046152151738	470.539952226584

Tabela 3 - Rezultati algoritmov BM3D, KSVD, FOE in EPLL za $\sigma(\sigma) = 15$

MERA	ORIGINAL	BM ₃ D	KSVD	FOE	EPLL
MSE	417.799567286173	1854.13061841329	1472.17591400146	346.332693163554	3052.23666725159
PSNR	28.05106489921	29.4732661427322	32.0976378760292	30.7289629265889	27.467060643531
SSIM	0.503258650626292	0.579589695549325	0.78631396103294	0.655923669335412	0.747963822598516
TIME	-	4.09463048435028	102.067556888341	72.2930437216367	464.022409431046

Tabela 4 - Rezultati algoritmov BM3D, KSVD, FOE in EPLL za $\sigma(\sigma) = 20$

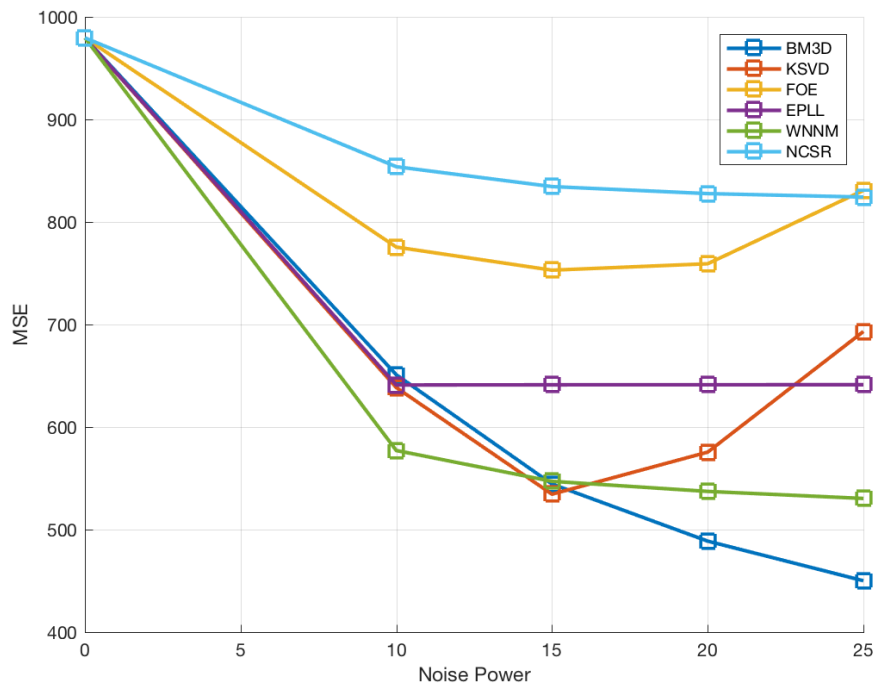
MERA	ORIGINAL	BM ₃ D	KSVD	FOE	EPLL
MSE	417.799567286173	2154.5869348526	1569.22057151794	356.031026649475	3052.5143863678
PSNR	28.05106489921	29.7334906041836	32.0754023103017	30.6175823531027	27.4662576695319
SSIM	0.503258650626292	0.60257843443217	0.791201320939154	0.650686513847135	0.747961322427379
TIME	-	4.04721102649289	110.485739583958	97.3735973713963	464.550138559487

Tabela 5 - Rezultati algoritmov BM3D, KSVD, FOE in EPLL za $\sigma(\sigma) = 25$

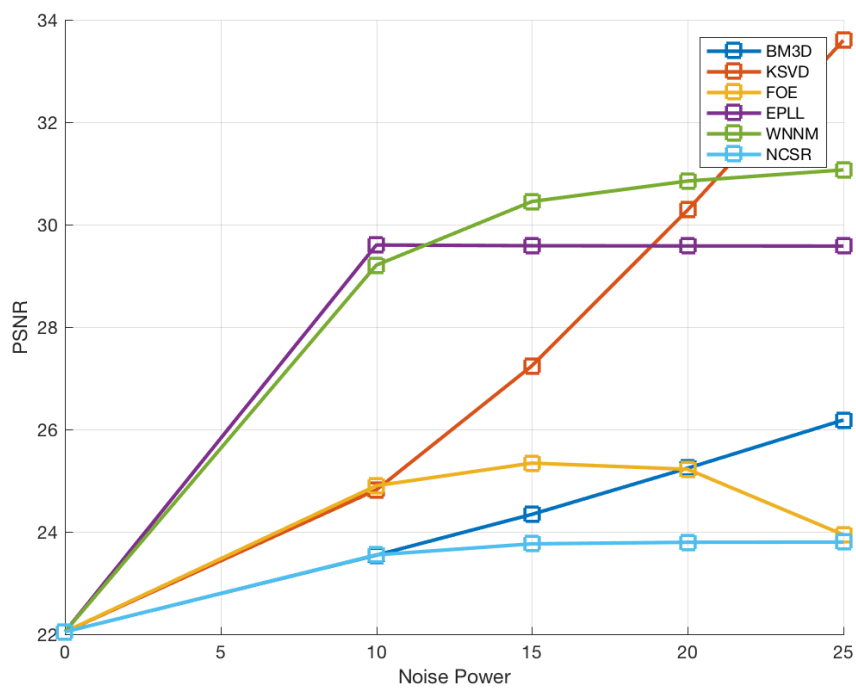
MERA	ORIGINAL	BM ₃ D	KSVD	FOE	EPLL
MSE	417.799567286173	3605.67248566945	2711.95502611796	407.955651028951	3052.88540643056
PSNR	28.05106489921	30.1380694687471	30.0466350917794	28.4576408996632	27.4652140127556
SSIM	0.503258650626292	0.646922619437927	0.749051394888013	0.526822518008497	0.74795754421696
TIME	-	6.75128353060625	138.462978977644	101.840249478132	464.770494408671

Tabela 6 - Rezultati algoritmov BM3D, KSVD, FOE in EPLL za $\sigma(\sigma) = 50$

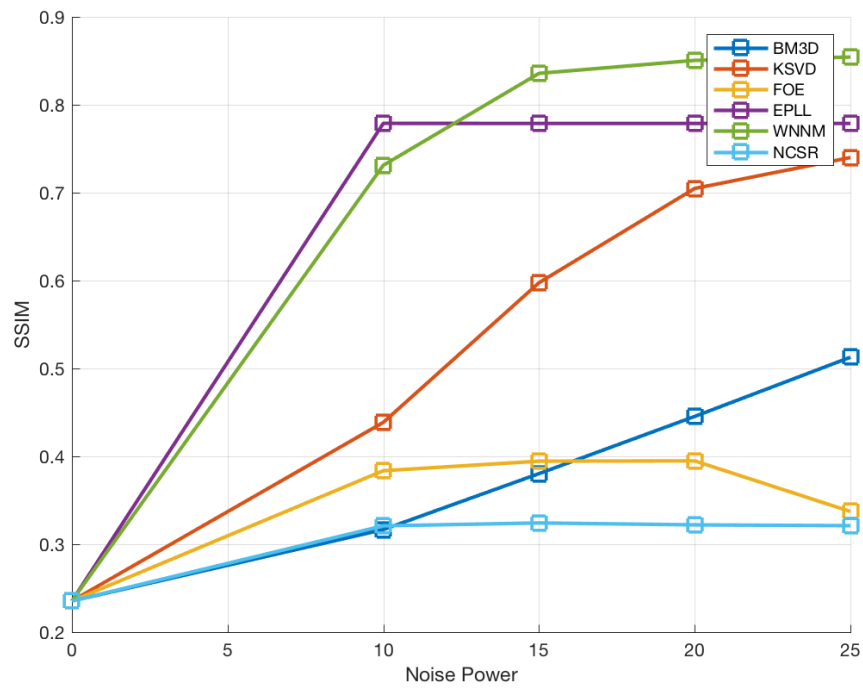
Zaradi slabega računalniške moči katere imamo na voljo, algoritmi NCSR in WNNM su bili opravljeni samo na eni sliki. Na tej eni sliki smo zagnali in ostale algoritme kateri su predstavljani v prejšnjim grafima. Algoritme smo testirali z vnaprej določenim vrednostima šuma σ : 10, 15, 20 in 25.



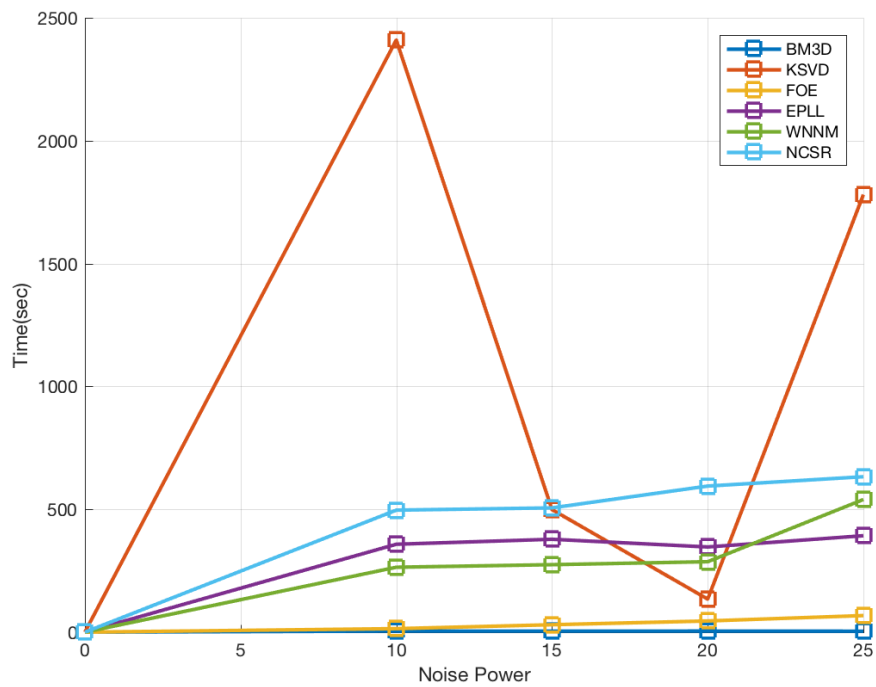
Slika 7 - Rezultati MSE metrike nad 1 sliko za algoritme BM3D, KSVD, FOE, EPLL, WNNM in NCSR



Slika 8 - Rezultat PSNR metrike nad 1 sliko za algoritme BM3D, KSVD, FOE, EPLL, WNNM in NCSR



Slika 9 - Rezultat SSIM metrike nad 1 sliko za algoritme BM3D, KSVD, FOE, EPLL, WNNM in NCSR



Slika 10 – Povprečni delovni čas nad 1 sliko za algoritme BM3D, KSVD, FOE, EPLL, WNNM in NCSR

POVZETEK IN RAZPRAVA

Vsi algoritmi kateri su uporabljeni od strane T. Ploetz, S. Roth (2017) su uporabljeni in v našem delu osim MLP in TNRD algoritmov. Tisti algoritmi su narejeni kot nevronske mreže in rabijo prej postopek učenja in tej del nisma uspeli narediti. Tudi čas izvajanja tista dva algoritma je zelo dolg. Algoritmi su bili dostopni čez različne spletne izvore in so bili dostopni skupaj z znanstvenim člankima kje je opisana implementacija. Težave su bile in z *WNNM in NCSR* ker čas izvajanja je bil dolg in smo jih zaradi tega naredili sem nad 1 sliko ker nimamo dostop do nekaterih močnih računalnika kateri bi lahko izvajali primerjalno analizo.

Tudi druga težava z izdelavom primerjalne analize je velikost nabora podatkov. Članek T. Ploetz, S. Roth (2017) uporablja nabor z 1000 sekljanih slik, in je naš nabor bil z 40 sekljalnih slik. Brez obzira na majni nabor vse skupni čas je bil zelo dolg. Zaradi tega rezultati katere smo mi dobili in katere su dobili avtorji glavneg članka su zelo drugači in kot rezultate kateri algoritmi su najboljši se razlikujejo. Članek T. Ploetz, S. Roth (2017) ugotavlja da je algoritem BM3D najboljši skupaj z ostalim člankima algoritmov ker se vsi ostali algoritmi tekmujejo z BM3D. Naši rezultati su ugotovili da su najboljši KSVDOM in WNNM, pač kak smo prej povedali naši rezultati nisu vredni ker nabor podatkov ni popolnoma dober.

Zaradi vsih teh težava pri implementaciji algoritmov in primerjalne analize naše je mnenje da je članek pre zahteven. Če se hoče vse narediti in prebrati rabili bi zelo veliko časa in bi bilo vredno detaljno prebrati, in ostale članke v katerima su opisani ostali algoritmi uporabljeni v primerjalni analizi. Tudi dostop do celotnog nabora podatkov članka *Benchmarking Denoising Algorithms with Real Photographs* ni mogoči in su dostopne sem slike z šumom. Brezšumnih slika ni, in potem primerjanje rezultatov ni mogoče. Če bi in bil celotni nabor dostopen, njegova priprava z našim računalnikima povprečne moči bi bila zelo zahtevna.

LITERATURA

- [1] T. Ploetz, S. Roth. *Benchmarking Denoising Algorithms with Real Photographs*. 2017
- [2] J. Anaya, A. Barbu. *RENOIR - A Dataset for Real Low-Light Image Noise Reduction*. 2014
- [3] Adrian Barbu's Research. *RENOIR - A Dataset of Real Low-Light Images*.
<http://adrianbarburesearch.blogspot.pt/p/renoir-dataset.html>. 2018
- [4] T. Ploetz, S. Roth.. *The Darmstadt Noise Dataset*. <https://noise.visinf.tu-darmstadt.de/>. 2018
- [5] The MathWorks, Inc. *Structural Similarity Index (SSIM) for measuring image quality*.
<https://www.mathworks.com/help/images/ref/ssim.html>. 2018
- [6] Wikipedia. *Structural similarity*. https://en.wikipedia.org/wiki/Structural_similarity. 2018
- [7] The MathWorks, Inc.. *Peak Signal-to-Noise Ratio (PSNR)*.
<https://www.mathworks.com/help/images/ref/psnr.html>. 2018
- [8] The MathWorks, Inc.. *Compute peak signal-to-noise ratio (PSNR) between images*.
<https://www.mathworks.com/help/vision/ref/psnr.html>. 2018
- [9] Wikipedia. *Noise (signal processing)*.
[https://en.wikipedia.org/wiki/Noise_\(signal_processing\)](https://en.wikipedia.org/wiki/Noise_(signal_processing)). 2018
- [10] Wikipedia. *Gaussian noise*. https://en.wikipedia.org/wiki/Gaussian_noise. 2018
- [11] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. *Image denoising by sparse 3D transform-domain collaborative filtering*. 2007
- [12] D. Zoran, Y. Weiss. *From Learning Models of Natural Image Patches to Whole Image Restoration*. 2011
- [13] S. Gu, L. Zhang, W. Zuo, X. Feng. *Weighted Nuclear Norm Minimization with Application to Image Denoising*. 2014
- [14] M. Aharon, M. Elad, A. Bruckstein. *K-SVD: Design of Dictionaries for Sparse Representation*. 2005
- [15] S. Roth, M. J. Black, *Fields of Experts*. 2009
- [16] W. Donga, L. Zhangb, G. Shia, X. Li. *Nonlocally Centralized Sparse Representation for Image Restoration*. 2012
- [17] H. C. Burger, C. J. Schuler, S. Harmeling. *Image denoising: Can plain Neural Networks compete with BM3D?*. 2012
- [18] Y. Chen, T. Pock. *Trainable Nonlinear Reaction Diffusion: A Flexible Framework for Fast and Effective Image Restoration*. 2016