

# Stupanj kompresije i utjecaj na kvalitetu Webp formata slike

---

Mijić, Josip

Master's thesis / Diplomski rad

2014

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Graphic Arts / Sveučilište u Zagrebu, Grafički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:216:013568>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-17**



Repository / Repozitorij:

[Faculty of Graphic Arts Repository](#)



SVEUČILIŠTE U ZAGREBU

GRAFIČKI FAKULTET

JOSIP MIJIĆ

STUPANJ KOMPRESIJE I UTJECAJ NA  
KVALITETU WEBP FORMATA SLIKE

DIPLOMSKI RAD

Zagreb, 2014 godina



Sveučilište u Zagrebu  
Grafički fakultet

JOSIP MIJIĆ

STUPANJ KOMPRESIJE I UTJECAJ NA  
KVALITETU WEBP FORMATA SLIKE

DIPLOMSKI RAD

Mentor:

Doc.dr.sc Ante Poljičak

Student:

Josip Mijić

Zagreb, 2014 godina

Rješenje o odobrenju teme diplomskog rada

## Sažetak

U ovom diplomskom radu se upoznaje i analizira novi format za kompresiju slike Webp. Webp je format za kompresiju slika i fotografija koji je nedavno razvila kompanija Google kako bi dodatno ubrzao učitavanje slikovnih sadržaja s interneta, te samim time uvelike smanjio promet na internetu. Prema istraživanjima koje je proveo Google slike i fotografija predstavljaju 65% današnjeg prometa na internetu stoga i malo poboljšanje stupnja kompresije, uz zadržavanje jednake kvalitete slike može osigurati velike uštede prilikom slanja ili pohranjivanja slikovnih informacija [1]. Webp spada u grupu formata sa gubitkom kvalitete (engl. lossy) format kao što je i JPEG. Za veću kompresiju u odnosu na JPEG format, WebP format koristi kompresiju zasnovanu na VP8 kodnom sustavu, open-source rješenju za sažimanje videa u formatu WebM.

Osim detaljne analize Webp formata, u radu će se obraditi i JPEG, kao najrašireniji format za kompresiju slike. Također, teoretski će biti obrađene i mjere za objektivnu kvalitetu slike, prvenstveno PSNR – omjer vršnog signala i šuma (engl. Peak Signal to Noise Ratio). U eksperimentalnom dijelu rada provesti će se usporedba Jpeg i Webp formata sa stajališta sposobnosti sažimanja slika i utjecaja na kvalitetu slike.

Ključne riječi: JPEG, WEBP, kompresija, slika, fotografija

## Sadržaj

Sažetak .....	4
1. Uvod .....	7
2. Kompresija slika .....	8
3. Kompresija sa gubitcima (engl. lossy) .....	11
3.1. Transformacijsko kodiranje .....	11
3.2. Vektorska kvantizacija .....	13
3.3. Fraktalno kodiranje .....	13
3.4. Prediktivno kodiranje .....	14
4. Kompresija bez gubitaka (engl. Lossless) .....	16
4.1. Run-length kodiranje .....	16
4.2. Huffmanovo kodiranje .....	17
4.3. Entropijsko kodiranje .....	18
4.4. Kodiranje područja .....	20
5. Jpeg - Joint Photographic Experts Group .....	21
2.1. Princip kompresije JPEG formata .....	22
2.1.1. Transformacija prostora boja i uzrokovanje - Downsampling .....	22
2.1.3. Podjela slike u blokove i diskretna kosinusna transformacija .....	23
2.1.5. Kvantizacija .....	25
2.1.6. Entropijsko kodiranje .....	26
6. WebP .....	28
6.1. Tehnologija .....	28
6.2. WebP kompresija s gubicima .....	30
6.3. WebP kompresija bez gubitaka .....	31
6.4. Podrška .....	32
6.5. Kritika .....	33
7. Mjere za određivanje stupnja kompresije i kvalitete komprimirane slike .....	34
7.1. SSIM indeks sličnosti struktura (engl. Structural Similarity Indeks) .....	34
7.2. PSNR – Omjer vršnog signala i šuma (engl. Peak Signal to Noise Ratio) .....	35
7.3. Stupanj kompresije (engl. compression rate) .....	35
7.4. Količina bitova po slikovnom elementu – bpp (engl. Bits per pixel) .....	36
8. Postupak eksperimenta .....	37
9. Diskusija rezultata .....	45
10. Zaključak .....	47

Popis tablica .....	49
Popis slika.....	49
Literatura.....	50

## 1. Uvod

Iz dana u dan količina podataka na internetu raste. U toj velikoj količini podatak nalaze se i slike. Kako bi se što više količina memorije koju zauzimaju slike smanjila koriste se razne vrste kompresija. Jedna od najzastupljenijih među njima je JPEG kompresija koja se već preko 15 godina smatra najpovoljnijom vrstom kompresije za slike.

2011 Google korporacija objavljuje novi format za slike koji bi prema njihovim analizama trebao postati zamjena JPEG tj. format budućnosti za slike na internetu. Format u kojem kako navode Google-ova istraživanja slike zauzimaju manje memorijskog prostora, a da zadržavaju istu ili čak bolju kvalitetu.



# Teorijski dio

## 2. Kompresija slika

Kompresija slike predstavlja nam umanjeње broja bitova grafičke datoteke (u daljnjem tekstu slike) tako da se vrlo malo ili uopće ne narušava kvaliteta slike. Kompresija nam omogućava da više slika pohranimo na jednome mjestu, da ubrzamo prijenos slika sa jednog medija na drugi ili smanjimo vrijeme učitavanja slika na internetu.

Postoji nekoliko različitih načina pomoću kojih možemo komprimirati slike. Ovisno o namjeni biraju se i načini kompresije. Za korištenje interneta uglavnom se koriste JPEG, GIF i PNG format dok se u 2010. godine pokušava uvesti novi format WebP koji je razvila Google korporacija [1]. JPEG se uglavnom koristi za fotografije dok se GIF koristi za crteže te pokretne fotografije tzv. animacije. Ukoliko postoji potreba za transparentijom tada koristimo PNG format koji ima tu mogućnost. Postoji još tehnika i formata za kompresiju slika, ali gore navedene su najzastupljenije i prihvaćene kao svjetski standardi.

Kada se govori o kompresiji neke tekstualne ili programske datoteke potrebno je biti vrlo oprezan, jer gubljenje samo jednog znaka ili jednog bita može dovesti do pogreške. Ukoliko izgubimo jedan znak u riječi „AUTO“ dolazi do pogreške i riječ će se promijeniti u „ATO“. Ako se prilikom kompresije programske datoteke izgubi dio koda vrlo vjerojatno taj program se neće moći izvršiti.

U kompresiji slika mali nedostatak ili gubljenje male količine podataka u većini slučajeva nije vidljiv. Ne postoji neka kritična točka do koje je moguće vršiti kompresiju slika, a da to bude neprimjetno. Stoga se kod kompresije slika uglavnom govori o toleranciji koliko se podataka smije izgubiti. Iz tog razloga slike možemo komprimirati više od tekstualne i programske datoteke.

Slika u Bitmap formatu (nekomprimirana slika) rezolucije obično zauzima 24 bita po slikovnom elementu (slika rezolucije 1024x786 u bitmap formatu zauzima 2,3

MB), stoga većina formata koristi neku tehniku kompresije kako bi smanjila iskorištenje memorije [2]. Većina tehnika kompresije uzimaju u obzir uzorke unutar slike koji su slični i traže reprezentativne ekvivalente koji zauzimaju manje memorije [3].

Operacijski sustavi bazirani na prozorima prikazuju datoteke na disku pomoću grafike, dok su u povijesti prikazivali pomoću komandnih linija. Napredak mnogih sistemskih operacija kao što su preuzimanje datoteka, također se može prikazati grafički. Mnoge aplikacije pružaju grafičko korisničko sučelje, koje olakšava korištenje aplikacija i grafički interpretira željeni rezultat. Računalna grafika koristi se u mnogim područjima svakodnevnog života kako bi se slikom prezentirale razne vrste složenih informacija [2]. Istina je da su u današnje vrijeme slike jako važne, ali problem s kojim se susrećemo je taj da se svakim danom stvara sve više slika koje isto tako zauzimaju sve više memorijskog prostora. U modernim uređajima slikovni elementi predstavljeni su kao 24-bitni brojevi, gdje prezentacije crvene, zelene i plave boje zauzimaju po 8 bita [2]. Takav 24 bitni slikovni element može reproducirati jednu od  $2^{24} \approx 16.78$  miliona boja. Rezultat navedenog je slika rezolucije 512x512 koja zauzima 786,432 bitova. Kod rezolucije od 1024x1024 dolazi do zauzimanja četiri puta više prostora jer sadrži 3,145,725 bitova. To je razlog zašto je kompresija slika važna i neophodna. Važna karakteristika kompresije slika je da se može izgubiti dio podataka, a da se to ne primijeti. Slika, prije svega, postoji radi ljudi te stoga kada se komprimira može izgubiti dio podataka na koje ljudsko oko nije osjetljivo [2]. To je jedna od glavnih ideja na kojoj su zasnovani mnogi algoritmi za kompresiju slika. Općenito informacije se mogu komprimirati ako predstavljaju višak. Spomenuto je nekoliko puta da se kompresija odnosi na smanjenje ili uklanjanje viška podataka[2].

Razvojni programeri i korisnici lossy metoda kompresije slika trebaju standardnu metriku za mjerenje kvalitete rekonstruiranih slika u usporedbi sa originalnim [2]. Što god bolje rekonstruirana slika liči na original to je veća standardna vrijednost date slike. Takav sustav vrijednosti bi trebao reproducirati bezdimenzionalne brojeve i takvi brojevi ne bi trebali biti osjetljivi na male varijacije u rekonstruiranoj slici. Uobičajena mjera koja se koristi u tu svrhu je PSNR (engl. Peak Signal to Noise Ratio)[2]. PSNR je jednostavno izračunati, ali ima limitiranu povezanost

sa percipiranim greškama koje primjećuje ljudski vizualni sistem. Zbog gore navedene činjenice veće vrijednosti PSNR-a impliciraju sličnije vrijednosti između rekonstruirane vrijednosti i originalne, ali one ne garantiraju da će promatraču svidjeti rekonstruirana slika.

### 3. Kompresija sa gubitcima (engl. lossy)

Kompresije sa gubitkom su načini komprimiranja slika čiji algoritmi rade na principu brisanja ili zaboravljanja nepotrebnih podataka. Ovom metodom gubi se dio informacija o slici, ali s druge strane dobijemo sliku koja zauzima manje memorijskog prostora.

Kompresije sa gubitcima temelje se na manama ljudskoga oka[2]. Ljudsko oko manje je osjetljivo na promjene u tonu boje nego na promjene u svjetlini [2]. Iz navedenog razloga se boja i svjetlina razdvajaju u različite kanale. Više se komprimiraju dijelovi slike koji manje doprinose ukupnom izgledu. Kompresije sa gubicima uzrokuju degradaciju slike u svakom koraku, ali najčešće omogućuju daleko veće omjere kompresije nego metode bez gubitaka. Također svakim sljedećim korakom kompresije slika se dodatno degradira.

JPEG je format koji koristi kompresiju sa gubicima, a ujedno je i standard za kompresiju slike sa gubitkom informacija odnosno lossy kompresiju.

Neki od algoritama koji se koriste u Lossy metodama:

1. Transformacijsko kodiranje
2. Vektorska kvantizacija
3. Fraktalno kodiranje
4. Prediktivno kodiranje

#### 3.1. Transformacijsko kodiranje

Jedan od postupaka kompresije sa gubitcima je transformacijsko kodiranje. Dva postupka koja se provode kod transformacijskog kodiranja su transformacija signala iz prostorne u frekvencijsku domenu i kvantizacija. Na diskretnim valnim oblicima transformacija slike se vrši pomoću direktne transformacije [3]. Transformacijsko kodiranje se koristi kako bi se prostorne slike u vrijednostima slikovnih elemenata pretvorile tj. transformirale u vrijednosti koeficijenata. Budući da je to linearni proces i nema izgubljenih informacija, broj proizvedenih

koeficijenata jednak je broju transformiranih slikovnih elemenata. Željeni učinak je da većina energije slike bude sadržana u nekoliko velikih transformiranih koeficijenata. Ako nastane nekoliko istih koeficijenata koji sadrže najviše energije u većini slike, te koeficijente dodatno kodiramo pomoću entropijskog kodiranja i tako gubimo dio informacija. Isto tako vrlo je vjerojatno da se manji koeficijenti grubo kvantiziraju ili izbrišu (kodiranje s gubicima) bez vidljivih oštećenja na reproduciranoj slici.

Energija slikovnog elementa može biti definirana kao kvadrat njezine vrijednosti pomnožen faktorom skaliranja

$$E_p = V_p^2 * \alpha \quad (1)$$

Slično energija transformiranog koeficijenta može biti definirana kao kvadrat njezine vrijednosti puta neki faktor skaliranja. Sa zadovoljavajućim faktorom skaliranja ukupna energija slikovnog elementa na slici će biti jednaka ukupnoj energiji u transformacijskom koeficijentu. Različite transformacije se razlikuju između sebe na tri osnovna načina koja su zanimljiva u kodiranju slika :

1. Stupanj koncentracije energije u nekoliko koeficijenata
2. Područje utjecaja nekog koeficijenta na rekonstruiranu sliku
3. Pojava i vidljivost kodiranog šuma zbog nedostatka kvantizacije koeficijenata

Kod transformacijskog kodiranja slike dimenzija NxN dijele se u manje blokove dimenzija nxn te se na svakom od njih primjenjuje transformacija. Cilj ove transformacije je dekodiranje originalnog signala, a to najčešće rezultira raspodjelom najvećeg dijela energije u nižim koeficijentima transformacije. Pomoću ovog načina mnogi se koeficijenti mogu zanemariti nakon kvantizacije.

Četiri koraka transformacijskog kodiranja su :

- Podjela slike u blokove
- Transformacija slike
- Kvantizacija koeficijenata
- Huffmanovo kodiranje

Transformacijsko kodiranje izvodi se u dva koraka. Prvi korak je segmentacija slike tj. podjela slike na dvodimenzionalne vektore. Drugi korak je transformiranje odnosno primjena izabrane transformacije.

### 3.2. Vektorska kvantizacija

Shannon je u svojem istraživanju došao do ideje vektorske kvantizacije [3]. Efikasnije kodiranje postiže se ako se promatrani uzorci kvantiziraju kao vektori, a ne kao skalari. Slika se dijeli u blokove određenih dimenzija te se blokovi uspoređuju sa skupom reprezentativnih vektora. Osnovna zamisao vektorske kvantizacije sastavljena je od pamćenja indeksa najčešćeg reprezentativnog vektora. Performanse vektorske kvantizacije ovisi o skupu reprezentativnih vektora [4]. Jako važan korak u vektorskoj kvantizaciji je dizajniranje što boljeg rječnika (*engl. Codebook*). Rječnik se generira na temelju uzorka za treniranje. Prvo se koristi algoritam za izradu početnog rječnika, te se u sljedećem koraku koristi algoritam za grupiranje (*engl. clustering*) sa kojim se dolazi do konačnog rječnika. Što je veći rječnik ostvaruju se bolji rezultati [4]. Kodiranje i dekodiranje izvode se asimetričnom brzinom. Kod kodiranja, uspoređuje se trenutni blok s vektorima iz rječnika, a kod dekodiranja čita se indeks i kopira vektor s tim indeksom na pravilno mjesto.

### 3.3. Fraktalno kodiranje

Benoit Mandelbrot je uveo termin fraktala koji specifikira geometrijski lik gdje je svaki dio tog lika usporediv s umanjenom kopijom cjeline [4]. Fraktali su nepravilni geometrijski oblici koji imaju svojstvo samosličnosti. Objekti su samoslični i kao takvi su neovisni o skaliranju. U svakoj iteraciji objekt je transformirana verzija objekta iz prošle iteracije. Često se koriste u računalnom modeliranju nepravilnih uzoraka i struktura u prirodi.

Mandelbrotov skup je skup točaka u kompleksnoj ravnini čija granica formira fraktal. Sama formula je opisana jednostavnim izrazom 3.1.

$$Z_{n+1} = Z_n^2 + c \quad (2)$$

Za kompleksni broj  $c$  se kaže da je unutar Mandlbrot skupa ako bez obzira na broj iteracija  $n$  apsolutna vrijednost od  $z(n)$  ne prijeđe neku granicu, uzimajući  $z(0)=0$ . Uvrštavajući  $c=1$  vidimo da 1 ne spada u Mandelbrot set jer niz teži u beskonačno. Ako računamo za kompleksne brojeve i crtanje u kompleksnoj ravnini primjećuje se kompleksna granica koja se s povećanjem ne pojednostavljuje.

Barnsley je prepoznao potencijal fraktala u području kompresije slika [5]. Objavom knjige *Fractals Everywhere*, fraktalno kodiranje je dobilo znatnu popularnost. Poznata slika Barnsleyeva paprat (slika 1) koju je prvi puta opisao u spomenutoj knjizi kako bi na najjednostavniji način objasnio svoje istraživanje.



*Slika 1 - Barnsleyeva paprat*

Ova vrsta kodiranja postala je popularna zbog Barnsleyove tvrdnje da je omjer kompresije 10000:1, ali navedeni omjer je postignut samo kod posebnih slučajeva i to uz ljudsku intervenciju [5]. Kompresija uz pomoć ljudskog vođenja je nepraktična i nije primjenjiva u stvarnim situacijama.

### 3.4. Prediktivno kodiranje

Prediktivno kodiranje temelji se na pretpostavci da se signal između dva uzroka relativno malo mijenja tj. da postoji korelacije između susjednih uzoraka. Korelacija može postojati u vremenu ili u prostoru i koristi se kako bi se na temelju prethodnih vrijednosti predvidjele vrijednosti uzoraka. Dobivena razlika između stvarnog i predviđenog uzorka se kodira, a niz takvih razlika tvori signal razlike.

Signal razlike ima manji raspon vrijednosti i manje se mijenja u odnosu na originalni signal, te se iz tog razloga može bolje kodirati [6]. Prediktivno kodiranje se može provoditi i unutar slike na temelju prethodno komprimiranih dijelova te iste slike. Na taj način dolazimo do puno manjeg niza podataka kojeg je potrebno kodirati te samim time i slike koja zauzima manje memorijskog prostora.



## 4. Kompresija bez gubitaka (engl. Lossless)

Kompresija bez gubitaka podataka je način sažimanja kod kojeg ne dolazi do gubitka podataka i kvalitete informacija. Drugi naziv za ovu vrstu kompresije je nezaboravna kompresija. Prilikom kompresije slike ovom metodom svi podaci slike ostaju sadržani prilikom kompresije, te se isto tako prilikom dekompresije mogu vratiti u izvorno stanje. Nakon dekompresije slika se vraća u izvorni oblik u apsolutno identičnoj formi. Svi slikovni elementi nakon dekompresije poprimaju iste vrijednosti koje su imali prije kompresije. Vrlo često programi za obradu slika koriste ovu vrstu algoritama za kompresiju. Najzastupljeniji formati koje koriste kompresiju bez gubitaka su : GIF, PNG, TIFF i BMP.

Algoritmi koji se koriste kod kompresije bez gubitaka su:

- Run-length kodiranje
- Huffmanovo kodiranje
- Entropijsko kodiranje

### 4.1. Run-length kodiranje

Ideja koja stoji iza pristupa Run-Length je sljedeća: ako se podatak  $d$  pojavljuje  $n$  uzastopnih puta na ulaznom kanalu zamjenjuje se s jediničnim parom  $nd$ .  $n$  uzastopnih ponavljanja podatka se zove run-length od  $n$ -a i taj pristup kompresiji podataka se zove *run-length encoding* ili skraćeno RLE [2]. Ovu ideju primjenjujemo za kompresiju teksta i onda za kompresiju slika.

Run-length je jednostavna metoda kodiranja koja iskorištava činjenicu da većina datoteka sadrže nizove istih vrijednosti. Run-length algoritam prolazi kroz datoteku, te kada pronađe niz od dva ili više istih znakova koji se ponavljaju ubacuje specijalne znakove (engl. token). Primjer Run-length kodiranja :

Niz znakova kojem ćemo primijeniti kodiranje: abbaabbaababbaaaabaa

Tablica 1 - Run-length kodiranje

Originalni niz	a	b	b	A	a	b	b	a	a	b	a	b	b	a	a	a	a	b	a	a
Run-length kodirano	a	b 2	a 2	b 2	a 2	b	a	b 2		a 4		b	a 2							

Run-length kodiranje lako je za implementiranje na programskoj i sklopovskoj razini. Algoritam je vrlo brz, lako se provjerava, a jedna od mana mu je ta što ima ograničene mogućnosti kompresije (dobro komprimira ako u kodu postoji velika frekvencija ponavljanja niza znakova).

#### 4.2. Huffmanovo kodiranje

Algoritam je dobio ime po svojem izumitelju D.A.Huffman-u. Temelji se na činjenici da se neki znakovi pojavljuju češće nego drugi. Na temelju navedene činjenice tj. frekvencije ponavljanja pojedinih znakova algoritam izgrađuje težinsko binarno stablo. Svaki element napravljenog stabla dobiva novu kodnu riječ koja određuje poziciju znaka u stablu. Znak sa najvećom frekvencijom ponavljanja postaje korijen stabla, te mu je pridružuje najkraća kodna riječ. Znak sa najmanjom frekvencijom ponavljanja u većini slučajeva ima kodnu riječ veću od samog znaka. Huffmanovo kodiranje spada u grupu statičkih metoda za komprimiranje.

Karakteristike : Odnos kompresije iznosi oko 1 : 2 za nekorelirane slike (nekoreliranost ili degradacija je tendencija koeficijnta transformacije da su obično nekorelirani ako su ulazni elementi korelirani), za tipične slike odnos kompresije iznosi oko 1 : 1.2 do 1 : 2.5.

## Primjer Huffmanovog kodiranja:

Tablica 2 - Huffmanovo kodiranje

Slovo	Vjerojatnost	Huffmanov kod
I	0.273	10
J	0.182	111
P	0.091	000
S	0.091	001
M	0.091	010
O	0.091	011
Ć	0.091	1100
–	0.091	1101

J O S I P \_ M I J I Ć  
111 011 001 10 000 1101 010 10 111 10 1100

Byte-ovi

byte 1                      byte 2                      byte 3                      byte 4  
11101100                      11000011                      01010101                      11101100

### 4.3. Entropijsko kodiranje

Entropijsko kodiranje je oblik kodiranja u kojeg ne unosimo gubitke. Temelj tog oblika kodiranja je ulazni niz podataka. Ako ulazni podaci imaju promjenjivu razdiobu onda je njihova entropija manja od  $N$ . U tom slučaju postoji kod koji koristi manje od  $N$  bitova po koeficijentu.

Entropijsko kodiranje zasniva se na statistici ulaznog niza podataka. Ako ulazni podaci nemaju uniformnu distribuciju onda je njihova entropija manja od  $N$  [7]. Tri najčešće metode za entropijsko kodiranje su Huffmanovo, aritmetičko

kodiranje i Lempel-Ziv kodiranje. Lemple-Ziv kodiranje se zasniva na tome da koder i dekoder imaju jednak rječnik meta simbola. Svaki rječnik predstavlja slijed ulaznih znakova. Ako se slijed ponovi nakon što je pronađen simbol za nju slijed se zamjeni tim simbolom. Rječnik je sadržan u koderu i dekoderu te se ne mora nalaziti i u kodiranim podacima. Negativna karakteristika koja entropijskog kodiranja je relativno problematičan postupak implementacije zbog tablica koje rastu s izvođenjem algoritma. Primjer kodiranja kodnom tablicom :

Niz znakova kojem ćemo primijeniti kodiranje: abbaabbaababbaaaabaa

Tablica 3 - Kodiranje kodnom tablicom

Originalni niz	a	b	b	a	a	b	b	a	a	b	a	b	b	a	a	a	a	b	a	a
LZ kodirano	0	1	1	0	2	4	2	6	5	5	7									

Početni riječnik	kod	zapis
	0	a
	1	a
Generirani riječnik	2	a b
	3	b b
	4	b a
	5	a a
	6	a b b
	7	b a a
	8	a b a
	9	a b b a
	10	a a a
	11	a a b

#### 4.4. Kodiranje područja

Kodiranje područja poboljšana je verzija run-length kodiranja. Temelji se na iskorištavanju dvodimenzionalnih karakteristika slika. Kodiranje područja se rabi kada nam je određeni sadržaj slike važniji od ostalog tj. kada govorimo o nekom objektu na slici i pozadini tog objekta. U većini slučajeva pozadina koja se nalazi iza promatranog objekta nam nije toliko važna, nego je važan promatrani objekt. Glavna zadaća kodiranja područja je ta da izdvoji promatrani objekt na slici te da ga označi važnim i kodira ga višom kvalitetom ili kvalitetom bez gubitaka u odnosu na pozadinu objekta [8]. MAXSHIFT je metoda koju koristi JPEG2000 za kodiranje područja od interesa. MAXSHIFT metoda proširuje metodu skaliranja i pomicanja. Algoritam kodiranja područja ima zadaću pronaći pravokutne dijelove jednakih karakteristika te ih kodirati u opisanoj formi kao elemente s dvije točke i određenim strukturama. Slika mora biti opisana tako da bi se omogućilo dekodiranje bez gubitaka. Performanse algoritma se temelje na pronalaženju najvećih područja jednakih karakteristika što predstavlja vrlo kompleksan problem. Kodiranje područja je efikasan postupak kompresije. Nedostatak mu je nemogućnost sklopovske implementacije zbog nelinearnosti, te je zbog toga relativno spor.

## 5. Jpeg - Joint Photographic Experts Group

JPEG je predstavljen kao prva međunarodna norma za kompresiju nepokretnih slika [9]. Koristi se za kompresiju statičnih slika i spada u skupinu „intraframe“ metoda za kompresiju. Postoji sličan standard MPEG koji se bavi kompresijom pokretnih slika „interframe“. JPEG sadrži dvije osnovne metode za kompresiju slika. Prva metoda koristi DCT (diskretna kosinusna transformacija) i spada u skupinu kompresije sa gubitcima „lossy“ [9]. Druga metoda temelji se na prediktivnom kodiranju i spada u skupinu kompresije bez gubitaka „lossless“. Metoda temeljena na diskretnoj kosinusnoj transformaciji je osnovna metoda (Baseline metoda) te se najčešće upotrebljava.

Osnovna metoda JPEG kompresije zasnovana na DCT je metoda sa gubitcima što znači da dekomprimirana slika nije potpuno jednaka originalu kojeg smo komprimirali. Osmišljen je tako da iskorištava ograničenja ljudskog oka. Vodi se činjenicom da je ljudsko oko osjetljivije na promjene u svjetlini nego na promjene u boji. Međutim, mogu nastati i određeni problemi ukoliko ovako obrađene slike analizira računalo, a ne čovjek.

Jedna od važnih osobina JPEG metode je mogućnost biranja kompromisa između kvalitete slike i njene veličine. Ako želimo dobru kvalitetu slike stupanj kompresije nam je ograničen, ali ako nam kvaliteta nije toliko važna možemo postići visok stupanj kompresije. Još jedna od važnih činjenica vezana uz kvalitetu slike je ta da možemo birati između kvalitete slike i brzine njenog dekodiranja. Koriste se manje točne aproksimacije, ali zato jako brze i obrnuto [2].

Temeljni zahtjevi koje JPEG mora ispunjavati su da mora zadovoljavati širi raspon kvaliteta slika te omogućiti određivanje željenog odnosa između kvalitete slike i kompresije. Ne smije ga ograničavati dimenzija slike, dubina boja i omjer visina/širina tj. mora biti primjenjiv na sve vrste slika. Ne smije biti previše kompleksan kako bi mogao implementirati što više procesa uz prihvatljive performanse računala.

JPEG je predviđen za kodiranje slika iz prirode, te se zasniva na činjenicama da većina korisnog sadržaja slike se mijenja kada se prostorno promatra, ali relativno sporo, tj. nema velikih promjena intenziteta unutar blokova 8x8 slikovnih elemenata što se vidi na slici 5.



*Slika 2- promjena intenziteta unutar bloka*

Gledajući iz područja prostornih frekvencija, niže prostorne frekvencija sadrže puno više podataka od viših prostornih frekvencija koje uglavnom predstavljaju manje bitne podatke i smetnje. Mnoga istraživanja su pokazala da je ljudsko oko osjetljivije na gubitak visokih prostornih frekvencija i da dio podataka na slici prosječnom promatraču nije vidljiv [9].

## 2.1. Princip kompresije JPEG formata

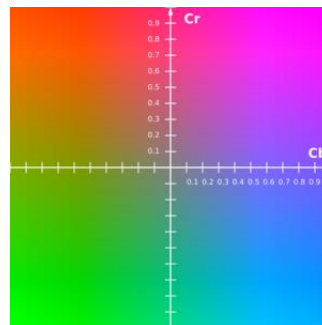
Postoji više princip kompresije JPEG formata. Osnovni princip se najčešće sastoji se od sljedećih koraka [10]:

- Transformacija prostora boja
- Uzrokovanje – Downsampling
- Podjela slike u blokove
- Direktna kosinusna transformacija (DCT)
- Kvantizacija
- Entropijsko kodiranje

### 2.1.1. Transformacija prostora boja i uzrokovanje - Downsampling

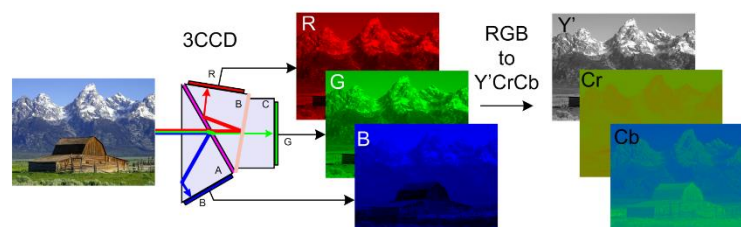
Kao što je predhodno u tekstu navedeno, ljudsko oko je osjetljivije na promjene u luminanciji nego na promjene u tonu boje. Stoga je za JPEG kompresiju potrebno koristiti prostor boja u kojemu je informacija o luminanciji odvojena od

informacija o boji [10]. Za slike u boji JPEG koristi YCbCr prostor boja, dok se za crno-bijele slike ovaj korak preskače. U YCbCr podatci o luminanciji su zapisani u Y kanalu dok su informacije o boje zapisane u Cb i Cr kanalima.



Slika 3 - YCbCr prostor boja

Nakon odvajanja informacije o luminanciji uz pomoć YCbCr prostora boja ostala su nam dva kromatska kanala nad kojima možemo izvršiti downsampling – uzorkovanje kojim se smanjuje prostorna rezolucija. Najčešće se vrši redukcija sa faktorom 2 u horizontalnom i vertikalnom smjeru (2h2v ili 4:1:1 downsampling). No postoje i drugi načini, recimo 4:2:2 ili 2h1v gdje se kromatski kanali reduciraju faktorom 2 horizontalno, a 4:4:4 predstavlja slučaj kada se downsampling ne radi [10].

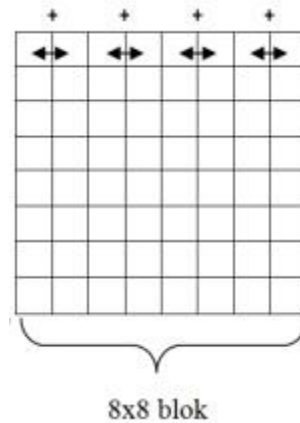


Slika 4 - Usporedba RGB i YCbCr slike po kanalima

### 2.1.3. Podjela slike u blokove i diskretna kosinusna transformacija

Prije transformacije, slika se podjeli u blokove po  $8 \times 8$  elemenata slike [10]. Ukoliko ukupan broj elemenata slike u retku ili stupcu nije djeljiv s 8, prije podjele u blokove posljednji donji redak elemenata slike, odnosno posljednji desni stupac elemenata slike kopira se onoliko puta koliko je potrebno da se dobije broj djeljiv s 8.





Slika 5 - 8x8 blok

U koder ulaze uzorci slike grupirani u blokove 8x8 slikovnih elemenata koji su pretvoreni iz cjelobrojnog broja bez predznaka u intervalu  $[0, 2^P - 1]$  u cjelobrojni oblik s predznakom s opsegom  $[-2^{P-1}, -2^{P-1}-1]$ , te dovedeni na ulaz bloka diskretne kosinuse transformacije. Jednadžba u nastavku (3), (4) prikazuju matematički prikaz 8x8 DCT i 8x8 IDCT (inverzna diskretna kosinusna transformacija) :

$$F(u, v) = \frac{1}{4} C(v) \left[ \sum_{x=0}^7 \sum_{y=0}^7 f(x, y) \cos \frac{(2y+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right] \quad (3)$$

8X8 DTC

$$F(u, v) = \frac{1}{4} \left[ \sum_{x=0}^7 \sum_{y=0}^7 C(u)C(v)F(u, v) \cos \frac{(2y+1)u\pi}{16} \cos \frac{(2y+1)v\pi}{16} \right] \quad (4)$$

8X8 IDTC

Gdje je :  $C(u), C(v) = \frac{1}{\sqrt{2}}$  za  $u, V = 0$

$C(u), C(v) = 1$  inače.

Kao diskretni signal možemo promatrati svaki blok 8x8 s 64 elementa koji je funkcija prostorne dimenzije x i y. Tako oblikovani signal dolazi na ulaz DCT, a na izlazu dobijemo 64 DCT koeficijent. Dobiveni signal predstavlja jednu od 64

jednodimenzionalnih 2D prostornih frekvencija te predstavlja spektar ulaznog signala [11]. DC koeficijent je onaj koeficijent koji ima obje prostorne nule, a ostalih 63 koeficijenta nazivaju se izmjenični koeficijenti. Vrijednosti uzorka u 8x8 bloku slabo variraju tj. vrijednosti su kolerirane za sve realne slike. Na toj činjenici se zasniva DCT kompresija jer od takvih vrijednosti usmjerava koeficijente u niže prostorne frekvencije. Iz tog razloga većina koeficijenata ima vrijednost blizu nule ili nula i nije ih potrebno kodirati.

U dekoderu iz 64 DCT koeficijenta rekonstruira se originalni blok 8x8. DCT je kopija između prostorne i frekvencijske domene. Ako DCT i IDCT izračunavamo u beskonačnu preciznost, a pritom ne obavljamo kvantizaciju, 8x8 blok bio bi jednak originalu nakon rekonstrukcije. DCT ne uzrokuje gubitke u dijelovima slika nego ih transformira u domenu gdje će biti efikasnije kodirani.

Na svakom se bloku provodi diskretna kosinusna transformacija. U ovom koraku postoji mogućnost gubitka jednog dijela informacije o slici zbog ograničene aritmetičke preciznosti procesora. Taj gubitak je vrlo mali i ima zanemariv utjecaj na rekonstrukciju slike [10].

#### 2.1.5. Kvantizacija

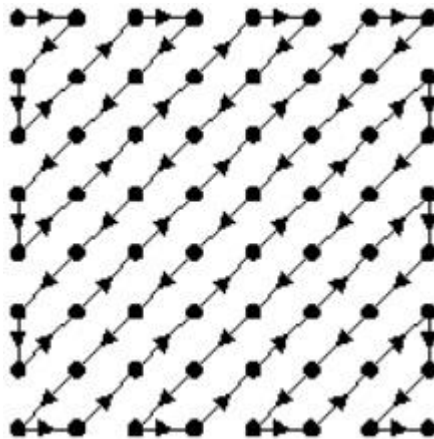
Nakon DCT svaki blok od 64 koeficijenta je pojedinačno kvantiziran ovisno o pripadajućem članu iz tablice kvantiziranja koja također sadrži 64 elementa [11]. Kvantizacijska tablica mora biti usklađena sa aplikacijom ili je korisnik mora sam uskladiti. Bilo koji element tablice može biti broj od 1 do 255 te određuje veličinu koraka kvantizacije za odgovarajući koeficijent DCT. Postizanje kompresije je zadaća kvantizacije, tj. da se koeficijenti DCT ne reprezentiraju sa većom preciznošću nego što je potrebno kako bi se postigla željena kvaliteta slike. Cilj ovog koraka je da se uklone podaci koji ne doprinose vizualnom poboljšanju slike. Stoga je osobina kvantizacije gubitak informacija nakon dekvantizacije te je to osnovni razlog gubitka informacija kod JPEG kompresije zasnovane na diskretnoj kosinusnoj transformaciji [11].

Ukoliko nam je cilj što jače komprimirati sliku, ali bez vidljivih odstupanja, svaki element kvantizacijske tablice treba posebno obraditi kako bi se definirao prag

kod kojega počinju biti vidljiva odstupanja. Vrijednosti pragova su ovisne o karakteristikama slike, karakteristikama zaslona na kojemu se slika reproducira te udaljenosti promatranja slike. Udaljenosti se najčešće određuju psihovizualnim eksperimentima. JPEG standard ima niz kvantizacijskih tablica koje su definirane gore navedenim eksperimentom. Upotreba unaprijed definiranih tablica nije obavezna, ali se nalaze u standardu kao informacije [11].

#### 2.1.6. Entropijsko kodiranje

Posljednji korak kod JPEG kompresije temeljenoj na DCT-u je entropijsko kodiranje. Ovaj korak dodatno povećava stupanj kompresije, ali ne utječe na daljnje gubljenje podataka o slici. Entropijskim kodiranjem se zapravo efikasnije kodiraju DC koeficijenti na temelju njihovih statičkih karakteristika. Dvije metode koje koriste kod JPEG-a u entropijsko kodiranja su Huffmanova metoda i aritmetičko kodiranje. Osnovna metoda koja se koristi je Huffmanova metoda. 64 koeficijenta u matrici svakog bloka poredaju se „zig-zag“ skeniranjem [10].



Slika 6- Zig-zag skeniranje

Navedeni postupak provodi se kako bi se brojevi poredali tako da oni koji su različiti od 0 smjestili na početak niza, a nule nakon njih. Zatim se nad nizom provodi Run Length Encoding (RLE) algoritam koji kodira vrijednosti bita i broj uzastopnih ponavljanja tog bita. Posljednji korak je primjena Huffmanovog algoritma koji osigurava da se dobiveni niz kodira sa što kraćom duljinom bitova. Ideja Huffman-ovog kodiranja je ta da brojevi imaju kodove različitih dužina tj.

brojevi koji se ponavljaju češće imaju kraće kodne riječi, a oni koji se ponavljaju rjeđe imaju duže kodne riječi.

Aritmetičko kodiranje za razliku od Huffmanovog kodiranja ne zahtjeva dodatne tablice jer se može adaptirati na statičke značajke slike za vrijeme procesa kodiranja. Aritmetičko kodiranje daje 5-10% veći stupanj kompresije u usporedbi sa Huffmanovim kodiranjem, ali je znatno kompleksnije i zbog toga nije pogodno za sklopovsku implementaciju [11].

## 6. WebP

WebP (čita se „weppy“) je format slike koji pruža kompresiju s gubicima i kompresiju bez gubitaka. WebP baziran je na tehnologiji On2 Technologies, koju je Google kupio i danas je sam razvija. Kao izvod iz video formata VP8, WebP je sestrinski projekt multimedijalnom container formatu WebM. WebP računalna podrška objavljena je pod BSD (engl. Berkeley Software Distribution) licencom [1].

Format je prvotno najavljen u 2010. i od strane programera prezentiran kao novi otvoreni standard za kompresiju bez gubitka true-color grafike na webu, koji je proizvodio manje datoteke i jednako dobru kvalitetu slike kao starija JPEG shema. Trećeg listopada 2011. Google objavljuje WebP sustav za podržavanje animacija, ICC profile, XM metapodataka i popločavanje (sklapanje vrlo velikih slika do maksimalno 16384x16384 slikovnih elemenata.

18. studenoga 2011. Google počinje eksperimentirati sa kompresijom bez gubitaka i podrškom za prozirnost (alpha channel) u obje kompresije (s gubicima i bez). Podrška je omogućena kao zadana u libweb 0.2.0 (16. kolovoz, 2012.). Prema Google-ovim ranijim izračunima, pretvorba PNG-a u WebP rezultira smanjenjem od 45% veličine datoteke kada su u pitanju PNG-ovi pronađeni na webu, dok nešto manja kompresija od 28% se dobije ako su PNG-ovi bili prije komprimirani s pngcrush i PNGOUT [1].

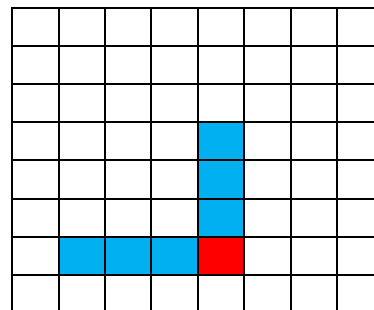
Google je predložio korištenje WebP za animirane slike kao alternativu za popularni GIF, navodeći prednosti 24-bit boja s prozirnosti, mogućnost kombiniranja okvira kompresiranih s gubicima i bez gubitaka u istoj animaciji, i isto tako podršku za traženje određenih okvira. Prema njihovim izvještajima pretvorba animiranoga GIF-a u WebP s gubicima rezultira smanjenjem od 64% dok pretvorba u Webp bez gubitaka smanjuje veličinu za 19% [1].

### 6.1. Tehnologija

WebP kompresijski algoritam s gubicima baziran je na intra-frame kodiranju VP8 video formata i klasičnoga Resource Interchange File Format (RIFF) kao

sadržajnog (container) formata. Kao takva transformacijska shema bazirana na blokovima sa 8 bitnom dubinom boja i luminance-chrominance modelom sa chrome pod uzrokovanjem pri omjeru 1:2 (YcbCr 4:2:0). Bez daljnega sadržaja mandatorni RIFF container rezultira dodatkom od samo 20 byte-a i može držati dodatne metapodatke. Dužina stranice WebP slika limitirana je na 16383 slikovnih elemenata tj. 14 bita dok je dužina stranice kod JPEG formata limitirana na 65535 slikovnih elemenata.

WebP kompresija zasnovana je na blokovima. Svaki blok se predviđa u odnosu na vrijednosti tri bloka iznad istoga i tri bloka lijevo od istoga (dekodiranje bloka se izvodi u raster-scan redu: s lijeva na desno i od vrha prema dnu) (Slika 4).



Slika 7 - Predviđanje blokova

Pogrešno predviđeni podaci i nepredviđeni blokovi komprimiraju se u 4x4 slikovna elementa po bloku s diskretnom kosinusnom transformacijom i (rijetko) Walsh-Hadamard transformacijom. Obje transformacije se obavljaju s fiksirane točke aritmetike da se izbjegnu ponavljajuće pogreške. Izlaz je komprimiran s entropijskim kodiranjem.

WebP-ova kompresija bez gubitaka koristi napredne tehnologije kao što su entropijsko kodiranje za različite kanale boja, iskorištavanje 2D lokaliteta povratnih referentnih udaljenosti i boja koje su predhodno spremljene u privremenoj memoriji. Ovo kompletira temeljne tehnike kao što su kodiranje riječnika, Huffmanovo kodiranje i indeksno transformiranje boja. Slika dobivena WebP kompresijom bez gubitaka zauzima 26% manje memorije nego ista slika u PNG formatu [1].

Prediktivno kodiranje koristi vrijednost u susjednim blokovima slikovnih elemenata kako bi predvidjeo vrijednost bloka, a zatim kodira samo razliku

(ostatak) između stvarne vrijednosti i predviđene. Ostatak koda obično sadrži mnogo nula koje se mogu vrlo učinkovito komprimirati. Rezultat se zatim transformira, kvantizira, te se izvrši entropijsko kodiranje. WebP također koristi varijable veličine bloka. WebP s gubicima daje slike koje imaju 25-34% manje SSIM indekse u odnosu na JPEG slike.

Transparentnost je jedna od opcija koju ovaj format također podržava (poznato kao alfa kanal) i to samo sa 22% dodatnih bitova. Transparentnost je također podržana kod WebP lossy formata, te WebP slika koji sadrži transparentciju zauzima tri puta manje memorije u odnosu na istu sliku u PNG formatu.

## 6.2. WebP kompresija s gubicima

Slika se dijeli na manje blokove zvane makroblokovima. U svakom makrobloku se traži suvišna informacija boje i pokreta od prethodnih makroblokova. Suvišni podatci se potom oduzimaju od bloka, a ostane samo ostatak na kojem se potom izvodi direktna kosinusa transformacija. Rezultat DCT je mnoštvo koeficijenata koji su jednaki nuli, te se oni bolje komprimiraju. Podatci se kvantiziraju i entropijski kodiraju. Jedini korak s gubitkom u primjeru WebP-a je kvantizacija.

### Intra-frame predviđanje

Intra-frame predviđanje blokova stvara tri vrste makroblokova : 4x4 svjetlina, 16x16 svjetlina, 8x8 chrominance. Kolorimetrijska razlika se reducira u omjeru 4:2:0. Veličina blokova koji sadrže podatke o svjetlini se određuje na temelju količine detalja na sceni. Za preciznije dijelove se koristi 4x4 makroblok.

Sama predikcija(predviđanje) nudi 4 načina :

H\_PRED (Horizontalna Predikcija) - svaki stupac bloka se napuni stupcem koji je susjedni s lijeve strane.

V\_PRED (Vertikalna Predikcija) - svaki redak bloka se napuni retkom koji se nalazi iznad bloka.

DC\_PRED (DC Predikcija) - cijeli blok se puni jednom bojom, koju se dobije temeljem prosjeka stupaca susjednog na lijevo, a retka iznad bloka

TM\_PRED (TrueMotion Predikcija) - koristi predikciju pokretana na temelju retka

iznad bloka, stupca s lijeva i slikovnih elemenata lijevo iznad bloka.

U makrobloku 4x4 je moguće H\_PRED i V\_PRED raširiti u još šest smjerova.

Stvoreni pretpostavljeni snimak se oduzima od realnog dijela, a sprema se samo razlika koja podliježe daljnjim procesima.

Na proizašli rezultat se primjenjuje DCT, a nekada se pod iznimkom koristi Walsh-Hadamardova transformacija. Obje transformacije se provode sa dekadskim zarezom da bi se minimalizirao gubitak informacija.

Sljedeći korak koji se provodi je adaptivna kvantizacija. Slika se dijeli do vizualno sličnih segmenata, a svaki taj segment može biti komprimiran s različitim parametrima. Dopusšteno je maksimalno četiri segmenta po slici [12].

Za entropijsko kodiranje kod ovog načina kompresije izabrano je aritmetičko kodiranje.

### 6.3. WebP kompresija bez gubitaka

Temelj kodiranja bez gubitaka su transformacije slike s korištenjem raznih tehnika. Kodirani su zamjenski parametri te tehnike s transformiranim podacima slike.

Specijalna transformacija sa predviđanjem

Transformacija s prostornom predikcijom koristi činjenicu da susjedni slikovni elementi skupa često koreliraju. Vrijednost aktualnog slikovnog elementa je pretpostavljena na osnovi već kodiranih slikovnih elemenata i kodirana je samo razlika između aktualnog i pretpostavljenog slikovnog elementa. Slika se dijeli na nekoliko regija i u okviru regija se može koristiti samo jedna vrsta predikcije. Vrste predikcije se dijele prema smjeru predikcije opcionalno, a nekada koriste prosječnu vrijednost.

Transformacija prostora boja

Transformacija prostora boja transformira RGB vrijednost svakog slikovnog elementa. Zelena mapa se zadrži u osnovnom obliku. Crvena mapa se transformira na osnovu zelene, a plava se prvo transformira na osnovu zelene, a potom crvene mape.



### *Transformacija oduzimanjem zelene*

Ova transformacija oduzima zelenu mapu od crvene i plave mape.

### *Transformacija indeksiranjem boja*

Tehnika indeksiranja u webP formatu je jako slična tehnici indeksiranja u PNG formata. Ako slika koristi manje od 256 boja onda se stvara paleta, a uzorak se odnosi na indeks u paleti .

### *Kodiranje*

Nakon transformacije dolazi do samog kodiranja. Za kodiranje se koristi LZ77 i Huffmanovo kodiranje [12].

## 6.4. Podrška

Gledajući web preglednike, Google Chrome i Opera podržavaju WebP. Svi WebM kompatibilni preglednici mogu također prikazati WebP preko JavaScript-a. WebP se može prikazati na većini preglednika koji koriste WebPJS JavaScript knjižnicu (IE6+podrška je osigurana kroz korištenje Flasha).

Kada su u pitanju grafički računalni programi, Picasa (od verzije 3.9), Photline, Pixelmator, ImageMagick, Konvertor, XnView, IrfanView i GDAL svi podržavaju WebP. Telegraphics objavio je besplatan plug-in koji omogućuje WebP podršku za Adobe Photoshop. GIMP i Paint.NET podržavaju WebP preko plug-inova. Google je objavio plug-in za Microsoft Windows koji omogućuje WebP podršku za Windows Photo Viewer, Microsoft Office 2010, FastPictureViewer i bilo koju drugu aplikaciju koja koristi Windows Imaging Component.

Gmail i Picasa Web Albums (Google-ove aplikacije) podržavaju WebP. Podrška za WebP planirana je i za Google App Engine. Opciju Instant Previews Google Searcha trenutno koristi WebP interno, da bi smanjili prostor na disku korišten kod pregleda. Android 4.0 podržava encoding i dekodiranje WebP slika (preko bitmapa i Skia-e). SDL\_image podržava format od 1.2.2011.

## 6.5. Kritika

*U rujnu 2010., Fiona Glaser, x264 encoder programer, napisala je vrlo ranu kritiku na račun WebP. Uspoređujući različite encodinge (JPEG, x264 i WebP) slike uzorka, zaključila je da je encoding WebP-a bio najgori od gore navedena tri, poglavito zbog previše blura na slici [1].*

## 7. Mjere za određivanje stupnja kopresije i kvalitete komprimirane slike

### 7.1. SSIM indeks sličnosti struktura (engl. Structural Similarity Indeks)

SSIM metoda koristi se u svrhu mjerenja sličnosti između dvije slike. SSIM indeks se može upotrijebiti kao mjera za kvalitetu kada uspoređujemo slike, pod uvjetom da su se na jednoj slici vršile neke promjene, a druga slika je kvalitetni original.

Uzroci signala slike pokazuju veliku međusobnu korelaciju koja je ujedno pokazatelj količine informacija sadržane u strukturi objekta obuhvaćenih određenom scenom [13]. Osnovna ideja pristupa određivanja sličnosti struktura zasniva se na činjenici da ljudski vizualni sustav vrlo uspješno izdvaja informacije vezane za strukture u slici pa se zbog toga može upotrijebiti kod mjerenja sličnosti struktura kao dobra aproksimacija subjektivnog vrednovanja kvalitete slike. Kako bi na gore navedeni način pristupili u određenom algoritmu moramo odgovoriti na dvije stvari [13]. Kako definirati strukturna i nestrukturna izobličenja i kako ih izdvojiti. Odgovori na ovo pitanje nalaze se u metodi indeksa sličnosti struktura. SSIM metoda se pokazala vrlo uspješnom kod statičnih slika komprimiranih prema normama JPEG i JPEG2000 kod kojih su greške uzrokovane različitim stupnjem kompresije vrlo različite [13]. Vrijednost SSIM indeksa u veličine u skali od 0 do 1. Ukoliko je SSIM indeks jednak 1 to znači da je komprimirana slika zadržala sve podatke o slici, te da je kvaliteta komprimirane slike jednaka originalnoj kvaliteti. Dobiveni indeksi koji imaju vrijednosti koje su različite od 1 (npr. 0,99030) daju nam podatak koliko je podataka slika izgubila prilikom kompresije tj. koliko je slika izgubila na kvaliteti u odnosi na idealnu sliku koja ima vrijednost SSIM indeksa jednaku 1.

### 7.2. PSNR – Omjer vršnog signala i šuma (engl. Peak Signal to Noise Ratio)

PSNR je objektivna mjera za određivanje kvalitete slike. Definirana je kao omjer maksimalne moguće vrijednosti signala i šuma u signalu [10]. Ako je originalna slika  $f(x,y)$ , a komprimirana slika  $g(x,y)$  tada je PSNR definiran kao :

$$PSNR = 10 \cdot \log \left( \frac{MAX^2}{\frac{1}{mn} \sum_{x=0}^{m-1} \sum_{y=0}^{n-1} \|g(x,y) - f(x,y)\|^2} \right) \quad (4)$$

MAX predstavlja najveću moguću vrijednost koju element slike može imati, a m i n predstavljaju dimenzije slike [10]. Vrijednost PSNR-a je izražena u decibelima.

Ako izračunom dobijemo rezultat od 40 dB smatramo da nam je razlika između izvorne i komprimirane slike mala tj. da je komprimirana slika zadržala dobru kvalitetu. Ukoliko dobijemo vrijednosti ispod 20 dB znači da nam je razlika između originalne i komprimirane slike velika tj. da smo komprimiranjem dobili sliku loše kvalitete [10].

### 7.3. Stupanj kompresije (engl.comprasion rate)

Stupanj kompresije ili skraćeno CR je broj koja nam govori kolika komprimirana slika zauzima manje memorijskog prostora u odnosu na originalnu sliku. CR dobijemo tako da se broj bitova originalne slike  $N_{orig}$  podjeli sa brojem bitova po slikovnom elementu komprimirane slike  $N_{komp}$  (6). Štp je ovaj omjer veći to je i stupanj kompresije veći, a to znači da će komprimirana slika zauzimati manje prostora.

$$CR = \frac{N_{orig}}{N_{komp}} \quad (5)$$

#### 7.4. Količina bitova po slikovnom elementu – bpp (engl. Bits per pixel)

Slikovni elementi su u zaslonima su posloženi tako da formiraju dvodimenzionalno polje na kojem se fotografija prikazuju tako da se vrijednost pojedinog slikovnog elementa postavi na određenu boju. Broj boja koje slikovni element može reproducirati ovisi o broju bita po slikovnom elementu (bpp). Slike koji imaju vrijednost  $\text{bpp}=1$  (jednobitne slike) mogu reproducirati samo crnu i bijelu boju, a slike koje imaju vrijednost  $\text{bpp}=2$  (dvobitne slike) mogu reproducirati 4 boje. Povećanjem broja  $\text{bpp}$ -a povećava se kvaliteta slike, te količina memorije koja je potrebna da se slika pohrani [14].  $\text{bpp}$  se izračunava tako da se količina memorije slike podjeli sa ukupnim brojem slikovnih elemenata slike (jednadžba (6)).

$$\text{bpp} = \frac{\textit{količina memorije originalne slike}}{\textit{ukupan broj slikovnih elemenata po slici}} \quad (6)$$

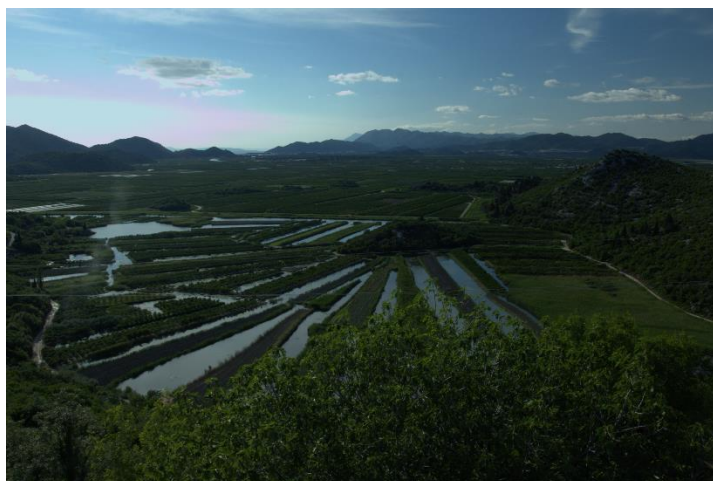
# Eksperimentalni dio

U eksperimentalnom dijelu rada provesti će se usporedba Jpeg i Webp formata. Eksperiment je podijeljen u dva dijela, usporedbi stupnja kompresije formata JPEG i formata Webp, te određivanju utjecaja navedenih formata na kvalitetu originalne slike korištenjem PSNR i SSIM mjera za objektivnu procjenu kvalitete slike.

Program koji je korišten za kompresiju slika je XnView.

## 8. Postupak eksperimenta

Dvadeset slika iz izvornog CR formata rezolucije 3888X2592, teme pejzaža i prirode, pomoću programa XnView pretvoreno je u BMP format radi lakše daljnje obrade. Pretvoreni BMP format slike odabran je kao novi izvorni format slika nad kojima će se u nastavku rada provodi eksperiment. Dvadeset slika iz izvornog BMP formata komprimirano je pomoću JPEG algoritma sa stupnjevima kompresije 1,50 i 100. Tablica 5 prikazuje količinu memorije koju zauzimaju slike u izvornom BMP formatu izražene u kilobajtima (KB), te rezultati kompresije 1(JPEG 1),50(JPEG 50) i 100(JPEG 100) JPEG algoritma također izražen u KB.



Slika 7 - Slika u izvornom BMP formatu

Tablica 4 - Rezultati JPEG kompresije

Originalna slika	BMP [KB]	JPEG 1 [KB]	JPEG 50 [KB]	JPEG 100 [KB]
Slika 1	29525	166	487	5148
Slika 2	29525	176	723	6088
Slika 3	29525	159	194	4389
Slika 4	29555	167	366	6400
Slika 5	29555	173	461	5957
Slika 6	29555	167	339	6391
Slika 7	29555	169	607	7152
Slika 8	29555	167	617	7463
Slika 9	29548	187	931	8217
Slika 10	29555	163	508	6160
Slika 11	29555	171	805	7952
Slika 12	29555	169	596	7532
Slika 13	29525	165	430	4887
Slika 14	29595	182	638	5701
Slika 15	29525	172	601	5475
Slika 16	29525	169	522	5582
Slika 17	29525	178	746	6221
Slika 18	29525	177	704	6031
Slika 19	29525	163	522	5859
Slika 20	29555	204	893	7565

Isti postupak je napravljen sa slikama za WebP format. Slike iz izvornog BMP formata komprimirane su u WebP format sa stupnjem kompresije 1,50 i 100. U tablici 6 su prikazani podatci koji nam prikazuju količinu memorije koju zauzimaju slike nakon WebP kompresije.

Tablica 5 - Rezultati WebP kompresije

Originalna slika	BMP [KB]	WebP 1 [KB]	WebP 50 [KB]	WebP 100 [KB]
Slika 1	29525	58	209	8738
Slika 2	29525	100	374	9969
Slika 3	29525	24	33	7406
Slika 4	29555	49	134	10308
Slika 5	29555	70	201	9764
Slika 6	29555	47	120	10388
Slika 7	29555	72	301	11228
Slika 8	29555	66	307	11437
Slika 9	29548	156	610	12372
Slika 10	29555	45	188	9886
Slika 11	29555	89	442	12049
Slika 12	29555	69	283	11890
Slika 13	29525	53	175	8116
Slika 14	29595	99	317	9431
Slika 15	29525	85	311	8992
Slika 16	29525	71	242	9217
Slika 17	29525	105	390	10372
Slika 18	29525	100	362	10111
Slika 19	29525	45	194	9919
Slika 20	29555	134	469	10704





*Slika 8 - Usporedba WebP 1 i JPEG 1*



*Slika 9 - Usporedba WebP 50 i JPEG 50*



*Slika 10 - Usporedba WebP 100 i JPEG 100*

U sljedećem koraku izračunat je stupanje kompresije (engl. compression rate) kako bi dobili podatak koliko komprimirana slika sa određenim stupnjem kompresije zauzima manje memorijskog prostora od izvorne slike. Izračun je

napravljen tako da je podijeljena veličina izvorne slike u KB sa veličinom komprimirane slike u KB (jednadžba 6).

Tablica 6 - Stupanj kompresije za pojedinu sliku

JPEG1	JPEG50	JPEG100	Webp1	Webp50	Webp100	Originalna slika
177,861	60,626	5,735	509,052	141,268	3,379	Slika 1
167,756	40,837	4,850	295,250	78,944	2,962	Slika 2
185,692	152,191	6,727	1230,208	894,697	3,987	Slika 3
176,976	80,751	4,618	603,163	220,560	2,867	Slika 4
170,838	64,111	4,961	422,214	147,040	3,027	Slika 5
176,976	87,183	4,624	628,830	246,292	2,845	Slika 6
174,882	48,690	4,132	410,486	98,189	2,632	Slika 7
176,976	47,901	3,960	447,803	96,270	2,584	Slika 8
158,011	31,738	3,596	189,410	48,439	2,388	Slika 9
181,319	58,179	4,798	656,778	157,207	2,990	Slika 10
172,836	36,714	3,717	332,079	66,867	2,453	Slika 11
174,882	49,589	3,924	428,333	104,435	2,486	Slika 12
178,939	68,663	6,042	557,075	168,714	3,638	Slika 13
162,610	46,387	5,191	298,939	93,360	3,138	Slika 14
171,657	49,126	5,393	347,353	94,936	3,283	Slika 15
174,704	56,561	5,289	415,845	122,004	3,203	Slika 16
165,871	39,578	4,746	281,190	75,705	2,847	Slika 17
166,808	41,939	4,896	295,250	81,561	2,920	Slika 18
181,135	56,561	5,039	656,111	152,191	2,977	Slika 19
144,877	33,096	3,907	220,560	63,017	2,761	Slika 20
<b>172,08</b>	<b>57,52</b>	<b>4,810</b>	<b>461,300</b>	<b>157,580</b>	<b>2,970</b>	<b>Prosječna vrijednost</b>

Radi lakše usporedbe i preciznijih rezultata izračunana je i srednja vrijednost stupnja kompresije za svih dvadeset slika te se taj podatak isto nalazi u tablici 7. Nakon određivanja stupnja kompresije jedan od bitnih podataka kod usporedbe algoritama za kompresiju je podatak koliko je bita potrebno da se opiše jedan slikovni element . Podatak o količini bitova po slikovnom elementu pomoću formule (7). Radi lakše analize podataka izračunata je i srednja vrijednost bpp-a za sve slike.

Rezultati dobiveni izračunom za sve slike nalaze se u tablici 7.

Tablica 7 – bbp indexi

Slika 1	24,00	0,13	0,40	4,18	0,05	0,17	7,10
Slika 2	24,00	0,14	0,59	4,95	0,08	0,30	8,10
Slika 3	24,00	0,13	0,16	3,57	0,02	0,03	6,02
Slika 4	24,00	0,14	0,30	5,20	0,04	0,11	8,38
Slika 5	24,00	0,14	0,37	4,84	0,06	0,16	7,94
Slika 6	24,00	0,14	0,28	5,20	0,04	0,10	8,44
Slika 7	24,00	0,14	0,49	5,81	0,06	0,24	9,13
Slika 8	24,00	0,14	0,50	6,07	0,05	0,25	9,30
Slika 9	24,00	0,15	0,76	6,68	0,13	0,50	10,06
Slika 10	24,00	0,13	0,41	5,01	0,04	0,15	8,04
Slika 11	24,00	0,14	0,65	6,46	0,07	0,36	9,79
Slika 12	24,00	0,14	0,48	6,12	0,06	0,23	9,67
Slika 13	24,00	0,13	0,35	3,97	0,04	0,14	6,60
Slika 14	24,00	0,15	0,52	4,63	0,08	0,26	7,67
Slika 15	24,00	0,14	0,49	4,45	0,07	0,25	7,31
Slika 16	24,00	0,14	0,42	4,54	0,06	0,20	7,49
Slika 17	24,00	0,14	0,61	5,06	0,09	0,32	8,43
Slika 18	24,00	0,14	0,57	4,90	0,08	0,29	8,22
Slika 19	24,00	0,13	0,42	4,76	0,04	0,16	8,06
Slika 20	24,00	0,17	0,73	6,15	0,11	0,38	8,70
<b>Srednja vrijednost</b>	<b>24,00</b>	<b>0,14</b>	<b>0,48</b>	<b>5,13</b>	<b>0,06</b>	<b>0,23</b>	<b>8,22</b>

Kako bi dobili podatak o međusobnoj sličnosti između originalne i komprimirane fotografije izračunali smo SSIM indeks. Dobiveni brojevi (tablica 8) u skali od 0 do 1 će nam dati podatak o sličnosti svih komprimiranih slika različitih kompresija sa originalnom slikom. Isto tako na dnu tablice 9 je izračunata srednja vrijednost dobijenih vrijednosti SSIM indeksa za pojedine vrste kompresije. Izračun SSIM indeksa je napravljen pomoću programa MatLab koji ima ugrađenu opciju za izračun SSIM indeksa.

Tablica 8 -SSIM indexi

	JPEG 1	JPEG50	JPEG100	WebP1	WebP50	WebP100
	SSIM	SSIM	SSIM	SSIM	SSIM	SSIM
Slika 1	0,75008	0,99030	0,99976	0,92832	0,98363	0,99972
Slika 2	0,77618	0,98883	0,99974	0,93907	0,98372	0,99973
Slika 3	0,73732	0,99112	0,99977	0,94645	0,98627	0,99977
Slika 4	0,79281	0,98966	0,99979	0,93367	0,98280	0,99979
Slika 5	0,84463	0,99374	0,99979	0,96620	0,99152	0,99975
Slika 6	0,86000	0,99471	0,99985	0,96705	0,99253	0,99983
Slika 7	0,67710	0,98900	0,99973	0,91363	0,98198	0,99969
Slika 8	0,81433	0,99244	0,99980	0,95801	0,98855	0,99979
Slika 9	0,87545	0,99631	0,99992	0,97349	0,99419	0,99993
Slika 10	0,88859	0,99503	0,99988	0,97047	0,99279	0,99988
Slika 11	0,78413	0,98673	0,99974	0,93657	0,98178	0,99942
Slika 12	0,78687	0,99242	0,99979	0,95149	0,98734	0,99976
Slika 13	0,88888	0,99497	0,99986	0,96509	0,99222	0,99964
Slika 14	0,81275	0,98888	0,99936	0,91202	0,97373	0,99912
Slika 15	0,71456	0,99249	0,99982	0,95164	0,98877	0,99982
Slika 16	0,84011	0,99415	0,99983	0,95797	0,99179	0,99983
Slika 17	0,82593	0,99156	0,99977	0,96100	0,98921	0,99977
Slika 18	0,83971	0,99202	0,99975	0,96282	0,98974	0,99977
Slika 19	0,75573	0,98982	0,99972	0,94885	0,98636	0,99973
Slika 20	0,89980	0,99693	0,99990	0,97354	0,99569	0,99991
<b>Srednja vrijednost</b>	<b>0,8083</b>	<b>0,9921</b>	<b>0,9998</b>	<b>0,9509</b>	<b>0,9877</b>	<b>0,9997</b>

Zadnji izračun u eksperimentalnom dijelu je izračun PSNR-a. PSNR će nam dati objektivnu mjeru u decibleima (dB) kako bi mogli odrediti kvalitetu slike. Iznosi dobiveni izračunom koji imaju vrijednost 40 dB i više smatra se da imaju dobru kvalitetu, tj. da nije velika razlika između originalne i komprimirane slike nakon kompresije. Iznosi koji imaju vrijednosti manje od 20 dB smatra se da imaju lošu kvalitetu tj da je razlika između originalne i komprimirane slike velika nakon kompresije. Dobiveni rezultati PSNR-a sa komprimirane slike sa različitim stupnjevima kompresije nalaze se u tablici 9, te srednja vrijednost za sve izračune. Izračun PSNR-a napravljeno je pomoću programa MatLab koji ima ugrađenu opciju za izračunavanje PSNR-a.

Tablica 9- Rezultati PSNR-a

	<b>JPEG 1</b>	<b>JPEG50</b>	<b>JPEG100</b>	<b>WebP1</b>	<b>WebP50</b>	<b>WebP100</b>
	<b>PSNR[dB]</b>	<b>PSNR[dB]</b>	<b>PSNR[dB]</b>	<b>PSNR[dB]</b>	<b>PSNR[dB]</b>	<b>PSNR[dB]</b>
Slika 1	24,4305	38,9	48,24729	31,12779	37,15811	48,19277
Slika 2	23,94442	37,69156	45,61242	30,62684	36,31328	45,58967
Slika 3	22,47772	33,03407	40,08138	27,78769	32,31144	40,03379
Slika 4	24,13837	34,61923	40,83517	29,74797	33,64467	40,75055
Slika 5	27,36923	40,71116	49,22304	32,2827	38,62802	49,34937
Slika 6	25,28255	37,70032	47,03942	29,46063	36,1058	46,95519
Slika 7	23,08839	36,93511	46,7716	29,8026	35,34292	46,73766
Slika 8	23,65979	37,58519	46,02012	29,78948	35,91263	45,97589
Slika 9	24,69754	37,51007	47,28493	29,76151	35,61061	47,29146
Slika 10	24,8896	37,67481	47,74162	29,26638	35,65651	47,73522
Slika 11	25,34074	37,99138	47,25707	30,59812	36,57792	45,23604
Slika 12	22,57315	36,48096	46,65667	28,773	34,59501	46,60277
Slika 13	23,15514	34,07962	41,14199	27,3392	33,50531	39,36268
Slika 14	27,47124	42,26296	48,79821	32,37745	41,10568	48,88838
Slika 15	23,41102	37,30103	43,10624	31,24661	36,54537	43,04268
Slika 16	23,71509	38,03161	45,30588	30,49346	36,75901	45,27453
Slika 17	24,66643	37,1057	42,95879	31,05446	36,37151	42,89655
Slika 18	22,07581	35,07165	42,39807	28,79987	33,99551	42,33403
Slika 19	22,29346	33,92039	40,66394	28,45962	33,16112	40,62817
Slika 20	21,92037	31,97398	39,94892	25,8876	31,41191	39,89802
<b>Srednja vrijednost</b>	<b>24,03</b>	<b>36,83</b>	<b>44,85</b>	<b>29,73</b>	<b>35,54</b>	<b>44,64</b>

Tablica 10 - Srednje vrijednosti CR,SSIM, PSNR i bpp-a

<b>Srednje vrijednosti</b>	<b>BMP</b>	<b>JPEG 1</b>	<b>JPEG 50</b>	<b>JPEG 100</b>	<b>WebP 1</b>	<b>WebP 50</b>	<b>WebP 100</b>
CR	1	172,08	57,52	4,810	461,300	157,580	2,2970
SSIM	/	0,8083	0,9921	0,9998	0,9509	0,9877	0,9997
PSNR	/	24,03	36,83	44,85	29,73	35,54	44,64
bbp	24,00	0,14	0,48	5,13	0,06	0,23	8,22

## 9. Diskusija rezultata

Promatrajući rezultate može se vidjeti da WebP ima puno veći CR tj. da ima veću kompresiju. WebP većim stupnjem kompresije ostvaruje bolje SSIM vrijednosti ali ima lošije PSNR vrijednosti za razliku od JPEG formata. Gore navedeni zaključci nam mogu odgovoriti na moguće pitanje zašto je Google u svoje istraživanju [1] naveo samo rezultate SSIM mjerenja, a ne i PSNR-a.

WebP 1 u odnosu na JPEG 1 ima puno veći CR, a puno manji bbp. Kvaliteta je kod WebP 1 puno manje narušena nego kod JPEG 1. PSNR razlika između od svega 3 dB (PSNR je logaritamska funkcija) između WebP 1 i JPEG1 daje izrazito bolje rezultate u korist WebP 1.

### *JPEG 1 i WEBP 1:*

WEBP1 dominira na JPEG1 u svim mjerenjima. Prosjek CR-a mu je znatno veći, više od 2 puta. Prosječna vrijednost SSIM-a mu je bolja, gotovo pa postiže 1. Bolje prosjek PSNR vrijednosti kod ovog stupnja kompresije također ima WebP 1.

### *JPEG 50 i WEBP 50*

Objе kompresije imaju približno iste vrijednosti SSIM-a i PSNR-a, što nam govori da je kvaliteta slika nakon kompresije približno jednaka. Međutim, WebP 50 dominira kada promatramo vrijednosti CR-a. To nam govori da slike komprimirane WebP algoritmom zauzimaju puno manje memorijskog prostora.

Vizualno promatrajući slike JPEG ima oštrije prelaze rubnih vrijednosti promatrane slike, tj susjedni slikovni elementi se mogu znatno razlikovati dok kod WebP slike prijelazi su blaži tj. susjedni slikovni elementi imaju slične vrijednosti.

### *JPEG 100 i WEBP 100*

SSIM i PSNR imaju približno jednake prosječne vrijednosti. Slike nakon kompresije su približno jednake kvalitete. Prosječna vrijednost CR-a je znatno bolja kod WebP kompresije, te stoga slike komprimirane WebP metodom zauzimaju manje memorijskog prostora.

## 10. Zaključak

U radu su detaljno analizirane performanse JPEG i WEBP kompresija na slikama prirode (rezultati mogu biti drugačiji na drugim tipovima slika npr. slike lica). Iz rezultata se da zaključiti da niti WebP niti JPEG ne dominira u svim obavljenim analizama. WebP je znatno bolji ako nam kvaliteta slike nije toliko bitna, te zbog toga možemo koristiti veći stupanj kompresije (WEBP1 i WEBP50). Ukoliko nam je bitnija kvaliteta, JPEG postaje bolji odabir.

Promatrajući rezultate JPEG 1 i WebP1 kompresije vidljivo je da WebP 1 kompresija daje bolje rezultate. Zauzima manje memorijskog prostora, te daje bolje rezultate SSIM i PSNR mjerenja, a uz to koristi manje vrijednost bbp-a kako bi nam dala informacije o slikovnom elementu. Navedeni zaključak nam govori da ukoliko želimo sliku komprimirati većim stupnjem kompresije bolje rezultate ćemo postići ako koristimo WebP kompresiju.

Smanjenjem stupnja kompresije JPEG i WebP nam daju približno jednaku kvalitetu slike ukoliko promatramo vrijednosti SSIM i PSNR-a. Ako vizualno promatramo sliku komprimiranu istim stupnjem kompresije u oba algoritma, možemo zaključiti da nam JPEG daje oštriju slike tj. da su prijelazi na rubnim granicama promatranog objekta vizualno precizniji, dok rubne granice promatranog objekta kod WebP-a imaju manje oštar prijelaz tj. susjedni slikovni elementi vizualno su slični.

Ako slike komprimiramo kompresijom bez velikih gubitaka (npr. JPEG100 ili WebP100) može se zaključiti da WebP u takvim slučajevima kompresije zauzima više memorijskog prostora, a vrijednosti SSIM i PSNR-a su gotovo jednake JPEG-u. Gore navedeni zaključak nam govori da ukoliko želimo koristiti kompresije bez gubitaka JPEG nam je bolji odabir jer daje približno jednako kvalitetnu sliku, a pritom zauzima znatno manje memorijskog prostora.

Ovaj rad potvrđuje Google-ovu hipotezu da bi se korištenjem WebP formata na internetu znatno ubrzao prijenos slika, te smanjilo iskorištenje memorijskog prostora ako bi se WebP format bio više u uporabi koristiti tj. preuzeo dominaciju na internetu.



Odluka o odabiru između formata WebP i JPEG neće nam uvijek biti ista. Ona će ovisiti o tome koliko nam je važna kvaliteta, koji ćemo stupanj kompresije koristiti, te koliko nam je važna veličina memorijskog prostora koju slika zauzima.

## Popis tablica

Tablica 1 - Run-length kodiranje .....	17
Tablica 2 - Huffmanovo kodiranje .....	18
Tablica 3 - Kodiranje kodnom tablicom .....	19
Tablica 4 - Rezultati JPEG kompresije .....	38
Tablica 5 - Rezultati WebP kompresije.....	39
Tablica 6 - Stupanj kompresije za pojedinu sliku .....	41
Tablica 7 – bbp indexi.....	42

## Popis slika

Slika 1 - Barnsleyeva paprat .....	14
Slika 2- promjena intenziteta unutar bloka.....	22
Slika 3 - YCbCr prostor boja.....	23
Slika 4 - Usporedba RGB i YCbCr slike po kanalima.....	23
Slika 5 - 8x8 blok.....	24
Slika 6 - Zig-zag skeniranje .....	26
Slika 7 - Slika u izvornom BMP formatu .....	38
Slika 8 - Usporedba WebP 1 i JPEG 1.....	40
Slika 9 - Usporedba WebP 50 i JPEG 50 .....	40
Slika 10 - Usporedba WebP 100 i JPEG 100 .....	40

## Literatura

- [1] Google, "webP: A new image format for the Web," 2014. [Online]. Available: <https://developers.google.com/speed/webp/>.
- [2] D. Salomon, G. Motta, and D. Bryant, *Handbook of data compression*. New York, 2010.
- [3] B. J. L. Ronald A. DeVeore, Bjorn Jawerth, "Image Compression Through Wavelet Transform Coding," 1992, p. 38.
- [4] D. Saupe, R. Hamzaoui, and H. Hartenstein, *Fractal image compression: an introductory overview*. 1997.
- [5] A. D. Barnsley, M. F., Sloan, "Methods and apparatus for image compression by iterated function system," 1990.
- [6] D. Bojanjac, "Odabir parametara kompresije videosignala za televiziju visoke kvalitete," *Zhurnal Eksp. i Teor. Fiz.*, 2009.
- [7] M. Žagar, "Metode kompresije 4D podataka u oblikovanju virtualne stvarnosti Martin Žagar, dipl. ing.," pp. 1–7.
- [8] S. Grgić, "Kodiranje područja od interesa." [Online]. Available: <http://www.vcl.fer.hr/dtv/jpeg/roi.htm>.
- [9] D. Novosel, "Izvedba i analiza digitalnih nepokretnih slika prema normi JPEG2000," 2005.
- [10] A. Poljićak, *Zaštita vlasništva reproducirane slike umetanjem digitalnog vodenog žiga*. Zagreb, Doktorski rad, 2011.
- [11] G. K. Wallace, "The JPEG Still Picture Compression Standard 2 Background : Requirements and Selec-," pp. 1–17, 1991.
- [12] R. Eder, *Moderní kompresní formáty digitálního obrazu*. Brno, 2014.
- [13] D. Matkovi, "Mjerenja kvalitete slike u multimedijским aplikacijama," pp. 25–31.
- [14] J. Miano, *Compressed Image File Formats*. 1999.
- [15] "google.com." [Online]. Available: <https://developers.google.com/speed/webp/>.