

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 5908

**DEMOZAIK ALGORITAM TEMELJEN NA
DUBOKOM UČENJU**

Dajana Šeketa

Zagreb, lipanj 2019.

Zagreb, 6. ožujka 2019.

ZAVRŠNI ZADATAK br. 5908

Pristupnik: **Dajana Šeketa (0036502413)**
Studij: Računarstvo
Modul: Računalno inženjerstvo

Zadatak: **Demozaik algoritam temeljen na dubokom učenju**

Opis zadatka:

U sklopu završnog rada potrebno je istražiti algoritme temeljene na dubokom učenju za interpolaciju digitalnih slika u boji koje su dobivene senzorom s podotipkavanjem boja, kao npr. kod digitalnih fotoaparata s Bayerovom matricom boja. Duboka mreža koja se sastoji od konvolucijskih slojeva i potpuno povezanih slojeva treba biti trenirana na referentnim slikama za koje postoji informacija o svim primarnim bojama svakog pixela u punoj rezoluciji, te je pomoću regresijske mreže potrebno predvidjeti vrijednosti pixela za sve informacije koje nedostaju nakon prostornog podotipkavanja boja u skladu s korištenom matricom. Rezultate interpolacije ovih algoritama za različite arhitekture dubokih mreža potrebno je usporediti s u literaturi poznatim i često korištenim interpolacijskim postupcima, a kao mjeru distorzije koristiti PSNR i percepcijski utemeljenu subjektivnu mjeru degradacije.

Zadatak uručen pristupniku: 15. ožujka 2019.

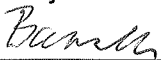
Rok za predaju rada: 14. lipnja 2019.

Mentor:



Prof. dr. sc. Davor Petrinović

Djelovođa:



Prof. dr. sc. Danko Basch

Predsjednik odbora za
završni rad modula:



Prof. dr. sc. Mario Kovač

Zahvala

Ovim putem htjela bih zahvaliti mentoru prof.dr.sc. Davoru Petrinoviću i asistentu Nikoli Vrebčeviću na savjetima i pomoći pri izradi završnog rada.

Također, veliko hvala mojim roditeljima bez kojih ništa od ovoga ne bi bilo moguće. Hvala sestri što me, u kriznim trenucima studiranja i želji za odustajanjem, nije uspjela prebaciti na građevinu. Hvala prijateljima s fakulteta koji su olakšali studentske dane i omogućili da ih pamtim u najboljem svjetlu, posebice Svenu što je imao svoje mjesto na hodniku ispred A202. I, naravno, hvala mojim zifama na konferencijskim pozivima, tješnju i ohrabrenjima.

Sadržaj

1. Uvod.....	4
2. Izvedbe CCD senzora.....	5
2.1. CCD senzor sa sekvencijskim dobivanjem boje.....	5
2.2. 3CCD senzor.....	6
2.3. Integrirani filtri za dobivanje boje.....	7
3. Bayer filter.....	8
4. Metode demozaiciranja.....	9
4.1. Metode temeljene na interpolaciji.....	10
4.1.1. Interpolacija najbližih susjeda.....	10
4.1.2. Gradijentom korigirana bilinearna interpolacija.....	10
4.2. Metode temeljene na rječniku.....	11
4.2.1. Demozaiciranje i uklanjanje šuma koristeći ResDNet.....	11
4.2.2. DMCNN.....	13
5. Artefakti demozaiciranja.....	15
5.1. Artefakt lažne boje.....	15
5.2. Artefakt resa.....	16
6. Usporedba kvalitete algoritama.....	17
6.1. PSNR test.....	17
6.2. PIQE test.....	18
8. Radna okolina.....	23
9. Zaključak.....	25
10. Literatura.....	26
11. Sažetak.....	28
12. Summary.....	29

1. Uvod

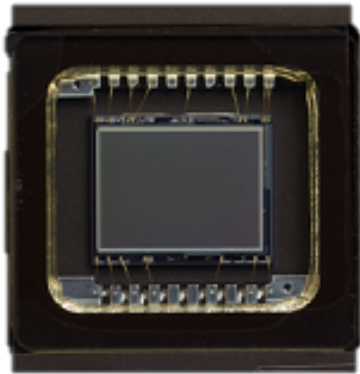
Digitalne kamere u današnje doba više nisu egzotična oprema, postale su tehnološki uređaj koji koristimo svakodnevno. Jedna od glavnih značajki koju gledamo pri kupnji kamere jest kvaliteta snimljenih fotografija.

Direktni utjecaj na kvalitetu fotografije u procesu obrade ima demosaiciranje. Uobičajene metode demosaiciranja, koje se koriste za brzu obradu sirovih fotografija, daju relativno loše rezultate. Velika većina modernih digitalnih kamera ima mogućnost spremanja fotografija u sirovom formatu, dopuštajući korisniku provođenje demosaiciranja korištenjem naprednih softvera i algoritama koji u konačnici daju puno kvalitetnije fotografije. Iz tog razloga počele su se razvijati metode demosaiciranja temeljene na dubokom učenju.

U ovom radu opisat ću proces nastanka sirove fotografije u digitalnoj kameri te usporediti kvalitetu interpoliranih fotografija nastalih primjenom algoritama zasnovanih na dubokom učenju i interpoliranih fotografija nastalih primjenom klasičnih, često korištenih, interpolacijskih postupaka.

2. Izvedbe CCD senzora

CCD (engl. Charge-Coupled Device) je uređaj s prijenosom naboja koji ima široku primjenu, a najpoznatija uporaba jest u sensorima svjetlosti digitalnih



Slika 1: CCD senzor [2]

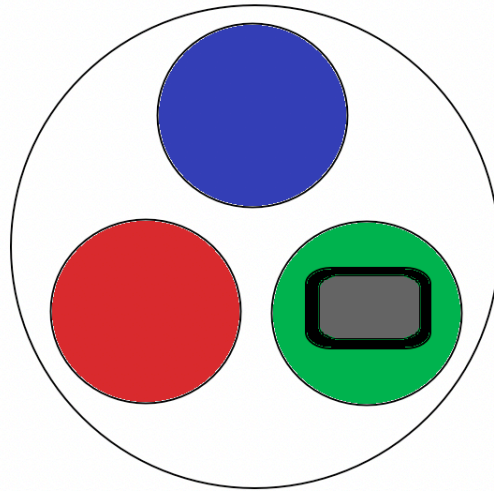
fotoaparata i kamera. Senzor se sastoji od piksela, a svaki piksel izveden je kao MOS kondenzator. Fotoelektričnim efektom svjetlost u svakom pikselu pretvara se u električni naboj i prenosi do sklopa koji ga pretvara u napon. Za svaki niz piksela naponski impuls sprema se u memoriju, a kada se obrade svi pikseli, formira se digitalna fotografija. [1]

CCD senzor nema sposobnost razlikovanja dijelova crvenog, plavog i zelenog svjetla. Zbog toga su se razvila tri načina za izdvajanje boje iz snimljene fotografije:

- CCD senzor sa sekvencijskim dobivanjem boje
- 3CCD senzor
- Integrirani filteri na samom senzoru

2.1. CCD senzor sa sekvencijskim dobivanjem boje

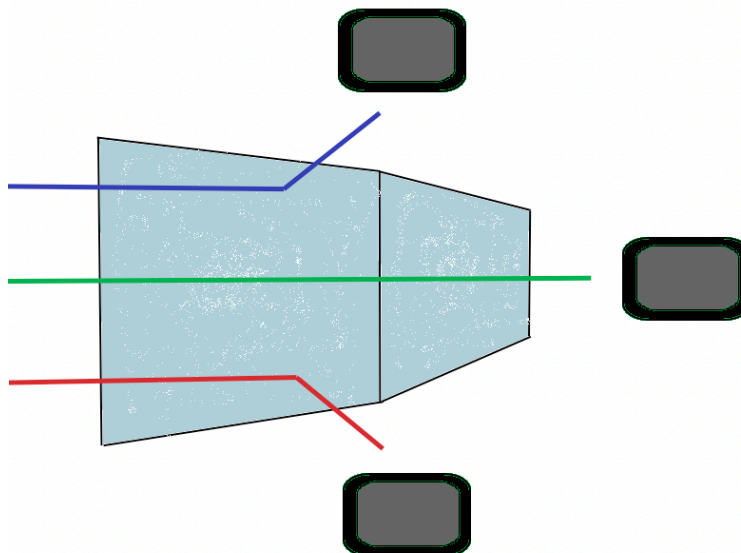
Slika se dobiva pomoću filtera za crvenu, plavu i zelenu boju smještenih na rotacijski kotač. Registriraju se jedna po jedna boja, a konačna fotografija dobiva se rekonstrukcijom intenziteta izvan senzora. Prednost ove metode jest dobivanje maksimalne rezolucije fotografije, a nedostatak što objekt koji se fotografira mora mirovati dulje vrijeme kako bi se izvršilo registriranje, te je potrebno više vremena za samo registriranje.



Slika 2: CCD senzor sa sekvencijskim dobivanjem boje

2.2. 3CCD senzor

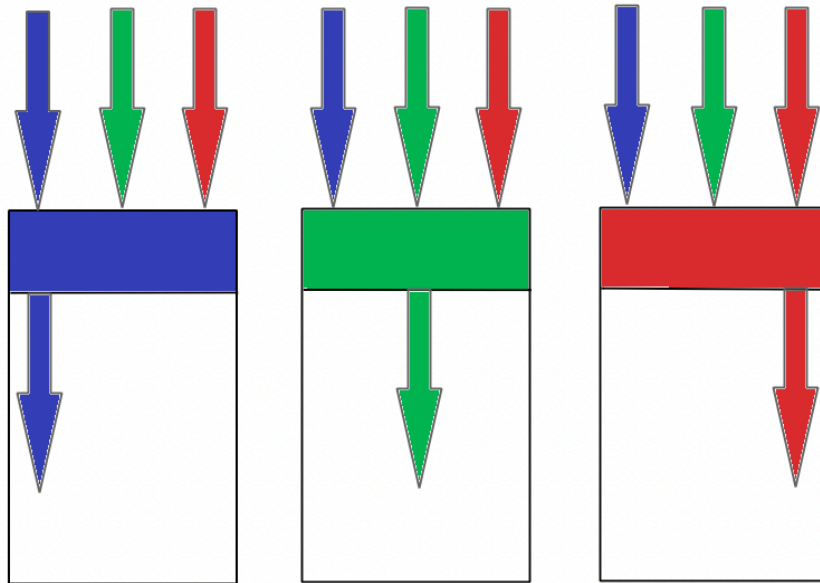
Sklop se sastoji od tri odvojena senzora, po jednog za svaku boju (crvena, plava, zelena). Svjetlo se na osnovne boje razdvaja pomoću prizme. Prednost ove metode jest istovremena registracija svih triju boja fotografiranog objekta, a nedostatak velika kompleksnost takvog uređaja.



Slika 3: Razdvajanje svjetla na osnovne boje

2.3. Integrirani filtri za dobivanje boje

Ideja ove metode jest filtriranje upadne zrake svjetla na samom senzoru. Svaki piksel može biti obojan jednom od tri osnovne boje te on, u ovisnosti o tome za koju boju je „zadužen“, pohranjuje informacije samo o toj boji.

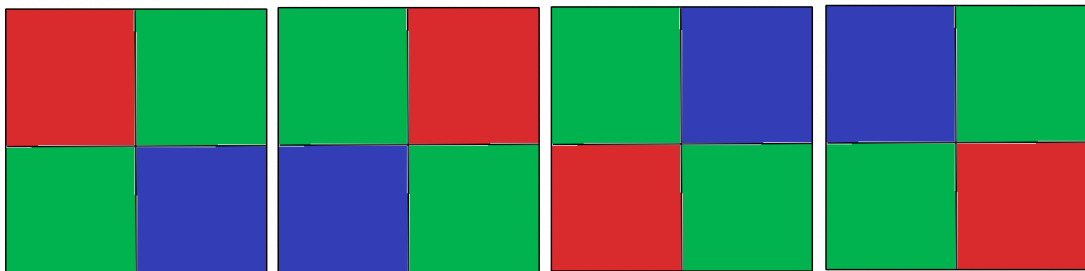


Slika 4: Filtriranje boja na pikselima

Takvim načinom pohranjivanja boja dobiva se mozaik piksela. Kako bismo iz mozaika dobili stvarni prikaz, kamera u svakom pikselu mora pretpostaviti intenzitet preostale dvije boje. Za približno određivanje preostale dvije boje u svakom pikselu koriste se razni algoritmi demosaiciranja koji predstavljaju glavnu temu ovoga rada.

3. Bayer filtar

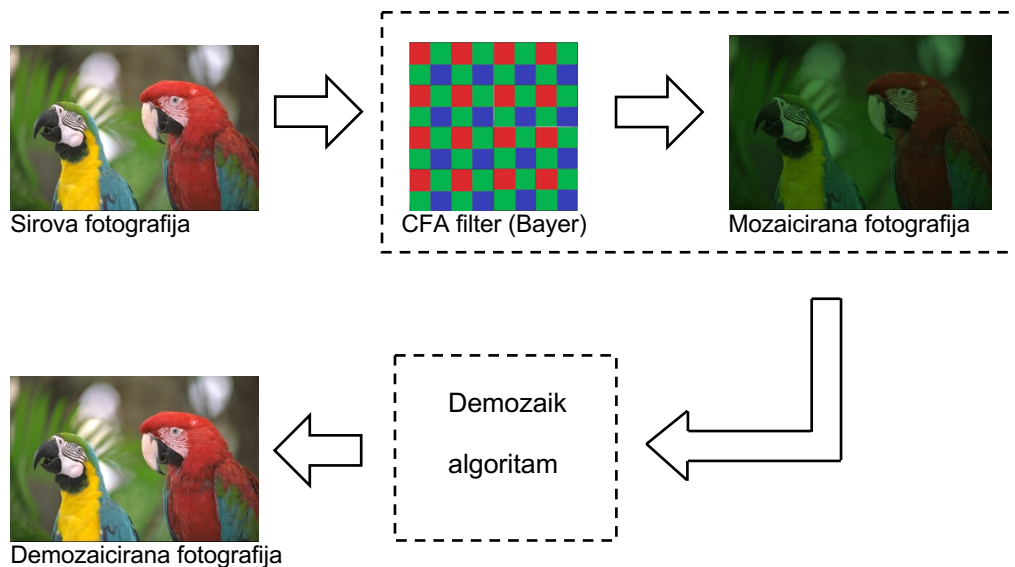
Najpoznatija izvedba filtra za generiranje informacija o bojama u izvedbi kamere s integriranim filtrima za dobivanje boje jest Bayerov filtar. Matrica se sastoji od četiri polja: jednog crvenog, jednog plavog te dva zelena. Razlog zbog kojeg je količina zelenih polja dvostruko veća od plavih i crvenih je taj što je ljudsko oko puno osjetljivije na zeleno svjetlo.



Slika 5: Moguće izvedbe Bayerovog filtra: RGGB, GRBG, GBRG, BGGR

4. Metode demozaiciranja

Algoritam demozaiciranja, CFA interpolacija ili Bayerova interpolacija je proces koji rekonstruira fotografiju u boji iz nepotpunih uzoraka dobivenih na sensorima fotoaparata prekrivenih mrežnim filterom boja (CFA).



Slika 6: Blok dijagram postupka demozaiciranja

Metode demozaiciranja mogu se grubo podijeliti u dvije osnovne skupine:

- Metode temeljene na interpolaciji
- Metode temeljene na rječniku

Metode temeljene na interpolaciji obično koriste opažanja lokalnih svojstava i iskorištavanja korelacije među njima. Međutim, ručno generirane značajke dobivene promatranjem često ne uspijevaju rekonstruirati složenije strukture. Iako se iteracijama i prilagodljivim shemama mogu poboljšati rezultati demozaiciranja metoda temeljenih na interpolaciji, one imaju ograničenja i za njihovu izvedbu potrebno je puno računanja.

Pristupi temeljeni na rječniku tretiraju demozaiciranje kao problem rekonstrukcije iz naučenih rječnika. Budući da je rječnik naučen, on pruža veću točnost boja rekonstruiranih slika. No, iako daju bolje rezultate, rekonstrukcija

metodama temeljenim na rječniku često traje dulje, čineći takve metode manje praktičnima.

U ovom radu opisane su i uspoređene četiri metode demozaiciranja: *interpolacija najbližih susjeda*, *gradijentom korigirana bilinearna interpolacija*, *demozaiciranje i uklanjanje šuma koristeći ResDNet* te *DMCNN*.

4.1. Metode temeljene na interpolaciji

4.1.1. Interpolacija najbližih susjeda

Najjednostavniji algoritam interpolacije koji radi tako da se vrijednosti plave i crvene boje u preostala tri polja Bayerove 2x2 matrice samo kopiraju. Vrijednost zelene boje u plavom i crvenom polju može se preuzimati iz susjednog zelenog polja vertikalno ili horizontalno. Ovaj algoritam ne koristi računanje za interpoliranje boje pa se koristi kod obrade kod koje je važnija brzina obrade od kvalitete konačne demozaicirane slike (npr. kod obrade videozapisa).

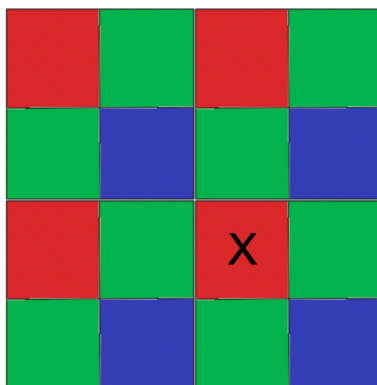
4.1.2. Gradijentom korigirana bilinearna interpolacija

Predstavlja poboljšanje algoritma bilinearne interpolacije. Kako bi interpolirali vrijednost zelene boje u crvenom polju (i,j) (X lokacija na Slici 7), popravljamo bilinearne interpoliranu vrijednost zelenog polja na lokaciji (i,j) pribrajanjem gradijenta vrijednosti crvene na toj lokaciji (i,j).

$$g_{\text{popravljeno}}(i, j) = g_{\text{bilinearno}}(i, j) + \alpha \Delta R \quad (1)$$

$$\Delta R = r(i, j) - r_{\text{prosječno}} \quad (2)$$

gdje $r_{\text{prosječno}}$ predstavlja prosječnu vrijednost u 4 najbliža crvena polja



Slika 7: Bayer 4x4 RGGB matrica

4.2. Metode temeljene na rječniku

4.2.1. Demozaiciranje i uklanjanje šuma koristeći ResDNet [4]

Ova metoda obrade „sirove“ fotografije dubokom neuronskom mrežom istovremeno provodi dva koraka – demozaiciranje i uklanjanje šuma.

Jedan od glavnih nedostataka prijašnjih metoda koje se bave problemom demozaiciranja na temelju dubokog učenja je taj što pretpostavljaju specifičan Bayerov filter korišten na sensorima kamere. Takva pretpostavka ograničava primjenjivost metoda, budući da postoje mnoge vrste kamere na tržištu koje koriste različite uzorke filtera boja. Problem kod tipičnih metoda interpolacije (opisanih u odjeljku iznad), koje također rade s bilo kojim uzorcima filtera boja, jest velik broj artefakata na rubovima objekata.

Cjelokupna arhitektura ovog pristupa demozaiciranju temelji se na *MM frameworku* i dubokoj rezidualnoj mreži za uklanjanje šuma. *MM* je iterativna optimizacijska metoda koja iskorištava konveksnost funkcije kako bi pronašla njihove maksimume ili minimume i provodi se u K diskretnih koraka. Rješenje početne funkcije računa se iterativno minimiziranjem zamjenske funkcije. Zamjenske funkcije daju gornju granicu početne funkcije i jednostavnije su za rješavanje. U svakom od K koraka iteracije demozaiciranja koristi se mreža *ResDNet* za uklanjanje šuma s istim skupom parametara. Korištenjem istih

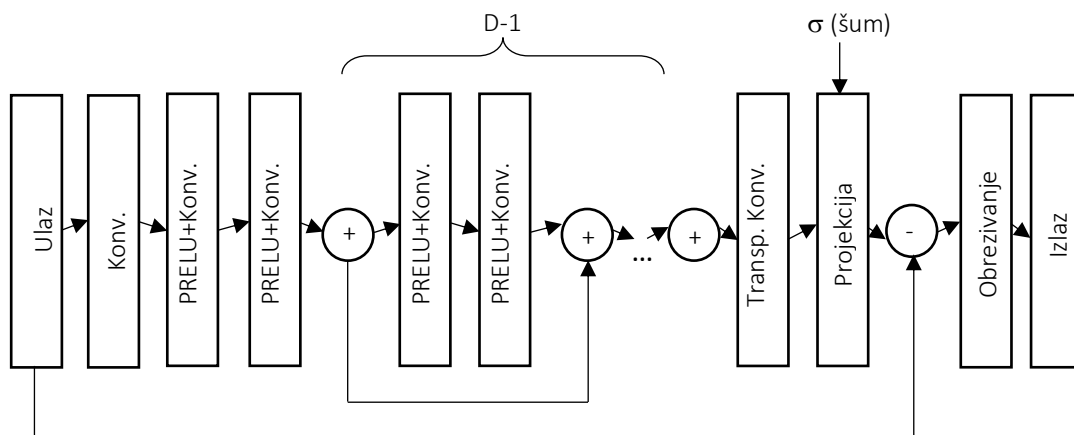
parametara za uklanjanje šuma u svim iteracijama *MM* algoritma pri demozaiciranju cjelokupni pristup zadržava mali broj ulaznih parametara i zahtijeva manji broj ulaznih fotografija za treniranje mreže.

Rezidualna mreža dizajnirana je tako da se izlaz mreže oduzima od njegovog ulaza, što znači da je zadatak mreže procjena količine šuma koji degradira ulaz. Međutim, treniranje duboke mreže koja zahtijeva veliki broj parametara za učenje svake razine šuma može biti prilično nepraktično, posebice u slučajevima kada se takve mreže želi koristiti na uređajima s ograničenom memorijom. Ovaj problem riješen je tako da mreža prihvaća dva argumenta na ulazu – zašumljenu ulaznu sliku i varijancu šuma koja se zatim koristi za pomoć mreži pri procjeni šuma koji degradira ulaz. Na taj način moguće je naučiti mrežu skup parametara i koristiti istu mrežu za ulaze koji su zašumljeni širokim rasponom razina šuma.

Mreža za uklanjanje šuma dubine D sastoji se od 5 osnovnih blokova. Prvi blok je konvolucijski sloj sa 64 kanala čije veličine kernela su 5×5 . Drugi je nelinearni blok koji se sastoji od PReLU funkcije i konvolucijskog sloja sa 64 kanala i kernelima veličine 3×3 . PReLU funkcija definirana je kao

$$\text{PReLU}(x) = \max(0, x) + k \cdot \min(0, x) \quad (3)$$

gdje je k vektor veličine jednake broju ulaznih kanala. Mreža sadrži $D \cdot 2$ nelinearnih blokova gdje je svaki drugi blok povezan prečicom. Izlaz nelinearnog bloka obrađuje se transponiranim konvolucijskim blokom koji smanjuje broj kanala sa 64 na 3, veličine kernela 5×5 . Zatim slijedi projekcijski blok koji kao dodatni ulaz prihvaća varijancu šuma i čija uloga je normalizirati procjenu šuma tako da ima ispravnu varijancu prije nego se ona oduzme od ulaza u mrežu. Konačno, rezultat je podrezan tako da su intenziteti izlaza u rasponu $[0, 255]$. Ovaj posljednji korak primjenjuje prethodno znanje o očekivanom rasponu važećih intenziteta piksela.



Slika 8: Arhitektura ResDNet duboke mreže

4.2.2. DMCNN [6]

Budući da su relevantne informacije za demosaiciranje često prisutne samo lokalno, kod metode DMCNN sitni dijelovi slika gusto su izdvojeni i prezentirani kao ulaz u mrežu. Korišteni su isječci $33 \times 33 \times 3$. Svaki piksel isječka sastoji se od 3 kanala u boji od kojih jedan sadrži informaciju o određenoj boji u tom pikselu (plavoj, zelenoj ili crvenoj), a ostala dva kanala su prazna. Druga mogućnost bila bi predati mreži na ulazu isječke dimenzija $33 \times 33 \times 1$, no tada bi bilo prezahtjevno shvatiti koju boju kanala svaki piksel predstavlja. Za takvu izvedbu bile bi potrebne 4 odvojene mreže za 4 različite lokacije u CFA matrici. Učinkovitije je ostaviti kanale koji nedostaju prazne i trenirati jedinstvenu mrežu. Također, na taj način dizajnirana mreža je i fleksibilnija za različite uzorke CFA matrica.

Arhitektura ove mreže sastoji se od 3 sloja od kojih svaki ima specifičan zadatak. Prvi sloj, sloj ekstrakcije svojstava, odgovoran je za izvlačenje korisnih lokalnih svojstava. Korišteno je 128 filtara veličine 9×9 koji su inicijalizirani kao Gaussovi kerneli. Izlaz ovog sloja može se promatrati kao mapa 128-d vektora svojstava. Drugi sloj naziva se nelinearni sloj mapiranja. Njegova uloga je mapiranje izvučenih vektora svojstava visoke rezolucije u vektore niže rezolucije. Pri tome je korišteno 64 kernela dimenzija 1×1 . Na taj način nelinearno mapiranje je izvedeno na samom pikselu, ne uzimajući u

obzir odnose među susjedima. Treći i posljednji sloj je rekonstrukcijski sloj, dizajniran za rekonstrukciju obojanog isječka iz danog seta značajki. Kerneli dimenzija 5x5, inicijalizirani kao Gaussovi kerneli, iskorištavaju lokalne informacije za rekonstrukciju finalne boje.

Kao aktivacijska funkcija korištena je ReLU (eng. *Rectified Linear Units*) jer može izbjeći problem nestajanja gradijenta do kojeg dolazi zbog međusobnog množenja malih brojeva u uzastopnim iteracijama. ReLU funkcija može se definirati formulom:

$$f(x) = \max(0, x) \quad (4)$$

Matematički, mreža se može predstaviti formulama:

$$F1(Y) = \max(0, W1 * Y + B1) \quad (5)$$

$$F2(Y) = \max(0, W2 * F1(Y) + B2) \quad (6)$$

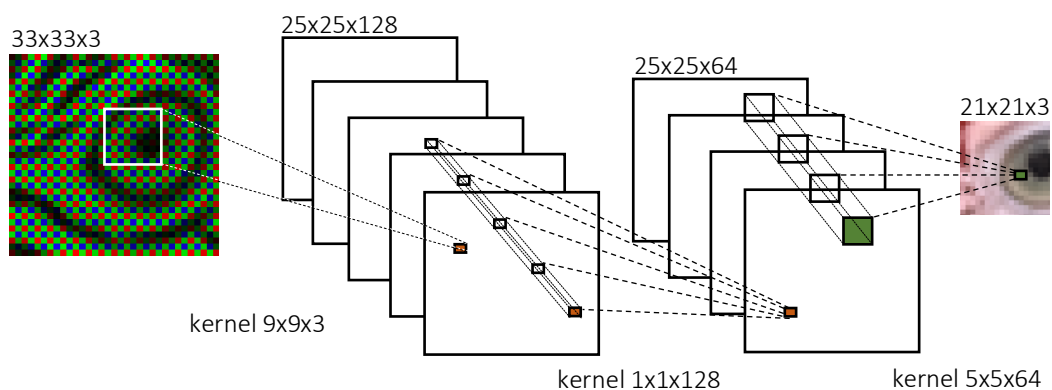
$$F(Y) = W3 * F2(Y) + B3 \quad (7)$$

gdje Y predstavlja ulazni isječak, F_i je izlazna mapa značajki, W_i je matrica težina, a B_i (eng. *Bias vector*) vektor pomaka i proširenje područja djelovanja neurona i-tog sloja. W i B su parametri neuronske mreže koji se mijenjaju u postupku učenja. Znak * u ovom slučaju predstavlja operator konvolucije.

Razlika očekivanog i dobivenog izlaza tereti se funkcijom gubitka

$$L(\theta) = \frac{1}{n} \sum_{i=1}^n \|F(Y_i; \theta) - X_i\|^2 \quad (8)$$

gdje je Y_i i-ti mozaicirani isječak, X_i odgovarajući obojani isječak (*groundtruth*), a n broj uzoraka za treniranje. Cilj neuronske mreže je minimizirati funkciju gubitka promjenama parametara u smjeru negativnog gradijenta funkcije gubitka. [11]



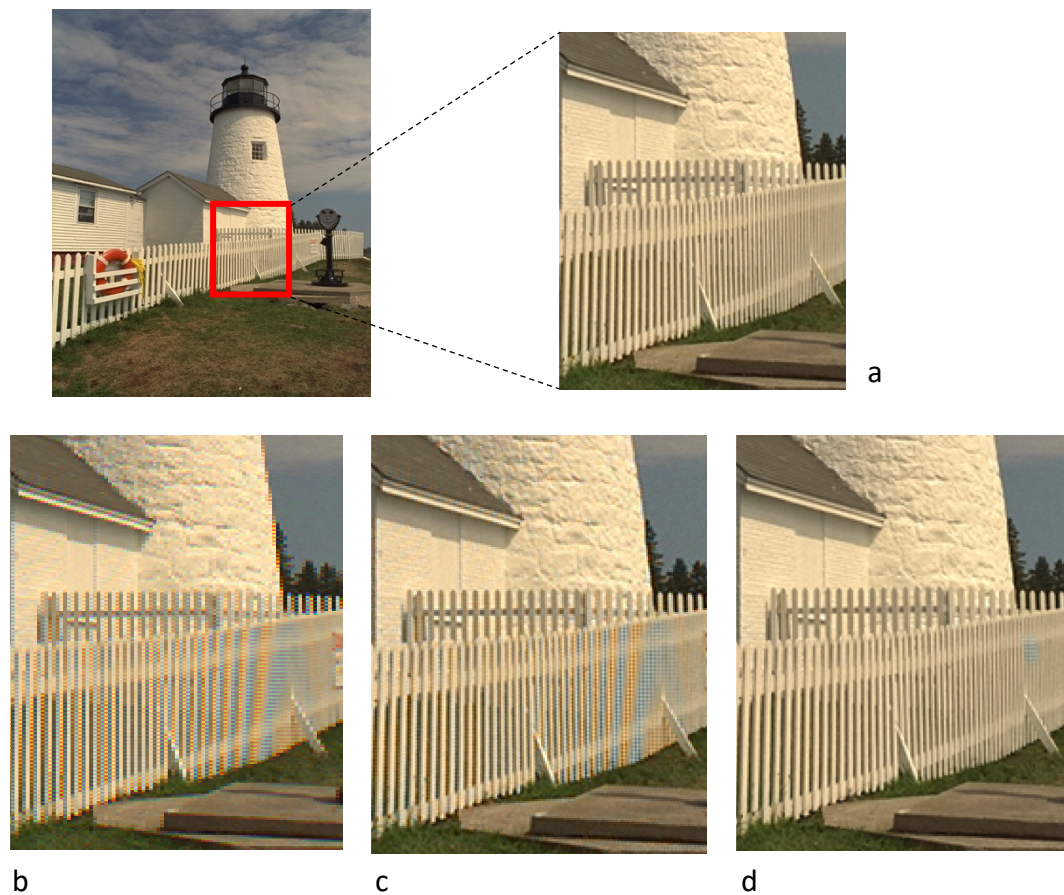
Slika 9: Arhitektura DMCNN duboke mreže

5. Artefakti demozaiciranja

Fotografiranje kamerom s Bayerovim filtrom daje samo 33% originalne informacije. Iz tog razloga prilikom demozaiciranja javljaju se razni artefakti (dijelovi slike koji nisu dobro „prevedeni“). Dva najpoznatija među njima su artefakt lažne boje i artefakt resa.

5.1. Artefakt lažne boje

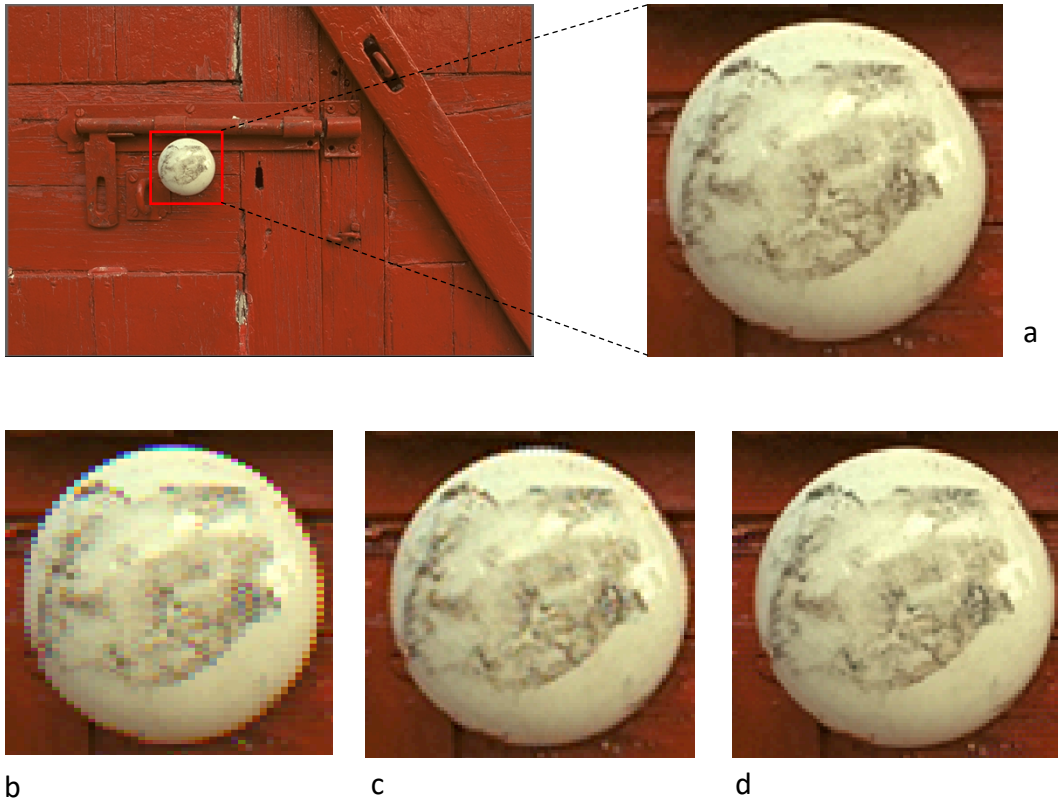
Artefakt lažne boje tipično se uočava na fotografijama na kojima dolazi do naglih promjena u bojama kao neprirodna promjena boja na rubovima elemenata. Do ove pojave dolazi zbog krive interpolacije fotografije poprijeko umjesto uzduž rubova.



Slika 10: Vizualna usporedba artefakata lažne boje
a) originalna fotografija b) interpolacija najbližih susjeda
c) gradijentom korigirana bilinearna interpolacija d) ResDNet metoda

5.2. Artefakt resa

Artefakt resa se, kao i artefakt lažne boje, također uočava na rubovima elemenata fotografije. Očituje se kao zamućenje i nejasno definiran rub elementa.



Slika 11: Vizualna usporedba artefakata resa
a) originalna fotografija b) interpolacija najbližih susjeda
c) gradijentom korigirana bilinearna interpolacija d) ResDNet metoda

6. Usporedba kvalitete algoritama

6.1. PSNR test

PSNR (eng. *Peak Signal To Noise Ratio*) je objektivni postupak za mjerenje kvalitete slike koji na temelju usporedbe s originalnim sadržajem računa stupanj izobličenja promatranog sadržaja. Stupanj izobličenja rezultat je formule:

$$\text{PSNR}_{\text{dB}} = 10 \log_{10} \frac{(2^n - 1)^2}{\text{MSE}} \quad (9)$$

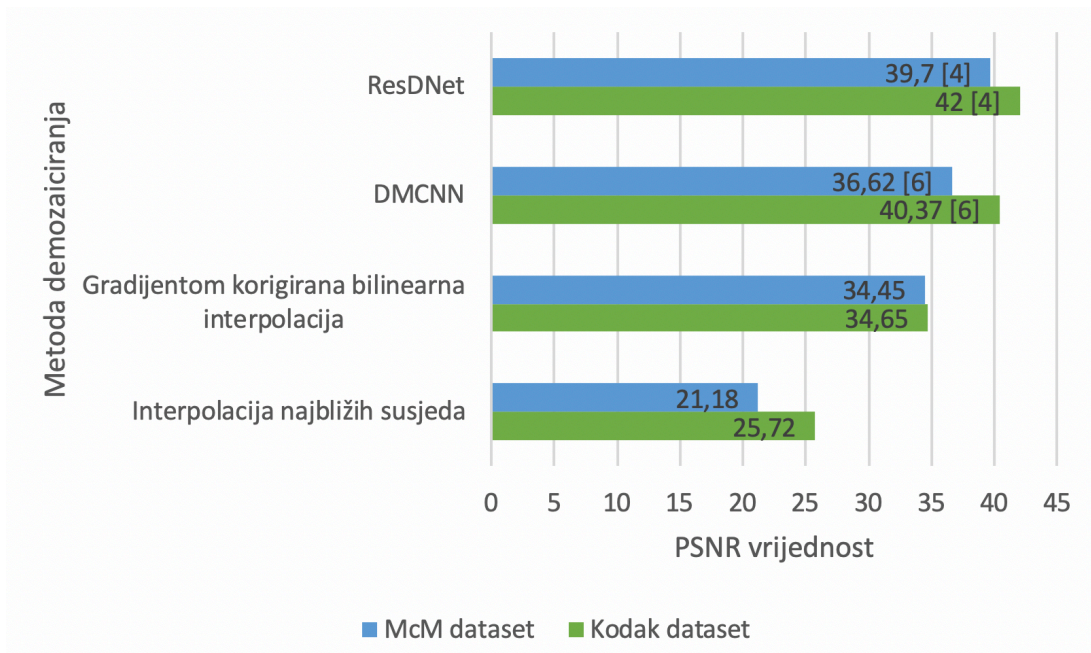
gdje:

- MSE (eng. *Mean Square Error*) predstavlja srednju kvadratnu pogrešku između elemenata originalne i izobličene fotografije
- n predstavlja broj bita za kodiranje elemenata izvorne fotografije. [5]

Vrijednosti PSNR metode su nenegativni decimalni brojevi, a interpolacija je uspješnija što je vrijednost veća.

Uspoređujući prosječne PSNR vrijednosti pojedinih algoritama, može se zaključiti da je od navedenih metoda demozaiciranja, prema PSNR testu, najuspješnija *metoda demozaiciranja i uklanjanja šuma koristeći ResDNet* s prosječnom PSNR vrijednosti 42,00 za *Kodak dataset* te 39,70 za *McM dataset*, a najmanje uspješna interpolacija najbližih susjeda s prosječnom PSNR vrijednosti 25,72 za *Kodak dataset* te 21,18 za *McM dataset*.

Algoritmi daju manje PSNR vrijednosti za McM dataset, što potvrđuje činjenicu da je jednostavnije izvesti interpolaciju na fotografijama s manje detalja i više uniformno obojenih dijelova.



Slika 12: Usporedba kvalitete algoritama na temelju PSNR testa

6.2. PIQE test

PIQE (eng. *Perception based Image Quality Evaluator*) je postupak koji izračunava ocjenu kvalitete slike bez referentne vrijednosti za sliku pomoću procjenitelja kvalitete slike temeljene na percepciji. [7]

Ocjena kvalitete računa se pomoću procjene iskrivljenja blokova koristeći sljedeće korake:

1. izračunavanje MSCN (eng. *Mean Subtracted Contrast Normalized*) koeficijenta za svaki piksel na slici koristeći algoritam predložen u radu autora N. Venkatanath i ostalih. [8]
2. podjela ulazne slike u blokove 16x16 koji se međusobno ne preklapaju
3. identificiranje visoko prostorno aktivnih blokova na temelju varijance MSCN koeficijenata
4. generiranje *activityMask*-e koristeći identificirane visoko prostorno aktivne blokove
5. procjena distorzije u svakom bloku nastale zbog artefakata i buke korištenjem MSCN koeficijenata

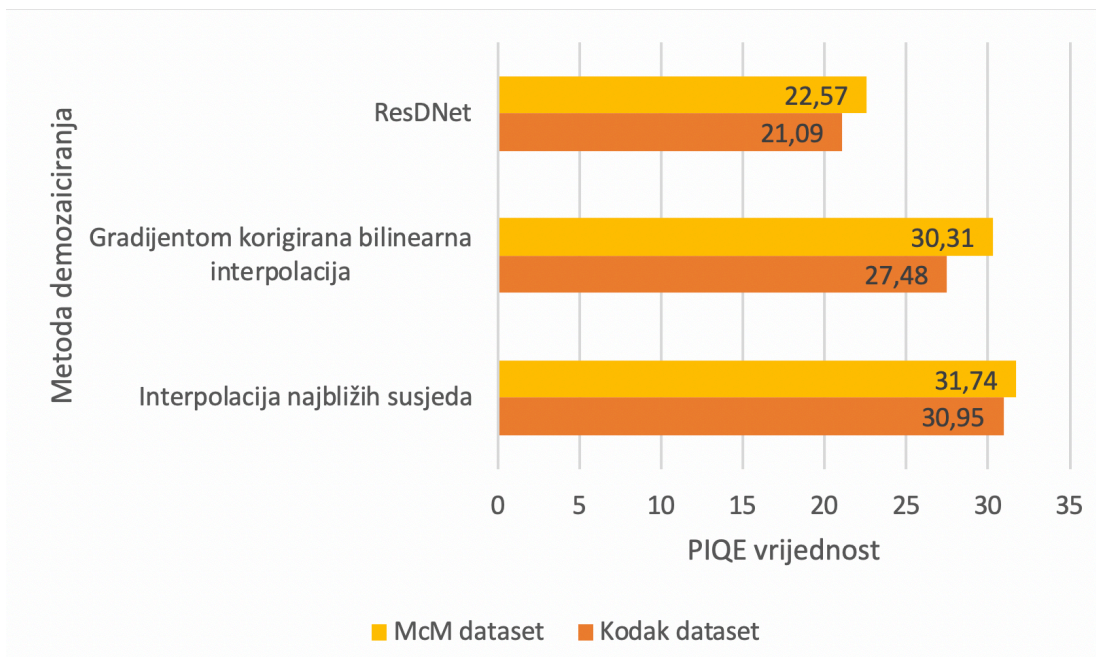
6. klasificiranje blokova kao iskrivljenih blokova s artefaktima, iskrivljenih blokova s Gausovim šumom ili neiskrivljenih blokova koristeći kriterije praga
7. generiranje *noticeableArtifactsMask*-e iz iskrivljenih blokova s artefaktima i *noiseMask*-e iz iskrivljenih blokova s Gausovim šumom
8. izračunavanje rezultata PIQE testa za ulaznu sliku kao srednje vrijednosti rezultata u iskrivljenim blokovima
9. svrstavanje rezultata u jednu od 5 kategorija iz Tablice 1 prema dobivenom rezultatu PIQE testa

Tablica 1: Ljestvica kvalitete fotografije na temelju njezine PIQE ocjene [8]

Ljestvica kvalitete	Raspon rezultata PIQE testa
odlično	[0, 20]
dobro	[21, 35]
zadovoljavajuće	[36, 50]
slabo	[51, 80]
loše	[80, 100]

Uspoređujući prosječne PIQE vrijednosti pojedinih algoritama, može se zaključiti da je od navedenih metoda demosaiciranja, prema PIQE testu, najuspješnija *metoda demosaiciranja i uklanjanja šuma koristeći ResDNet* s prosječnom PIQE vrijednosti 21,09 za *Kodak dataset* te 22,57 za *McM dataset*, a najmanje uspješna interpolacija najbližih susjeda s prosječnom PIQE vrijednosti 30,95 za *Kodak dataset* te 31,74 za *McM dataset*.

Unatoč varijacijama u rezultatima PIQE testa, sve tri metode demosaiciranja daju isti rezultat ljestvice kvalitete slike na temelju PIQE ocjene.



Slika 13: Usporedba kvalitete algoritama na temelju PIQE testa



Slika 14: Kodak dataset [9]

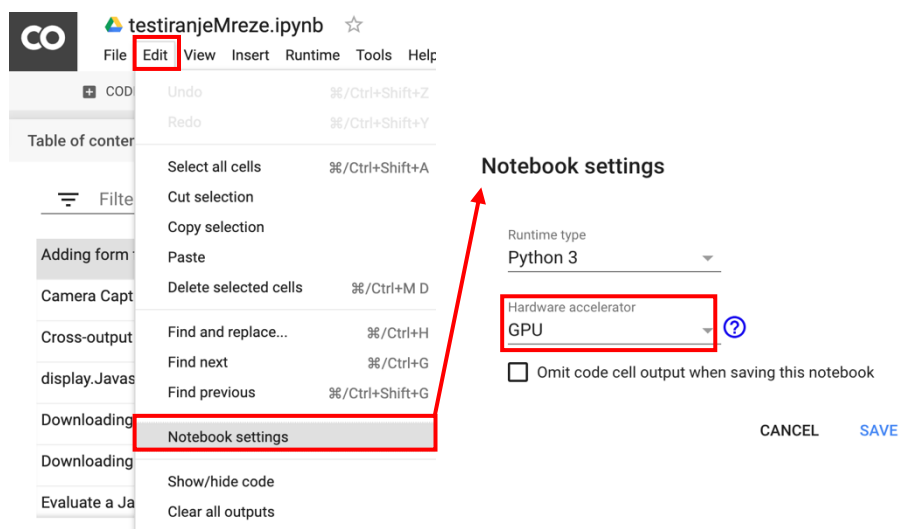


Slika 15: McM dataset [10]

8. Radna okolina

Radna okolina korištena za testiranje demozaik algoritama temeljenih na dubokom učenju naziva se *Google Colaboratory*. To je okruženje za pisanje i izvršavanje koda, spremanje i objavljivanje analiza te korištenje računalnih resursa (CPU, GPU) bez potrebe za instalacijom i preuzimanjem programa na vlastito računalo. Sve što je potrebno za pokretanje jest web preglednik. Colaboratory je temeljen na Jupyteru [12] – projektu otvorenog koda za interaktivno računarstvo na više desetaka programskih jezika.

Razlog odabira *Google Colaboratory*-ja za testiranje algoritama temeljenih na dubokom učenju je taj što omogućuje besplatno korištenje grafičkog procesora Tesla K80 te 12 GB RAM-a.



Slika 16: Omogućavanje GPU-a na *Google Colaboratory*-ju

Dokument u *Google Colaboratory*-ju sastoji se od ćelija (Slika 17) u kojima može biti običan tekst ili kod i izlaz njegovog izvršavanja. Sve ćelije mijenjaju isto globalno stanje, što znači da se varijable definirane izvršavanjem jedne ćelije mogu koristiti i u drugim ćelijama. Ćelije se pokreću klikom na gumb *play* u lijevom gornjem kutu ćelije, a u slučaju ulaska u beskonačnu petlju ili predugog izvršavanja ćelije, njen rad može se prekinuti pritiskom gumba *stop*. Po završetku izvođenja ćelije, ispod izvršene ćelije prikazat će se ili popis grešaka ili rezultat izvođenja ćelije.


```

import skimage as sk
from skimage import io, util
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import torch
import torch.nn as nn
import math, sys, os
from residual_model_resdnet import *
from MMNet_TBPTT import *
import glob
from subprocess import call, check_output
from tqdm import *
from subprocess import Popen, PIPE

```

```

-----
ModuleNotFoundError                               Traceback (most recent call last)
<ipython-input-1-c6fa8d78d73e> in <module>()
      7 import torch.nn as nn
      8 import math, sys, os
---->  9 from residual_model_resdnet import *
     10 from MMNet_TBPTT import *
     11 import glob

ModuleNotFoundError: No module named 'residual_model_resdnet'

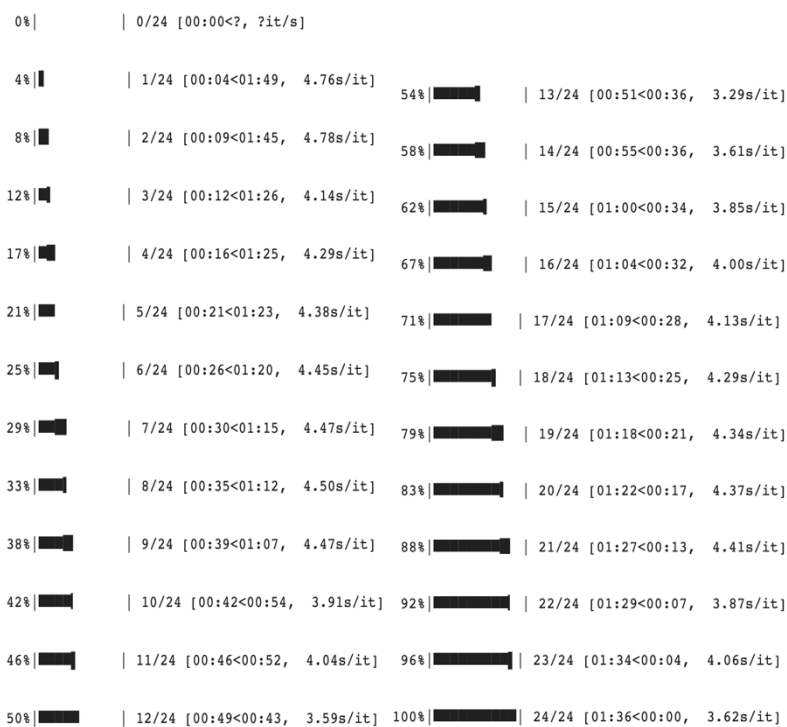
-----
NOTE: If your import is failing due to a missing package, you can
manually install dependencies using either !pip or !apt.

To view examples of installing some common dependencies, click the
"Open Examples" button below.

-----
OPEN EXAMPLES  SEARCH STACK OVERFLOW

```

Slika 17: Čelija i popis grešaka pri njenom izvođenju



Slika 18: Izlaz ćelije prilikom testiranja metode *demozaiciranja i uklanjanja šuma* koristeći *ResDNet* na *Kodak dataset-u*

Metode temeljene na interpolaciji te PSNR i PIQE testovi za usporedbu kvalitete algoritama pokretani su u *MATLAB-u* – interaktivnoj okolini za numeričko i matrično računanje, te za vizualizaciju i programiranje.

9. Zaključak

U ovom radu opisan je proces obrade „sirove“ fotografije koja nastaje u digitalnim fotoaparatom koji sadrže jedan senzor za očitavanje boje. Površina senzora prekrivena je nizom obojanih filtra poredanih u mozaik tako da svaki element senzora uzorkuje po jednu od tri moguće komponente boje (plavu, zelenu ili crvenu). Najraširenija izvedba filtra na senzoru, koja se i opisuje u ovom radu, je Bayerov filter.

Za procjenu boje u svakom pikselu na stvarnoj fotografiji potrebno je odrediti vrijednost dviju komponenti koje nedostaju na „sirovoj“ fotografiji prekrivenoj Bayerovim filtrom. Taj se proces obično naziva demozaiciranjem, a njegov rezultat demozaicirana fotografija.

Uspoređene su kvalitete izvedbe dvaju algoritama demozaiciranja temeljenih na dubokom učenju s dvama često korištenim interpolacijskim postupcima s obzirom na dvije vrste mjere kvalitete: PSNR i PIQE test.

Također, vizualnom usporedbom demozaiciranih fotografija, opisani su i prikazani artefakti lažnih boja i artefakti resa koji se javljaju prilikom demozaiciranja.

Prema rezultatima dobivenim mjerenjima može se zaključiti da algoritmi temeljeni na dubokom učenju daju bolje rezultate u oba testa. Uz to, algoritmi temeljeni na dubokom učenju omogućuju istovremeno demozaiciranje i uklanjanje šuma na fotografijama, no zahtijevaju puno veći početni angažman jer im je potrebno osigurati velik ulazni skup fotografija za treniranje duboke mreže te im je potrebno određeno vrijeme „učenja“.

10. Literatura

- [1] CCD, <http://www.vidipedija.com/index.php?title=CCD>
(svibanj 2019.)
- [2] Lucas Hofer, CMOS vs. CCD Sensors and Overview,
<https://www.dataray.com/blog-cmos-vs-ccd-overview.html>
(svibanj 2019.)
- [3] T.Ciceli: *Nastajanje boje u CCD senzoru* – Ekscentar 2007, br. 9, str. 30.-32., <https://hrcak.srce.hr/12469> (svibanj 2019.)
- [4] Kokkinos, F., Lefkimmiatis, S. (2018). Iterative Residual Network for Deep Joint Image Demosaicking and Denoising. arXiv:1807.06403,
<https://arxiv.org/pdf/1807.06403.pdf> (svibanj 2019.)
- [5] psnr,
<https://www.mathworks.com/help/images/ref/psnr.html>
(svibanj 2019.)
- [6] Nai-Sheng S., Yu-Sheng C., Yung-Yu C. (2018.) Learning deep Convolutional Networks for Demosaicing. arXiv:1802.03769,
<https://arxiv.org/pdf/1802.03769.pdf> (svibanj 2019.)
- [7] piqe,
<https://www.mathworks.com/help/images/ref/piqe.html>
(svibanj 2019.)
- [8] N. Venkatanath, D. Praneeth, Bh. M. Chandrasekhar, S. S. Channappayya, and S. S. Medasani. "Blind Image Quality Evaluation Using Perception Based Features", In *Proceedings of the 21stNational Conference on Communications (NCC)*. Piscataway, NJ: IEEE, 2015.

[9] Kodak Lossless True Color Image Suite,
<http://r0k.us/graphics/kodak/> (svibanj 2019.)

[10] L. Zhang, X. Wu, A. Buades, and X. Li, "Color Demosaicking by Local Directional Interpolation and Non-local Adaptive Thresholding," *Journal of Electronic Imaging* 20(2), 023016 (Apr-Jun 2011), DOI:10.1117/1.3600632, https://www4.comp.polyu.edu.hk/~cslzhang/CDM_Dataset.htm (svibanj 2019.)

[11] Viduka, I. Klasifikacija slika dubokim modelima i Fisherovim vektorima. Diplomski rad. Fakultet elektrotehnike i računarstva, 2017.

[12] <https://jupyter.org/>

11. Sažetak

Demozaik algoritam temeljen na dubokom učenju

Postoji više vrsta senzora za dobivanje fotografije u digitalnoj kameri, a u ovom radu detaljnije je opisan postupak obrade „sirove“ fotografije dobivene na senzoru prekrivenom integriranim Bayerovim filtrom. Taj postupak naziva se demozaiciranjem, a njegov rezultat demozaicirana fotografija. Postoje razne implementacije demozaik algoritama, od ugrađenih *MatLab* funkcija do onih temeljenih na dubokom učenju. U ovom radu provedena je usporedba kvalitete četiri takva algoritma PSNR i PIQE testovima te su vizualnom usporedbom interpoliranih fotografija prikazani najčešći artefakti koji se javljaju prilikom demozaiciranja: artefakt lažne boje i artefakt resa.

Ključne riječi: algoritmi demozaiciranja, Bayerov filter, duboko učenje, regresijska mreža

12. Summary

Deep Learning Based Demosaicking Algorithm

There are several types of built-in color sensors in digital cameras, and in this paper, a more detailed description of the „raw“ image processing procedure obtained on the sensor covered with the integrated Bayer filter is described. This process is called demosaicking, and its result is a demosaiced photograph. There are various implementations of demosaicking algorithms, from embedded *MatLab* functions to those based on deep learning. In this paper four such algorithms are compared with the PSNR and PIQE tests. Also, visual comparison of interpolated photographs shows the most common artifacts that occur during demosaicking: false-color and zippering artifact.

Key words: demosaicking algorithm, Bayer pattern, deep learning, regression neural network