Utjecaj normalizacije slike na perceptualni hash

Ferenčak, Matej

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Matej Ferenčak

UTJECAJ NORMALIZACIJE SLIKE NA PERCEPTUALNI HASH

DIPLOMSKI RAD

Varaždin, 2022.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Matej Ferenčak

Matični broj: 48881

Studij: Baze podataka i baze znanja

UTJECAJ NORMALIZACIJE SLIKE NA PERCEPTUALNI HASH

DIPLOMSKI RAD

Mentor/Mentorica:

Doc. dr. sc. Petra Grd

Varaždin, srpanj 2022.

Matej Ferenčak

Izjava o izvornosti

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

_____

Sažetak

Tema diplomskog rada je „Utjecaj normalizacije slike na perceptualni hash“. Početak razrade započinje opisivanjem toga što je to hash i što su to hash funkcije te njihova svojstva. Spominju se i hash tablice kako bi razumijevanje samog perceptualnog hasha bilo lakše. Zatim se spominje i opisuje perceptualni hash, glavni principi prema kojima on radi i principi na kojima rade perceptualne hash funkcije. Nakon perceptualnog hasha spominje se kriptografski hash, kojeg je bitno spomenuti zbog toga što se često uspoređuje s perceptualnim hashom, a nakon njegovog opisivanja dolazi do razlika između ta dva hasha. Nakon što je opisan i definiran perceptualni hash spominje se što je to normalizacija slike. Opisano je što je to normalizacija i način na koji se ona može provesti. Na samom kraju razrade nalazi se praktični dio u kojim se uspoređuju četiri algoritma za računanje perceptualnog hasha na različitim opcijama normalizacije. Praktičnim dijelom pokazane su razlike između pojedinog algoritma i prikazani su konačni rezultati prema kojima se može zaključiti koji je algoritam najbolji, a koji najlošiji.

Ključne riječi: hash; usporedba; normalizacija; perceptualni hash; A-Hash; P-Hash; D-Hash; W-Hash

Sadržaj

1. Uvod

Tema diplomskog rada je „Utjecaj normalizacije slike na perceptualni hash“. Ovu temu odabrao sam zbog toga što me hash još od prije jako zanimao pa mi je ovo bila prilika naučiti i upoznati što je to perceptualni hash. Tema je jako zanimljiva zato što obrađuje relativno nove tehnologije koje tek počinju imati svoju ulogu u stvarnom svijetu, govori se o periodu unazad dvije godine. Na samom početku razrade opisuje se osnovna hash funkcija, njezina uloga i njezina svojstva, spomenuta je i hash tablica kako bi se lakše povezali pojmovi vezani uz hash funkciju. Dalje se spominje i jedan od najbitnijih pojmova, a to je perceptualni hash, opisani su principi i načela na kojima perceptualni hash radi i opisane su perceptualne hash funkcije. Često se perceptualni hash uspoređuje s kriptografskim hashom, ali potrebno je navesti razlike između njih te je tako i napravljeno na kraju poglavlja o hash funkcijama. U samoj sredini razrade spominju se perceptualni hash algoritmi koji se kasnije koriste i u praktičnom dijelu diplomskog rada. Nakon objašnjenih i definiranih perceptualnih algoritama, spominje se i drugi najbitniji pojam u diplomskom radu, a to je normalizacija slike. Opisan je način na koji normalizacija radi i naveden je alat koji se koristi za provedbu normalizacije kako bi se mogao provesti praktični dio diplomskog rada. Na samom kraju razrade nalazi se praktični dio u kojem su provedene različite usporedbe, kako bi se mogao zaključiti utjecaj normalizacije na perceptualni hash.

Motivacija za odabir i izradu ove teme diplomskog rada je ta što me prije jako zanimao hash pa sam odlučio uzeti ovu temu kako bih naučio što je to perceptualni hash. Kako se općenito hash i pojmovi vezani uz hash ne spominju često, tu se i rodio interest za odabirom ove teme diplomskog rada.

Značajnost teme ovog diplomskog rada je prikazati postoji li kakav utjecaj normalizacije slike na promjenu perceptualnog hasha.

2. Metode i tehnike rada

U ovom diplomskom radu razrađene su dvije cijeline, teorijska i praktična. Za izradu ovih dviju cjelina korištena je desktriptivna metoda.

Teorijski dio temeljen je na činjenicama te je za izradu teorijskog dijela potrebno istraživanje stručne i znanstvene literature. Analizom i stručne i znanstvene literature napravljen je teorijski dio ovog diplomskog rada.

Opcije prema kojima su se uspoređivali algoritmi temeljene su na već postojećim opcijama koje nudi alat Adobe Photoshop Express. Ukupno se uspoređivala normalizacija za deset opcija. Opcije koje su se koristile za normalizaciju su redom sljedeće: Charm1, Faded B&W, Classic, Aqua, Blue Tint, Exposure, Vibrance, Fade, Sharpen i Blur. Algoritmi koji su se uspoređivali imaju istu namjenu, a to je računanje perceptualnog hasha i redom su A-Hash, P-Hash, D-Hash i W-Hash.

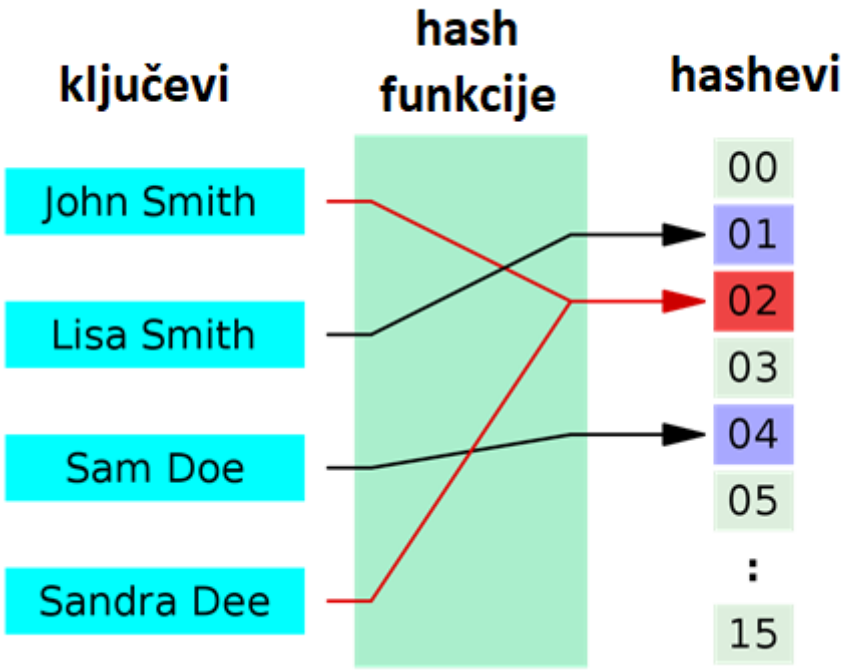
Alati koji su korišteni pri izradi ovog diplomskog rada su sljedeći, za provedbu normalizacije korišten je besplatan alat Adobe Photoshop Express i korišten je isto tako besplatan alat Visual Studio Code s ekstenzijom za Python kako bi se mogla izračunati hash vrijednost slika.

3. Općenito o hash funkcijama

Ovo poglavlje daje detaljniji uvid u to što je to zapravo hash funkcija. Kratko je napravljen osvrt na to što su to hash tablice, radi lakšeg shvaćanja i daljnjeg definiranja poglavlja. Kako se u naslovu teme diplomskog rada spominje hash, bitno je definirati što je to hash funkcija. Osim detaljnog definiranja hash funkcije, u ovom poglavlju se definira i što je to perceptualni hash te isto tako što je to kriptografski hash. Na kraju samog poglavlja, prikazane su razlike između perceptualnog i kriptografskog hasha na temelju navedenih prednosti i nedostataka pojedine vrste hash-a.

Kada se govori o hash funkcijama misli se na funkcije koje se koriste za mapiranje podataka određenih veličina u fiksirane vrijednosti. Vrijednosti odnosno podaci koje hash funkcija vraća, nazivaju se hash vrijednosti, ali u praksi se češće koristi naziv hashevi [1]. Hashevi se koriste za indeksiranje fiksnih vrijednosti unutar tablice koja se naziva hash tablica [4]. Proces u kojem hash funkcija indeksira hash tablicu, naziva se hashiranje ili stručnijeg naziva, adresiranje raspršene pohrane [30]. Najčešća primjena hash funkcija i njihovih hash tablica je u aplikacijama za pohranu podataka i aplikacijama za dohvaćanje podataka gdje je potreban brz odgovor od strane aplikacije. Mjesto koje im je potrebno za pohranu nešto je malo veće od samih podataka. Hashiranje je prostorno učinkovito pristupanje podacima kojim se želi izbjeći ne konzistentno vrijeme pristupa uređenim i ne uređenim listama te isto tako vrijeme pristupa strukturiranim stablima [1]. Osim u listama i strukturiranim stablima, proces hashiranja najčešće se koristi za implementaciju hash tablica [30]. Hash funkcije same po sebi imaju veliku ulogu u pohrani podataka, ali često se povezuju s kontrolnim znamenkama, otiscima prstiju, funkcijama randomizacije, kodovima za ispravljanje pogrešaka i šiframa. Koncepti i načini na koji su prethodno navedeni pojmovi dizajnirani, veoma su slični, ali sama uloga i optimizacija se razlikuju [1].

Sljedeća slika najbolje prikazuje ulogu hash funkcije u procesu hashiranja. Hash funkcija prima ključ koji se dalje prosljeđuje u hasheve u hash tablici. Hashevi na slici mogu se tumačiti i na način da su to memorijske lokacije u koje se pohranjuju zapisi [30]. Ovaj primjer slike ujedno prikazuje i jedan od slučajeva koji su mogući, a to je kolizija između ključeva. Kolizija ključeva detaljnije je objašnjena u nastavku poglavlja, točnije u podnaslovu 3.1. Hash tablica.



Slika 1: Grafički prikaz hash funkcije (Preuzeto sa: [30])

Način na koji hash funkcija radi je veoma specifičan, ali za daljnje razumijevanje perceptualnog hasha bitno je napomenuti, hash funkcija na svoj glavni ulaz odnosno svoj ulazni parametar gleda kao na ključ koji je povezan s nekim specifičnim zapisom (hash vrijednost) [1]. Ključ, odnosno ulaz, može biti fiksirane dužine te može biti brojčani (eng. integer), varijabilni (eng. variable) ili može biti čak i neko ime te osim navedenih vrsta, glavna osobina ključa jest ta da je to atribut koji olakšava identicikaciju pojedinog zapisa u hash tablici [30]. Nakon što hash funkcija uspješno prihvati svoj ulazni parametar, kao izlaz daje hash kod koji služi za indeksiranje hash tablice koja je zadužena za dugotrajnu ili kratkotrajnu pohranu podataka ili pokazivača na te podatke.

Glavne karakteristike hash funkcije su brzina i smanjenje dupliciranih vrijednosti. Kako bi hash funkcije osigurale brzinu i efikasnost koriste takozvane izračune distribuirane vjerojatnosti smanjujući vrijeme pristupa za najbližu konstantu. Prilikom toga bitno je napomenuti da i dizajn može utjecati na brzinu izvođenja hash funkcije. Hash funkcija se može dizajnirati tako da vrati najbolji mogući ishod u worst-case scenariju te je tu posebnost i značajnost korištenja hash funkcija, relativno brzo izvođenje u slučaju kada postoji puno podataka kojima je potreban pristup [1].

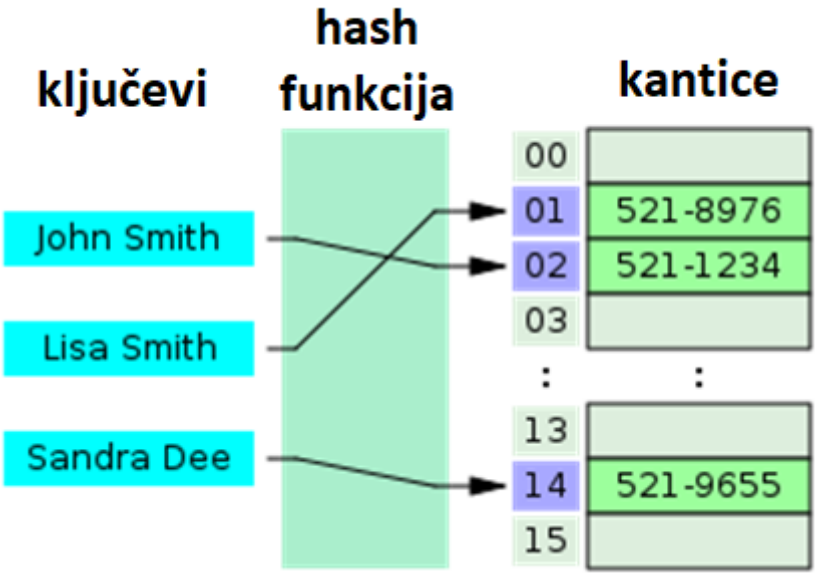
3.1. Hash tablica

U ovom dijelu obrađuje se pojam hash tablica, kroz jednostavni primjer biti će objašnjen pojam hash tablice i njezina najčešća upotreba u praksi.

Pojam hash tablica asocira na to da postoji neka tablica koja služi kao pohrana u koju se uz pomoć hash funkcije spremaju određene vrijednosti. Stučna definicija kaže da je hash tablica ili hash mapa podatkovna struktura koja koristi hash funkciju za preslikavanje ključeva u njima dodjeljene vrijednosti [2]. Kao što je na početku ovog poglavlja navedeno hash funkcija služi za pretvaranje ključa u indeks te uz pomoć indeksa traži odgovarajuću vrijednost u nizu elemenata [2]. Taj niz elemenata naziva se hash tablica. Prethodno se stalno spominjao pojam kolizija, to je pojam koji se javlja prilikom dodjele ključa nekom indeksu, gdje dva ključa imaju jednak indeks što ujedno znači da imaju i istu vrijednostu u hash tablici [3].

Hash tablice najčešće za mjeru učinkovitosti koriste izraz prosječna cijena svakog pronalaženja. Prosječna cijena podrazumijeva broj računalnih naredbi koje su potrebne za spajanje ključa s njegovim indeksom u hash tablici [3]. U većini slučajeva prosječna cijena ne ovisi o broju elemenata odnosno podataka koji su pohranjeni u hash tablici. Postoje različite implementacije hash tablica koje omogućuju neograničen broj unošenja ulaza, brisanje onih ključeva i vrijednosti koje se nalaze u tablici te sve to uz konstantnu cijenu tako da se ne prelazi prosječna cijena po naradbama [2].

Sljedeći primjer je najjednostavniji primjer grafičkog prikaza hash tablice. Primjer je strukturno veoma sličan, ako ne i isti kao i za hash funkciju. Prema samim primjerima vidljivo je da su hash tablica i hash funkcija ovisne jedna o drugoj. Ovaj primjer hash tablice dosta jasnije oslikava kako je zapravo jednostavan proces hashiranja, a opet tako bitan. Na slici je vidljivo da se ključevi odnosno ulazi nalaze na lijevoj strani, u sredini se nalazi hash funkcija i desno se nalazi hash tablica.



Slika 2: Grafički prikaz hash tablice (Preuzeto sa: [31])

Kao i kod hash funkcije glavna značajka hash tablice je brzina. Hash tablice su znatno brže u odnosu na druge tablične strukture koje se koriste u računalnom sustavu, osobito kada se radi o velikom broju podataka koji se nalazi u tablici. Razlog tomu je taj što su hash tablice jako učinkovite kada se količina podataka može unaprijed predvidjeti što bi značilo da se struktura tablice može unaprijed alocirati i biti optimalna bez da se kasnije mijenja njezina sturktura.

3.1.1. Upotreba hash tablica

U ovom dijelu spominju se najosnovniji oblici struktura u kojima se koristi hash tablica. To su najčešće strukture koje su potrebne u različitim dijelovima i područjima računalnog sustava.

Hash tablica koristi se u raznim podatkovnim strukturama koje najčešće čine bitne dijelove računalnog sustava. Pojavljuje se u sljedećim područjima:

- Asocijativne tablice
- Indeksi u bazama podataka
- Skupovi [2].

Asocijativne tablice su tablice koje u sebi sadrže asocijativne nizove. Asocijativni nizovi su nizovi čiji su indeksi neki proizvolji nizovi ili neke druge kompleksnije strukture podataka. Hash tablice se najčešće upotrebljavaju za njihovu implementaciju. Ovakve

strukture za pohranu podataka kiriste se u interpretiranim porgramskim jezicima kao što su Gawk i Perl [2].

Kod **indeksa u bazama podataka** hash tablice su u rangu s balansiranim stablima (eng. B - Tree). Hash tablice koriste se za implementaciju indeksa u baze podataka. Indeksi u bazama podataka koriste se kako bi pripomogli povećanju brzine rada baze podataka.

Skupovi u hash tablicama mogu biti zadani na način da umjesto vraćanja vrijednosti za zadani ključ, vraćaju odgovor na to postoji li uopće takav unos u skupu poznatih ulaza odnosno ključeva [2]. Tako se hash tablicama može implementirati skup koji može pomoći u traženju zadanog ključa. Skupovi se iz toga razloga dijele na statičke skupove i dinamičke skupove. Statički skupovi su već unaprijed poznati ključevi, dok su dinamički skupovi suprotnost i mogu biti bilo što te nisu poznati unaprijed.

3.2. Perceptualni hash

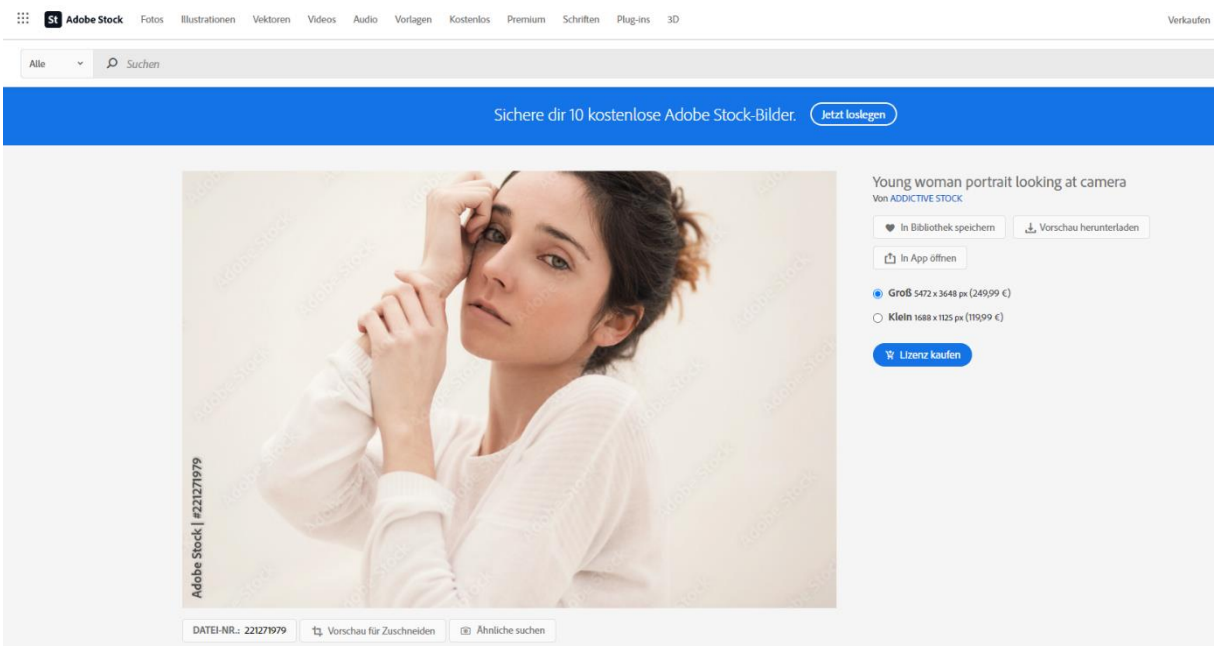
U ovom podnaslovu definiran je perceptualni hash, njegove karakteristike odnosno značajke, način na koji perceptualni hash radi, isto tako i njegove prednosti i nedostaci.

Godine 2010. prvi put ozbiljnije se počinje spominjati perceptualni hash od strane Christoph Zaunera koji je napisao *„Implementation and Benchmarking of Perceptual Image Hash Functions“* [6]. Kasnije, godine 2017., istraživanja su otkrila kako se Googlovo pretraživanje slika temelji na perceptualnom hashu [8]. Perceptualni hash sve više je prisutan u današnjoj tehnologiji kako bi se što više smanjio broj krivotvorenih multimedijskih sadržaja i kako bi se spriječila kršenja i povreda autorskih prava.

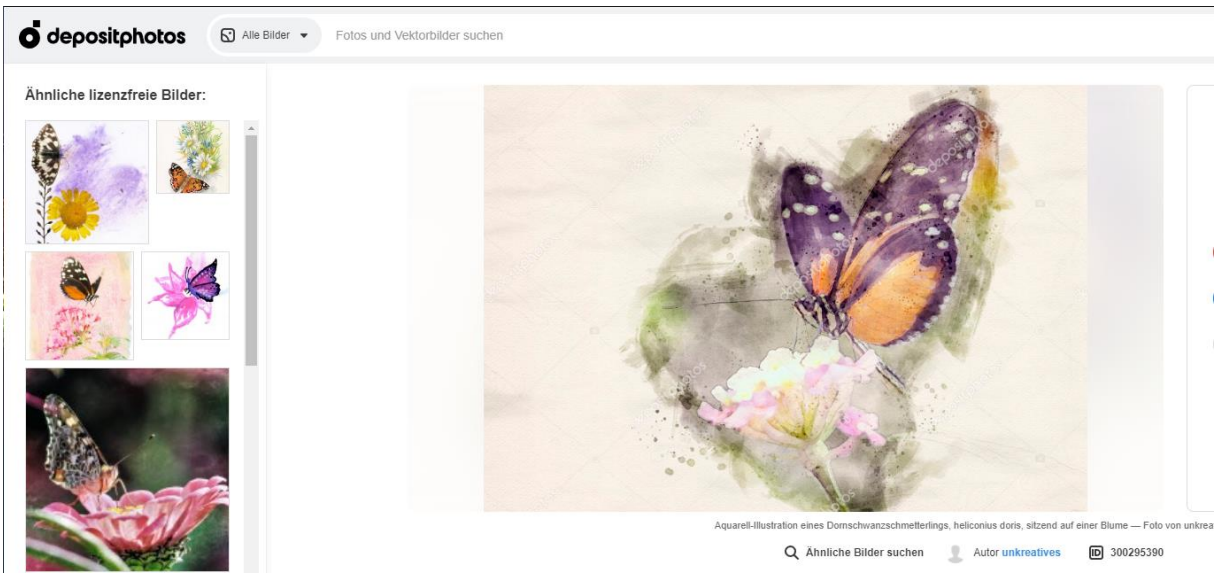
Perceptualni hash koristi algoritam za „otisak prsta“ kako bi proizveo isječak kroz različite oblike multimedije [9]. Drugim riječima to je otisak prsta multimedijske datoteke koji je izveden iz različitih svojstava i značajki njezinog sadržaja. Pod otiskom prsta smatra se digitalni zapis koji je kao i kod ljudi jedinstven za svaku osobu. U daljnjem dijelu ovog podnaslova detaljnije je definiran digitalni otisak prsta. Perceptualni hash sam po sebi je veoma osjetljiv na lokalitet, što znači da je analogan ako su značajke multimedije slične ili iste [6]. Kako hash ima hash funkcije tako postoje i perceptualne hash funkcije koje se koriste u digitalnoj forenzici. Najvažnija upotreba perceptualnog hasha je u pronalaženju slučajeva u kojima se krše

autorska prava pojedine multimedije, to se radi na način da se uspoređuje perceptualni hash multimedije na koju se sumnja da je krivotvorena s perceptualnim hashom orginalne multimedije kako bi se našla razlika u vodenom žigu odnosno digitalnom potpisu [7]. Vodeni žig je informacija u koju su pohranjeni podaci o autoru multimedije [10].

Tehnologija perceptualnog hasha sve više se pojavljuje u novim sustavima. Postoje razna istraživanja koja se onose na temu perceptualnog hasha. Istraživanje koje je izašlo 2021. objavljeno od strane Apple Inc. govori o tome kako Apple razvija bazu podataka koja služi za spriječavanje seksualnog zlostavljanja djece (eng. Child Sexual Abuse Material) [11]. Riječ je o sustavu koji je poznat pod nazivom NeuralHash. Oni tim sustavom umjesto tradicionalnog skeniranja fotografija na iCloud poslužiteljima, žele sustav uparivanja poznatih hasheva iz baze podataka koju osigurava Nacionalni centar za nestalu i iskorištavanu djecu, ali sve to da se odvija pozadinski na korisnikovu uređaju [11]. Ideja je da se ta baza podataka pretvori u nečitljiv skup hash vrijednosti koji je pohranjen na uređajima korisnika. Ovo istraživanje je podiglo veliku prašinu između Apple-ovih korisnika, ali i stručnjaka. Oliver Kuederle napisao je zanimljiv članak naslova „The Problem With Perceptual Hashes“ u kojem ukazuje na moguće probleme u takvim sustavima kao Apple-ov NeuralHash [12]. Kolizije koje se spominju u podnaslovu 3.2. Hash tablica, ovdje predstavljaju veliki problem. Rečeno je kako ne postoje idealne hash funkcije, pa tako i perceptualne hash funkcije, koje neće vratiti niti jednu koliziju prilikom hashiranja. Kuederle je u svom članku opisao jedan slučaj u kojemu je algoritam sličan NeuralHash-u vratio jednaku hash vrijednost za fotografski portet žene, koji je dostupan na komercijalnoj bazi podataka zvanom Adobe Stock pod oznakom 221271979 (slika 3.), s fotografijom djela apstraktne umjetnosti, koja se isto nalazi u komercijalnoj bazi podataka pod nazivom „deposit photos“ (slika 4.) [12]. Apple je naveo kako će se takvi slučajevi kolizija ručno pregledavati. Kuederle navodi kako će prilikom korištenja takvih algoritama za otkrivanje kriminalnih radnji, stradati i nevini korisnici te da će njihova privatnost biti poprilično narušena [12]. Postoji još jedno istraživanje koje se nadovezuje na Apple-ov NeuralHash, napravila ga je skupina istražitelja pod imenima Lukas Struppek, Dominik Hintersdorf, Daniel Neider i Kristian Kersting pod nazivom „Learning to Break Deep Perceptual Hashing: The Use Case NeuralHash“ u kojem istražuju ranjivosti NeuralHasha kao predstavnika algoritama dubokog perceptualnog hashiranja [13]. Ovo istraživanje pokazalo je doista bizarne činjenice, kao na primjer da se samo malim promjenama na slikama mogu dobiti kolizije između različitih slika [13]. Rezultati su pokazali kako je moguće iskoristiti algoritam za različite napade i zapravo kazneno goniti nevine korisnike. Korištenjem besplatnih alata za uređivanje slika potencijalni napadači mogu jednostavnim transformacijama slike nadmudriti sustav te na taj način izbjeći detekciju ilegalnog materijala.



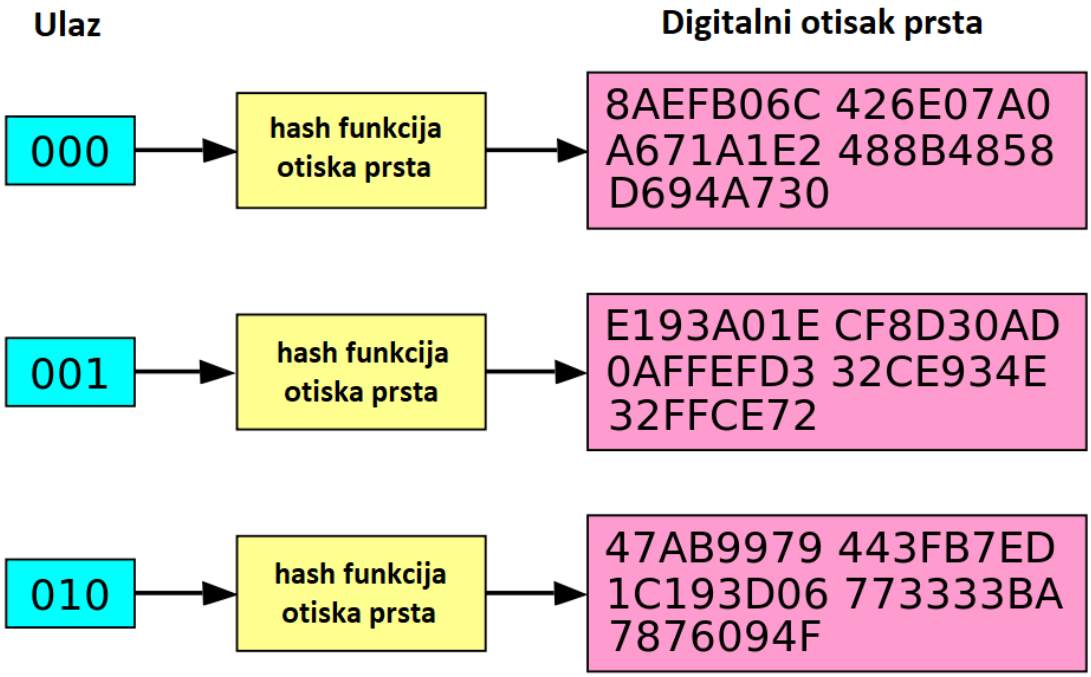
Slika 3: Fotografski portret 221271979



Slika 4: Fotografija djela apstraktne umjetnosti

3.2.1. Digitalni otisak prsta

U računalstvu digitalni otisak prsta je algoritam koji preslikava veliki podatak u puno kraći niz bitova, baš kao i kod ljudi, tako je i kod datoteka, svaka datoteka ima svoj otisak prsta [14]. Digitalni otisci prsta datoteke koriste se kako bi se izbjegle usporedbe i prijenos velikih količina podataka, pa tako web-preglednik ili proxy server može provjeriti je li datoteka izmjenjena samo na temelju usporedbe njezinog digitalnog otiska prsta s digitalnim otiskom prsta prethodne datoteke koja je dohvaćena [15]. Funkcije koje se koriste za izradu otiska prsta su visokoučinkovite hash funkcije koje se koriste za jedinstvenu identifikaciju blokova podataka [14].



Slika 5: Grafički prikaz nastanka digitalnog otiska prsta (Preuzeto sa: [32])

Kako perceptualni hash koristi algoritme otiska prsta, bitno je navesti koji su to algoritmi. Postoje dva različita algoritma koji se koriste za izradu otiska prsta, a to su Rabinov algoritam i kriptografske hash funkcije. Kriptografske hash funkcije definirane su u daljnjem dijelu ovog poglavlja.

Rabinov algoritam je prototip klase. Algoritam je brz te jednostavan za implementaciju, omogućuje spajanje, što je jedna od bitnijih karakteristika otiska prsta, te algoritam pruža matematički preciznu analizu vjerojatnost kolizija [14]. Nažalost to što je algoritam brz i jednostavan za implementaciju sa sobom nosi i posljedice koje se tiču sigurnosnog dijela

algoritma. Rabinov algoritam nije siguran protiv zlonamjernih napada. Napadač može jednostavno saznati ključ i koristiti ga kako bi promijenio datoteke bez da se otisak prsta promijeni [14]. Ovaj nedostatak pokrivaju kriptografske hash funkcije koje se spominju u daljnjem dijelu poglavlja.

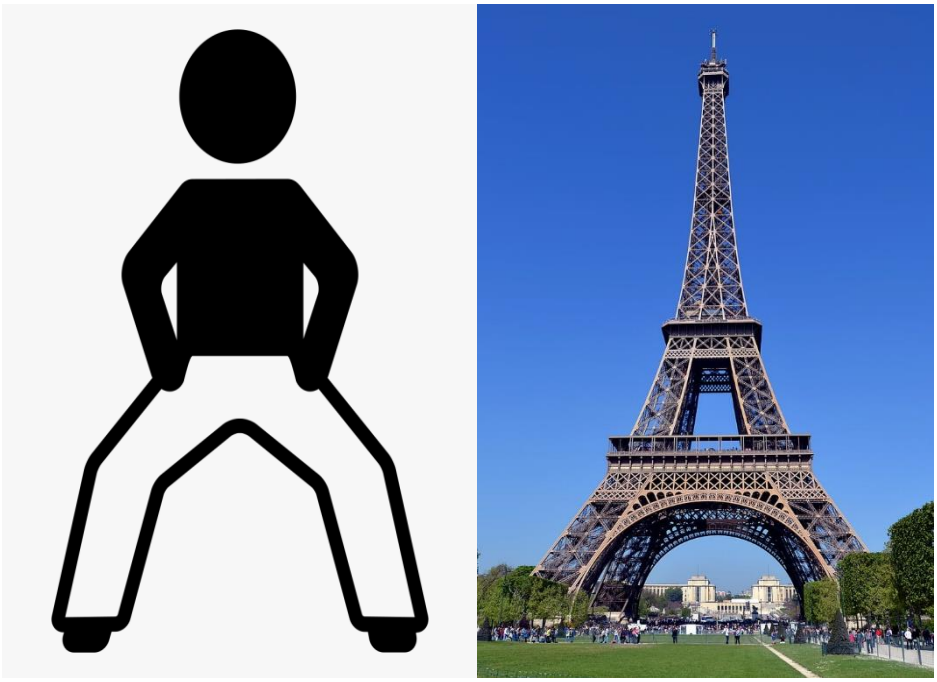
3.2.2. Perceptualne hash funkcije

Perceptualne hash funkcije same po sebi rade na način da uvijek daju isti ili sličan rezultat za slične slike ili zvukove [7]. Cilj je imati ljudsku percepciju kojom ljudi mogu unatoč različitim bojama ili frekvencijama, zaključiti kako se radi o istim ili sličnim slikama ili zvukovima [7]. U početku poglavlja spomenute su hash funkcije koje su osjetljive i na najmanje promjene, dok se kod perceptualnih hash funkcija dogodi i najmanja promjena koja može biti samo jedan bit, a prilikom takve promjene hash se može znazno izmjeniti, perceprualne hash funkcije vratiti će jednak izlaz kod promijenjenog i ne promijenjenog medija [7]. Perceptualne hash funkcije nisu toliko otporne na napade kao kriptografske hash funkcije. Dizajnirane su tako da slični mediji vrate jednak ili sličan izlaz, što ih čini ranjivima na lažiranje i na krivo usmjeravanje prilikom hashiranja.

3.2.2.1. Rad perceptualnih hash funkcija

Perceptualni hash je relativno novo područje tehnologije koje još u razvoju pa prema tome još uvijek ne postoje definirani standardi koji bi olakšali korištenje i implementaciju. Perceptualne hash funkcije rade na principu da medij podijele u relativno velike blokove te ih onda pretvaraju u slične oblike odnosno u istu vrijednost [7]. Za distribuciju vrijednosti u tim blokovima smatra se da ima vrlo nisku razlučivost te su uzorci često isti ili vrlo slični.

Perceptualne hash funkcije kod slika koriste prethodno navedene tehnike, ali s bojama i blokovima odnosno pikselima [7]. Često je slučaj da perceptualne hash funkcije koriste za usporedbu između slika njihov oblik, primjer tomu može biti slika osobe koja ima raširene noge i slika Eiffelovog tornja, obje slike su istog oblika [7].



Slika 6: Jednaki oblici (Preuzeto sa: [33] i [34])

U nastavku slijede neka područja u kojima je perceptualni hash zastupljen:

- Kršenje autorskih prava (eng. copyright infringement)
- Označavanje videozapisa (eng. video tagging)
- Pogrešno pisanje (eng. misspelling)
- Sigurnost i sukladnost (eng. compliance) [7].

Kod kršenja autorskih prava hash vrijednosti mogu otkriti i uskladiti slike, čak i ako je promjenjena veličina originala, obrezivanjem ili smanjenjem [7].

Kod označavanja videozapisa perceptualni hash može indeksirati lice iz videozapisa kako bi se utvrdile određene osobe koje su vidljive na nekim snimkama [7].

Što se tiče pogrešnog pisanja, perceptualne hash funkcije mogu kategorizirati riječi prema njihovim zvukovima i tako ukazati na pogrešno napisane riječi [7].

Uz pomoć perceptualnih hash funkcija mogu se pronaći i identificirati ljudi na videozapisima ili fotografijama, a ujedno se mogu i utvrditi postoje li neki predmeti na ljudima koji bi ukazali na opasnost te se zato koriste kod sigurnosti i sukladnosti [7].

3.3. Kriptografski hash

U ovom podnaslovu obrađen je kriptografski hash, navedena su načela i principi na kojima je kriptografski hash dizajniran i na samom kraju navedeni su neki od najpoznatijih kriptografskih hash algoritama.

Kriptografska hash funkcija je matematički algoritam koji preslikava podatke u niz bitova fiksne veličine, odnosno hash vrijednosti [16]. Kriptografska hash funkcija je funkcija koja je zapravo jednosmjerna što znači da je nemoguće obrnuti izračun koji daje. Drugim riječima, jedini način na koji je moguće pronaći poruku na temelju hasha koji je ta poruka dala je taj da se napravi pretraživanje na silu (eng. brute-force search) mogućih ulaza i na taj način da se nađe mogući par hasheva koji odgovara ili korištenje dugine tablice (eng. rainbow table) koja u sebi ima već unaprijed izračunate izlaze kriptografskih hash funkcija te se najčešće i koristi za razbijanje lozinki koje se nalaze u hashu [16]. Iz tog razloga kriptografske hash funkcije su osnovna današnje kriptografije.

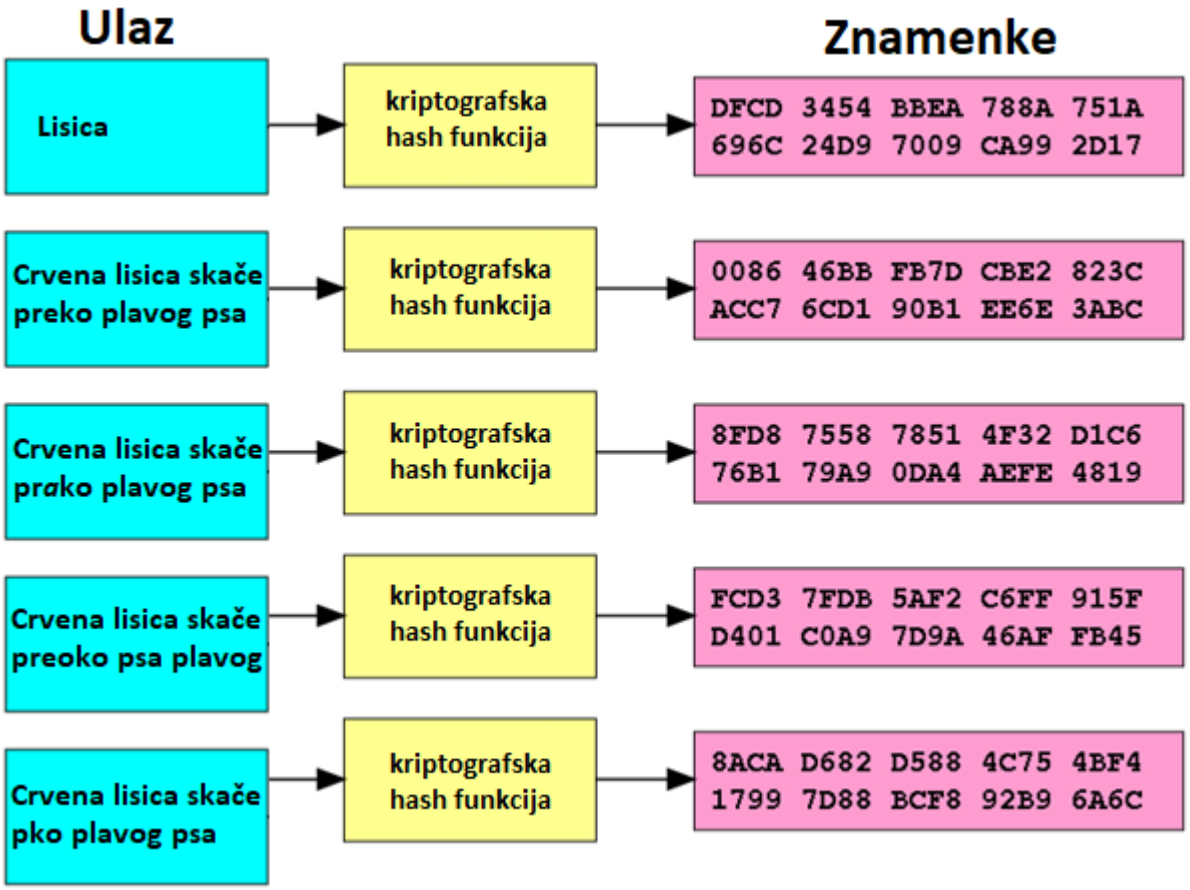
Kriptografska hash funkcija mora biti jednosmjerna, što znači da za ulaz (poruku) mora uvijek imati isti izlaz odnosno isti hash, ali uz to mora imati sljedeća svojstva:

- Brzo računanje hash vrijednosti za dane poruke
- Nemogućnost generiranja poruke za dani hash (jednosmjernost)
- Nemogući pronalazak dviju različitih poruka na temelju iste hash vrijednosti
- Male promjene u poruci moraju rezultirati potpuno različitim izlazom, tako da je u potpunosti nemoguće novu hash vrijednost usporediti sa starom vrijednosti (efekt lavine (eng. avalanche effect)) [17].

Kriptografske hash funkcije koriste se u različitim sigurnosnim aplikacijama kao što su digitalni potpisi, aplikacije za provjeru autentičnosti poruka (MAC) [18]. Pojavljuju se i u običnim hash funkcijama gdje se koriste za indeksiranje u hash tablice za potrebe kreiranja digitalnog otiska prsta. Kriptografske hash funkcije često se povezuju i čak nazivaju digitalnim otiscima prstiju, kontrolnim sumama ili hash vrijednostima, što zaista i je istina, ali ti pojmovi imaju u potpunosti različita svojstva i svrhe od kriptografske hash funkcije [18].

Sljedeća slika prikazuje grafički prikaz kako bi kriptografska hash funkcija trebala biti dizajnirana i na koji način bi trebala raditi. Na lijevoj strani se nalaze poruke koje su ulazi za kriptografsku hash funkciju, a na desnoj strani se nalaze izlazi koji kada se napravi i najmanja

promjena (zadnja četiri primjera) hash vrijednost se mijenja u potpunosti te je to takozvani efekt lavine.

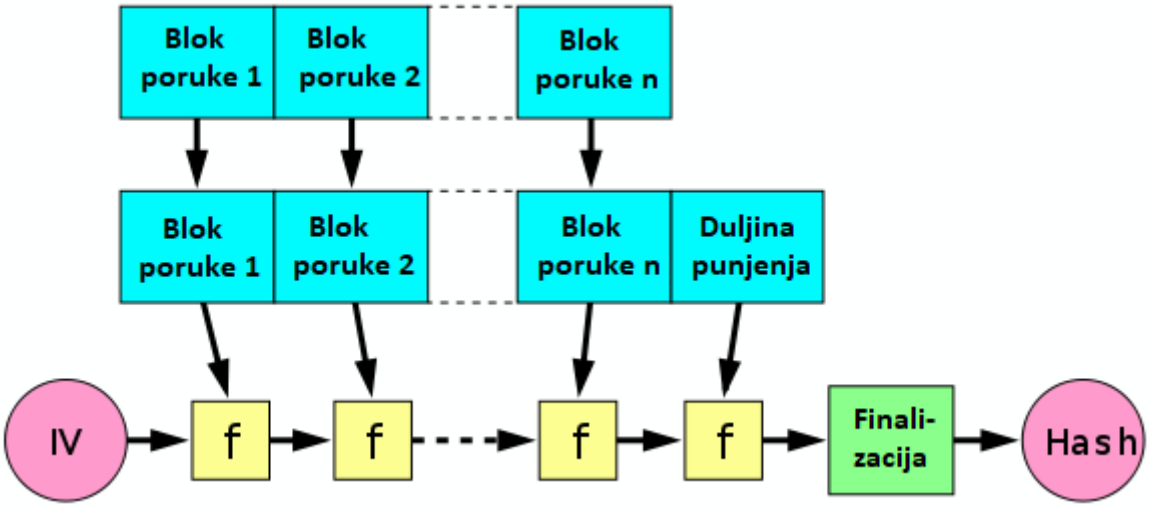


Slika 7: Grafički prikaz rada kriptografske hash funkcije (Preuzeto sa: [35])

3.3.1. Dizajn kriptografske hash funkcije

Najpoznatiji dizajn kriptografske hash funkcije je konstrukcija Markle – Damgård. Kriptografska hash funkcija mora biti spremna obraditi poruku koja može biti proizvoljne dužine u fiksnu vrijednost. Kako bi se to uopće moglo postići ulaz se mora rastaviti na niz blokova koji su jednake veličine te na taj način funkcije koje su zadužene za kompresiju mogu kompresirati poruku proizvoljne duljine [19]. Kriptografska hash funkcija koja se dizajnira prema konstrukciji Markle – Damgård otporna je na kolizije te kada dođe do toga, kroz funkciju se može doći do ulaza koji je uzrokovao koliziju [19]. Kod kompresije, zadnji blok koji se obrađuje, mora biti nedvosmisleno dopunjen, što daje dodatnu sigurnost kod ovakvog načina dizajna kriptografske hash funkcije [19].

Sljedeća slika grafički prikazuje konstrukciju Markle – Damgård. Plavom bojom označeni su blokovi poruke koji se kompresiraju u hash blokove, ovisno o kojem se kriptografskom hash algoritmu radi toliko je i blokova (bitova) koji se mogu napuniti.



Slika 8: Grafički prikaz kompresije kod konstrukcije Markle – Damgård (Preuzeto sa: [36])

3.3.2. Kriptografski hash algoritmi

U posljednje vrijeme postoji jako veliki broj kriptografskih hash algoritama, ali navedeni su oni najpoznatiji kriptografski hash algoritmi koji se najčešće koriste u praksi. Najčešći kriptografski hash algoritmi koji se koriste u praksi su:

- MD5
- SHA-1
- SHA-2
- SHA-3 [20].

MD5 kriptografski hash algoritam pojavio se 1991. godine kada ga je osmislio Ronald Rivest. Algoritam kolizije može izračunati unutar nekoliko sekudni, što ga čini sporim i neprikladnim za uporabu u većini slučajeva kada je potrebno koristiti kriptografski hash [20]. Algoritam je veličine 128 bitova.

SHA-1 razvijen je u sklopu Capstone projekta koji je pokrenula Američka vlada. SHA-1 kriptografski hash algoritam jako je loš po pitanju kolizija. Kolizije se mogu izazvati napadom zvanim SHAttered koji je oko sto tisuća puta brži nego brute-force napad [20]. Ovim napadom

SHA-1 kriptografski hash algoritam smatra se razbijenim, ali iako je razbijen još uvijek je u upotrebi. Veličina algoritma je 160 bitova.

SHA-2 razvijen je isto tako u sklopu Capstone projekta te je nadogradnja na SHA-1 algoritam. Ovaj algoritam je dizajniran Markle – Damgård konstrukcijom te je predstavljen 2001. godine [20]. SHA-2 kriptografski hash algoritam se zapravo sastoji od dva hash algoritma, a to su SHA-256 i SHA-512, što je ujedno i glavno svojstvo ovog algoritma [20]. Veličina SHA-256 kriptografskog hash algoritma je 32 bajta, dok je veličina SHA-512 kriptografskog hash algoritma 64 bajta.

SHA-3 kriptografski hash algoritam objavljen je 2015. godine. Glavno svojstvo mu je to što se bazira na Keccak algoritmu, taj algoritam je baziran na spužvastoj konstrukciji koja se također koristi za izradu kriptografskih hash funkcija [20]. SHA-3 kriptografski hash algoritam može imati četiri izlaza koji su sljedećih veličina 224, 256, 384 i 512 bita [20]. SHA-3 je jedan od najsigurnjih kriptografskih hash algoritama iz obitelji SHA (eng. Secure Hash Algorithm) [20].

3.4. Razlika između perceptualnog i kriptografskog hash-a

Iz samog naslova može se zaključiti kako se ovaj dio odnosi na osnovne razlike između preceptualnog hasha i kriptografskog hasha.

Obradom prethodnih naslova 3.3. Perceptualni hash i 3.4. Kriptografski hash može se zaključiti kako je glavna razlika između perceptualnog hasha i kriptografskog hasha način na koji se dobije izlazna vrijednost. Drugim riječima spomenuto je kako perceptualni hash može dvije slike svrstati u istu hash vrijednost, na temelju oblika koji se nalaze na slici, ali i male promjene na slici također ne utječu na izmjenu hash vrijednosti, dok kod kriptografskog hasha i najmanja promjena u poruci može rezultirati potpuno drugom hash vrijednosti za tu istu poruku koja je dobivena kao ulaz. Kako su oba hasha temeljena na hash funkciji svakako je bitno napomenuti kako je i sama svrha u kojoj su se pronašli različita. Kako je perceptualni hash zastupljen u pronalasku i otkrivanju kriminalnih radnji koje se tiču slika i krivotvorenja digitalnog otiska prsta, tako se kriptografski hash pronašao u kriptiranju poruka koje se šalju. Prednosti i nedostaci navedeni su u kontekstu kako su prethodno opisane svrhe pojedinog hasha.

U nastavku se nalaze tablice koje pobliže mogu predočiti prednosti i nedostatke za perceptualni hash i kriptografski hash.

Tablica 1: Prednosti i nedostaci perceptualnog hasha

Prednosti	Nedostaci
Male promjene ne mijenjaju hash vrijednosti	Za istu hash vrijednost može dati dvije slike koje su različitog konteksta te na taj način ugroziti nedužne korisnike
Jako dobar u radu sa slikama i videozapisima	Normalizacijom se hash vrijednost može u potpunosti promjeniti sve dok je ta promjena značajnija
Preuzeta ljudska percepcija tako da na temelju oblika ili frekvencija može zaključiti da su dvije slike ili zvukovi jednaki ili slični	Sigurnost je najveći problem zbog toga što se besplatnim alatima za uređivanje slika nemože izmjeniti hash vrijednost eksplicitnog i nedozvoljenog sadržaja

Tablica 2: Prednosti i nedostaci kriptografskog hasha

Prednosti	Nedostaci
Sigurnost je na prvom mjestu iz razloga što su funkcije temeljene na jednosmjernoj logici	Najmanja promjena u poruci rezultira u potpunosti novom hash vrijednosti
Otporan na različite napade te je za razbijanje kriptografskog hasha potrebno jako puno vremena	Jednom kada je kriptografski hash razbijen podaci koji se mogu izvući su jako osjetljivi i takva hash funkcija se više ne može koristiti, ako se želi zaštiti integritet podataka

4. Perceptualni hash algoritmi

Ovo poglavlje se odnosi na perceptualne hash algoritme koji se koriste u praktičnom dijelu diplomskog rada.

Perceptualni hash algoritmi koji se koriste u praktičnom dijelu, dio su Python biblioteke zvane ImageHash [27]. ImageHash biblioteka je jedna od najzastupljenijih biblioteka u kojoj se nalaze algoritmi bazirani na perceptualnom hashu. Biblioteka u sebi sadrži šest perceptualnih hash algoritama, a oni su sljedeći:

- Average hashiranje (A-Hash)
- Perceptual hashiranje (P-Hash)
- Difference hashiranje (D-Hash)
- Wavelet hashiranje (W-Hash) [27].

4.1. A-Hash

Prosječno hashiranje (eng. average) poznatiji kao A-Hash je najjednostavnija i najosnovnija metoda koja se koristi za generiranje perceptualnog hasha za dane slike [21]. Ovim algoritmom, hash vrijednosti se generiranju na temelju frekvencije slike, niske frekvencije predstavljaju strukturu slike, a visoke frekvencije predstavljaju detalje na slici [21]. A-Hash algoritmu je cilj pronalazak prosječne boje između svih vrijednosti iz matrice slike, na način da izračunava prosječnu vrijednost matrice slike [22]. Algoritam ima pet koraka koji opisuju njegovu implementaciju, 1. korak je mijenjanje veličine ulazne slike u 8 * 8 piksela, 2. korak je provedba konverzije prostora boja iz RGB-a u prostor boja sivijeg tona, 3. korak je izračunavanje srednje vrijednosti svih vrijednosti matrice osvjetljenja prethodne slike, 4. korak je provedba usporedbe svakog elementa iz matrice slike i izračunate srednje vrijednosti, te se na temelju toga dobiva nova binarna matrica od 64 elemenata i na posljetku 5. korak je kreiranje vektora iz matrice dobivene u prethodnom koraku, kako bi se dobio 64 bitni hash [22]. Dobivena hash vrijednost se kasnije može usporediti s drugim hashevima ostalih slika kako bi se odredila sličnost na temelju različitosti između dviju hash vrijednosti.

4.2. P-Hash

Perceptualni hash, poznatiji kao P-Hash je metoda koja se proširuje na A-Hash tako da koristi diskretne kosinusne transformacije (eng. discrete cosine transform) kako bi dobio informacije ljudskog vidnog sustava [21]. Jedina razlika je ta da kod izračunavanja hash vrijednosti ne koristi intenzitet slike kao A-Hash već koristi niske frekvencije dobivene nakon provođenja procesa diskretnih kosinusnih transformacija. P-Hash je implementiran kroz sedam koraka, 1. korak je mijenjanje slike u matricu veličine 32 * 32 piksela, 2. korak je pretvaranje skale boja u skalu sivijeg tona, 3. korak je izvođenje diskretne kosinusne transofrmacije nad matricom iz 1. koraka kako bi se dobila 32 * 32 matrica koeficijenta diskrente kosinusne transformacije, 4. korak je kreiranje vektora duljine 64 na temelju matrice koeficijenta iz prethodnog koraka, 5. korak je izračunavanje srednje vrijednosti na temelju koeficijenta, 6. korak je provođenje usporedbe 64 koeficijenta sa srednjom vrijednošću i na kraju u 7. koraku se dobiva 64 bitni binarni hash [22].

4.3. D-Hash

Hash razlike, poznatiji kao D-Hash je metoda koja se kao i prethodne dvije oslanja na A-Hash i stavlja strukturu slike u prvi plan. Algoritam smanjenjem veličine slike eliminira visoke frekvencije slike [22]. Značajka ovog algoritma je u tome da hash vrijednosti izračunava na temelju izmjena u gradijentu između susjednih piksela u matrici slike [21]. D-Hash algoritam ima četiri koraka implementacije, 1. korak je izmjena veličine slike na 9 * 8 piksela, 2. korak je pretvaranje skale boja u skalu sivijeg tona, 3. korak je provođenje usporedbe razlike između dva susjedna piksela kako bi se dobio ukupan broj od 8 razlika po redu i na kraju je 4. korak u kojem se izračunavaju 64 razlike za svaku sliku kako bi se dobila hash vrijednost od 64 bita [22].

4.4. W-Hash

Valni hash, poznatiji kao W-Hash je metoda koja za hashiranje koristi diskretnu valnu transformaciju za generiranje perceptualnih hasheva [22]. Glavna značajka ovog algoritma je ta da se analiza slika vrši u valovitoj domeni, uz spoznaju vremenske infomacije [23]. Ovakva transformacija se često koristi za uklanjanje redudancije u podacima s visoko koreliranim susjednim vrijednostima, a to su naravno pikseli slika [23]. W-Hash algoritam ima četiri koraka implementacije, 1. korak je nasumično izračunavanje valnih dekompozicija uz pomoć

Haarovog vala, 2. korak je kvantiziranje statičkog vektora korištenjem nasumičnog kvantizatora, 3. korak je vektor iz prethodnog koraka da se dekodira uz pomoć Reed-Mullerovog dekodera kako bi se ispravile greške prvog reda i kako bi se proizvela binarna hash vrijednost duljine n i na kraju 4. korak je provođenje druge faze dekodiranja nad hashom iz prethodnog koraka kako bi se u konačnici dobila kraća hash vrijednost [22].

5. Normalizacija slike

U ovom poglavlju definirano je što je to normalizacija slike te je na primjeru prikazana normalizacija slike koja se koristi nadalje i u praktičnom dijelu rada, ali prije prikaza načina provođenja normalizacije, opisat će se i alat koji se koristi za provedbu normalizacije nad slikama iz praktičnog dijela rada.

Pojam normalizacija slike sam po sebi daje odgovor, a to je prilagođavanje slike na način da joj se ne mijenja početni smisao, normalizira se [24]. Svrha normalizacije je pretvoriti ulaznu sliku na način da se promjena nalazi u rasponu vrijednosti piksela koji su poznati ljudskim osjetilima, kao što je i u prethodnoj rečenici navedeno, da slika ostaje normalna [24]. Normalizacija ostaje normalizacija sve dok se ne mijenja struktura i sadržaj slike, drugim riječima sve promjene koje se odnose na neko uklanjanje iz slike, dodavanje nekih oblika ili bilo kakvo takvo prilagođavanje slike, to se više ne svrstava u normalizaciju slike nego u manipulaciju slike. Normalizacija slike i manipulacije slike su zapravo dvije grane koje dolaze od pojma obrada slike.

Postoje formule prema kojima se normalizacija može napraviti ručno, a tada normalizacija može biti linerana i ne linarna [26]. Linarna normalizacija radi na temelju vrijednosti iz skale sivih tonova (eng. grayscale), dok ne linearna normalizacija radi na temelju Sigmundove funkcije [26]. Formula prema kojoj se izračunava linearna normalizacija je sljedeća:

$$I_N = (I - Min) * \frac{newMax - newMin}{Max - Min} + newMin\,,$$

gdje su $Min\, i\, Max$ vrijednosrti intenziteta orginalne slike, a $newMin\, i\, newMax$ vrijednosti intenziteta nove slike [25]. Formula prema kojoj se izračunava ne linearna normalizacija je sljedeća:

$$I_N = (newMax - newMin) * \frac{1}{1 + e^{-\frac{I-\beta}{\alpha}}},$$

gdje je α veličina raspona intenziteta, a β je intenzitet oko koje je raspon centraliziran [25]. Naravno da danas postoje programski alati koji u sebi imaju ugrađenu normalizaciju slike, te

se to naziva automatska normalizacija. U ovom radu za normalizaciju slike koristi se Adobe Photoshop Express o kojem se više govori u idućem podnaslovu.

5.1. Alat za provedbu normalizacije

Alat koji se koristi za provedbu normalizacije u ovom radu je Adobe Photoshop Express. Alat je besplatan za korištenje samo je potrebno napraviti registraciju i prijaviti se i alat se može koristiti [28]. Podrazumijeva se da alat ima mnogo manji broj opcija nego Premium verzija Adobe Photoshopa, ali opcije koje podržava Adobe Photoshop Express sasvim su dovoljne za provođenje normalizacije.

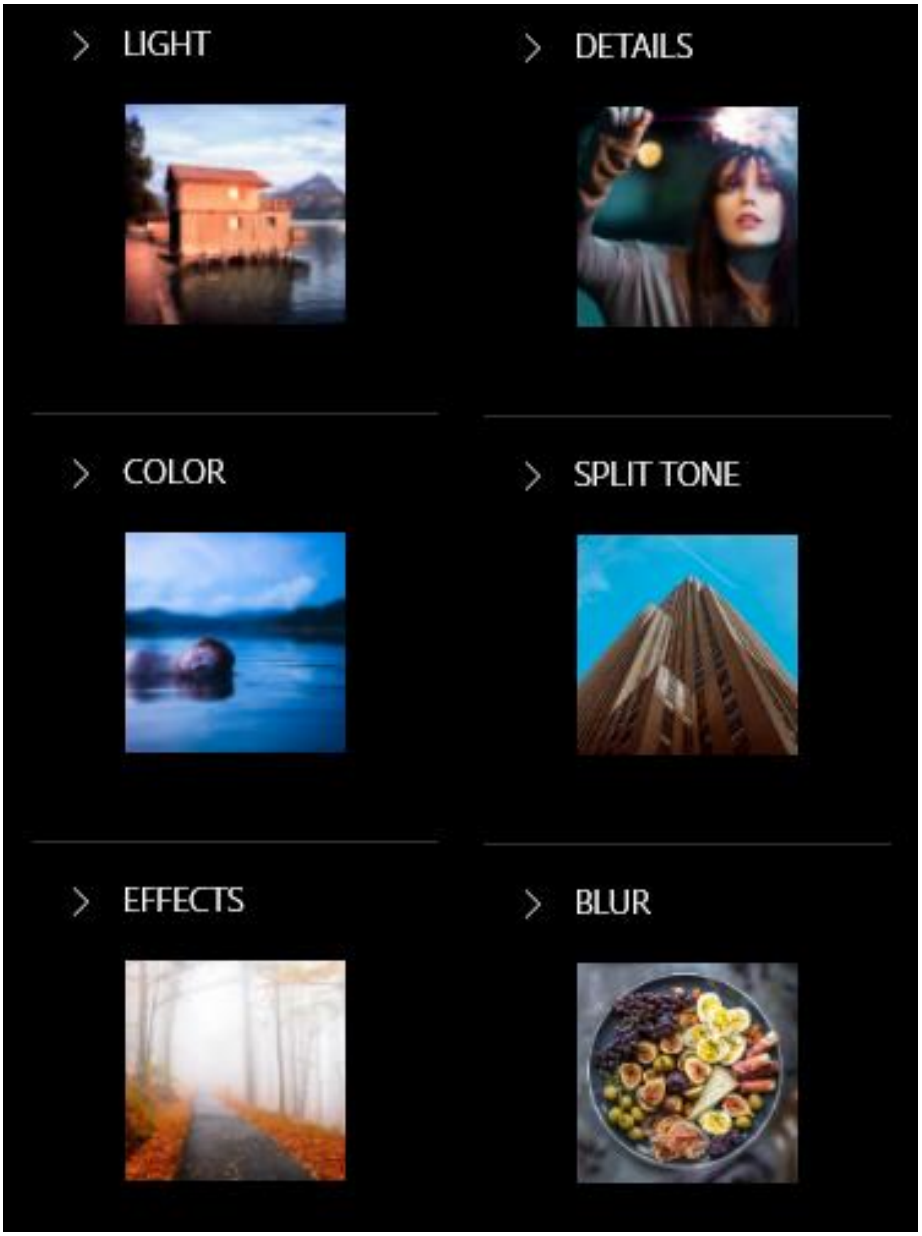
Slika 9: Adobe Photoshop Express logo (Preuzeto sa: [37])

Aspekti koji se uspoređuju u praktičnom dijelu su sljedeći:

- Izgledi
- Korekcije.

Svaki od ovih aspekata imaju svoje podopcije koje se najbolje mogu prikazati sa prikaza na sljedećim slikama. Prva slika predstavlja podopcije aspekta Izgledi, a druga slika podopcije aspekta Korekcije.

Slika 10: Podopcije aspekta Izgledi

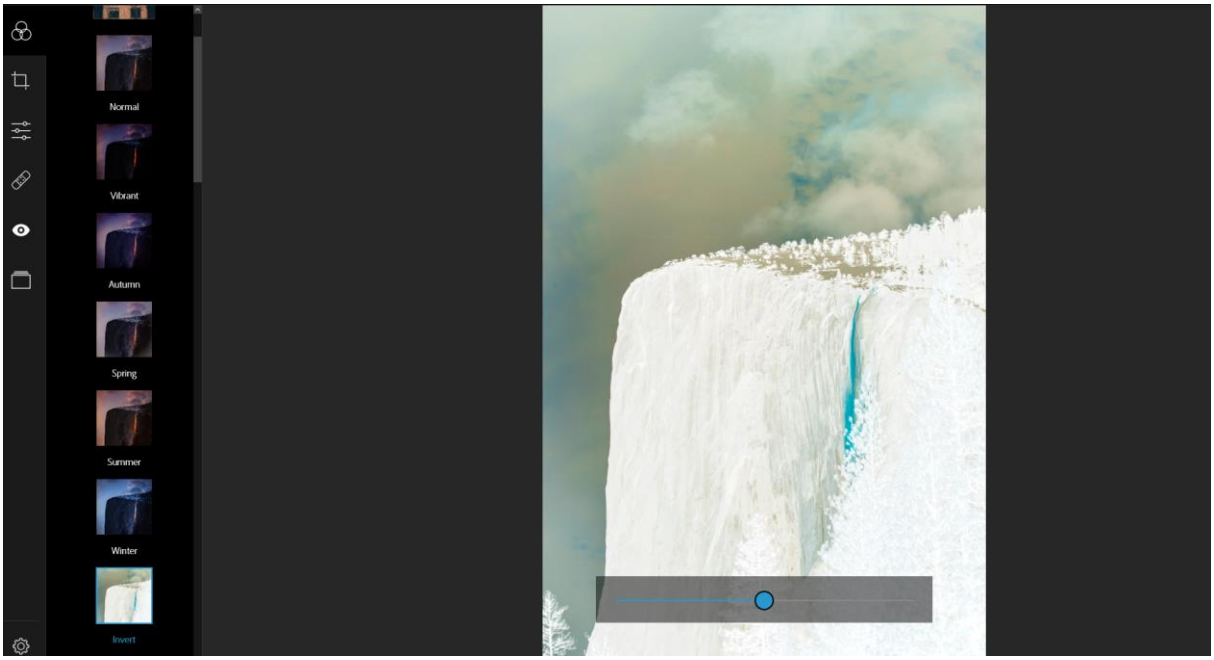


Slika 11: Podopcije aspekta Korekcije

5.1.1. Načini provođenja normalizacije

U ovom podnaslovu govori se o načinu na koji se provela normalizacija nad skupom slika iz praktičnog dijela.

Prvo što je potrebno je unijeti orginalnu sliku u alat Adobe Photoshop Express. Nakon toga odabere se opcija koja se želi, za prikaz načina provođenja normalizacije odabran je aspekt Izgledi i podopcija Basic. Sljedeća slika prikazuje koje još opcije nudi opcija Basic i odabranu opciju u Basic-u, zvanu Invert.



Slika 12: Opcija Invert

Na slici se nalazi klizač uz pomoć kojeg se može odrediti intenzitet opcije Invert, klizač može ići od 0 do 200.

6. Praktični dio

U ovom poglavlju provodi se praktični dio diplomskog rada. Kroz poglavlje će biti ukupno 10 usporedbi za različite opcije normalizacije na setu od 200 slika. Usporedbe su prikazane u tablicama, a svaka od usporedbi je opisana ispod svake tablice. Na početku je prikazan proces kako se provodi normalizacija i kako se izračunava hash za jednu sliku, a kasnije su usporedbe prikazane samo u tablicama.

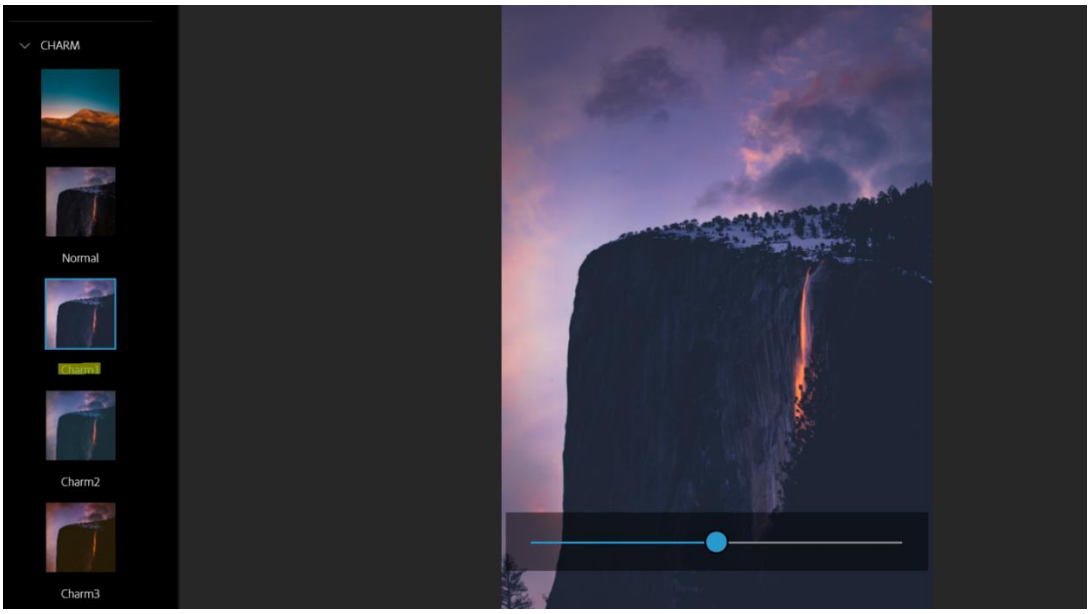
Cilj izrade ovog praktičnog dijela diplomskog rada je pokazati do kojih mjera je moguće normalizirati sliku te koje opcije utječu na potpuno novu hash vrijednost. Sve slike koje se koriste u praktičnom dijelu diplomskog rada su preuzete sa stranice Unsplash gdje se nalaze slike visoke rezolucije te su besplatne za skidanje [29]. Sve slike su iste tematike te prikazuju prirodne ljepote. Kao što je već ranije spomenuto za provođenje normalizacije se koristi alat Adobe Photoshop Express. Ovaj alat je odabran zato što je besplatan i jednostavan je za korištenje te je proces normalizacije relativno lako provesti. Za izračunavanje hash vrijednosti pojedinih slika koristi se Visual Studio Code, s ekstenzijom za Python, zato što sam se prije puno puta susreo s tim alatom, ali i zbog toga što je biblioteka s algoritmima za izračunavanje perceptualnog hasha rađena za Python. Ukupno je 10 usporedbi, a u svakoj po 20 slika, točnije 40 zato što svaka orginalna slika ima i svoju normaliziranu sliku. Normalizacija se provodi sa sljedećim opcijama, a za svaku opciju navedena je i točna podopcija koja je korištena:

- Izgledi > Charm > Charm1
- Izgledi > B&W > Faded B&W
- Izgledi > Nature > Classic
- Izgledi > Pop Color > Aqua
- Izgledi > Duo Tone > Blue Tint
- Korekcije > Light > Exposure
- Korekcije > Color > Vibrance
- Korekcije > Effects > Fade
- Korekcije > Details > Sharpen
- Korekcije > Blur > Full.

Odabrene su opcije takve da ne narušavaju pridodan izgled slike, tako da hash u nekim sličajevima može ostati isti, ali to je i cilj.

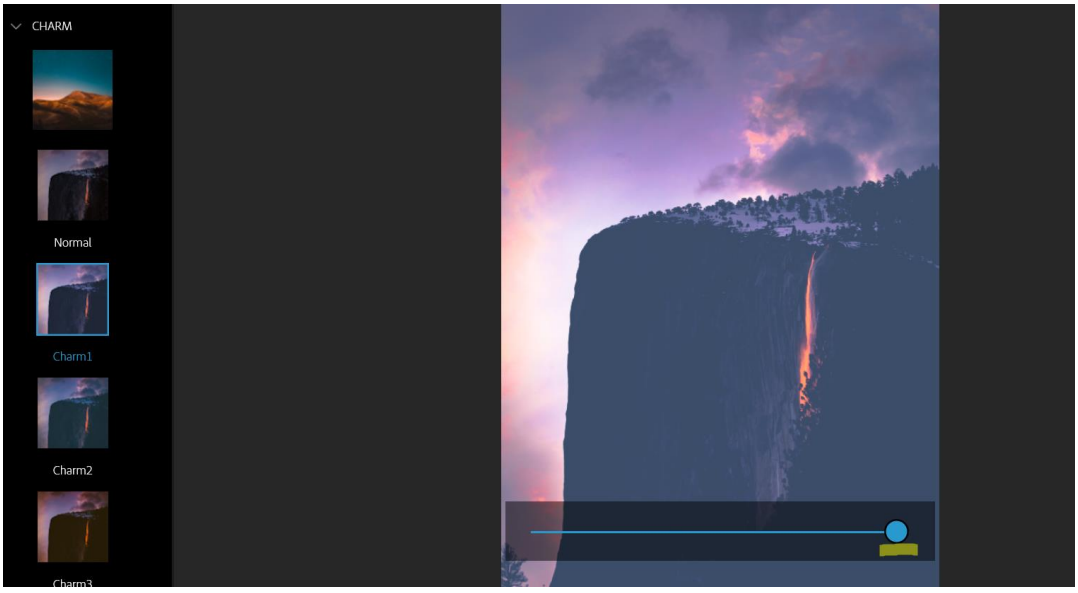
6.1. Usporedba 1

Kod usporedbe 1 radi se normalizacija Izgledi > Charm > Charm1 te se na sljedećoj slici vidi prva slika.



Slika 13: Slika1.jpg u alatu pod opcijom Charm1

Na sljedećoj slici se može vidjeti razlika kada se klizač stavi na vrijednost 200, u ovoj usporedbi za svaku sliku normalizacija je provedena na prikazani način. Svakoj slici klizač se stavio na 200. Opcija daje malo bljeđi prikaz, što na nekim slikama može biti dobro, a na nekima i ne zbog toga što već mogu biti bljeđe.



Slika 14: Slika1.jpg nakon normalizacije

U nastavku slijedi tablica usporedbe za 20 slika. Za svaku orginalnu sliku stavljena je njezina hash vrijednost te hash vrijednost nakon normalizacije. Samo se u ovoj usporedbi prikazuju hash vrijednosti, zbog količine podataka u kasnijim usporedbama biti će navedena samo razlika.

Tablica 3: Usporedba 1

Naziv	A-Hash	P-Hash	D-Hash	W-Hash
slika1.jpg	fffffec080808080	fe80053e7661c1cf	c4b8901c2c39282a	ffffffc0c4808080
slika1_edited.jpg	20fefec0c0c08080	fe81033e7e60c18f	c4b882042d292820	ffffffc0c0c08080
Razlika:	10	6	8	2
slika2.jpg	ffffff3f1f070100	80c17f3f0d60f0fc	e07088f0fcfffff7	ffff7f1f0f000000
slika2_edited.jpg	ffffff3f1f070000	88c47f3f1d62b158	806098e7789f6719	ffff7f1f07010000
Razlika:	1	10	21	2
slika3.jpg	ffff7f1b02000000	c5c3f23c3c1b43c5	20c4d573e6902005	ffffff3b13000000
slika3_edited.jpg	ffff7e1902000000	c5c7f23c3c1943c5	20c4d433a6100000	ffffff3b13000000
Razlika:	2	2	7	0
slika4.jpg	ffffffff7d000000	8bad7442c2771fa8	88f37bbbab790210	fefffff7f09000000
Slika4_edited.jpg	ffffffff79000000	8b897446c2d71fa9	c8f37bfb8b718200	3effff7f69000000
Razlika:	1	6	6	4
slika5.jpg	00e0fffffb000000	e3f3120d8db21b4d	0002090322c08850	c0f8fffffb600000
Slika5_edited.jpg	00e0fdfff0000000	e3f3130c8c9b1b4d	0002094362e08840	e0f8fffffb200000
Razlika:	4	6	4	2
slika6.jpg	ffffff0300000000	a1a7de5a2921d6d2	cf79cecf7fe08704	ffffff7f01000000
Slika6_edited.jpg	ffffff0300000000	a1a5d65a2925d5da	8f71cecf3b448300	fffffef01000000
Razlika:	0	6	9	2
slika7.jpg	0001f7fffffffb00	a93a2bd1149db3a6	fab36f36c4dae2ab	000127bff7ff7900
Slika7_edited.jpg	0001f7fffffffb00	ad3a1ad1149db3a6	d8936f76c4dae2aa	000027bfffff7900
Razlika:	0	4	5	2
slika8.jpg	fffffffc10000000	cf89322659c5372e	96f9e0e0e04d663d	fffffffe10000000
Slika8_edited.jpg	4fffffc10000000	cd89323e49f17706	9ef9e0f0e04d623c	fffffffe10000000
Razlika:	3	10	4	0
slika9.jpg	f0c0fffffffb0000	fdf202ec825aec83	00100000002229f4	e000ffffffff20000
Slika9_edited.jpg	8000fffffffb0000	fdd322ec8252ec92	0000000000222884	8000fffffffb0000
Razlika:	5	6	5	4
slika10.jpg	ffff3e3c10400000	8495e1f90a5ab735	0cc6e4e0328691c3	ffff7e3c18e20000
slika10_edited.jpg	ffff3e3c10420000	8497e0f80a5eb735	09d2e8703296d1c3	ffff7e3c18c20040
Razlika:	1	4	10	2
slika11.jpg	fffffffff000000	bac47abac5386790	40000000101f1b1f	ffffffff00000000
slika11_edited.jpg	fffffffff000000	bbc5713ac5122fd0	000000001e1f1b0d	ffffffff00000000
Razlika:	0	12	6	0
slika12.jpg	ff7e7c1800000000	c9c2d0b43c9dd3e4	a4d0f173b2927270	ffff7c3c18081818
slika12_edited.jpg	ff7e7c1800000000	c9c2f2343c9dc3e4	a4d0e173b2927270	ffff7c3c18081818

Razlika:	0	4	1	0
slika13.jpg	000000000f3f3e3e	956acc8c304f9bb3	b5a518189e7a72f0	c0c0c4844f3f3f3f
slika13_edited.jpg	000000000f3f3e3e	956accb9324e98b3	25a6190c9e7a7270	c0c0cc0c0f3f3f3f
Razlika:	0	8	8	4
slika14.jpg	fffec28080808080	dfc0828e8e2f3e34	e0c01e26246e6c4c	fffec6a296968080
slika14_edited.jpg	7efec28080808000	dfc0828f8e3e3c38	f0c01e2626666e4c	fffecf8296828280
Razlika:	3	6	4	6
slika15.jpg	00387c7e3e1c1c1c	90e33e3b32e2c84f	b9e0e0e8f8b8f8e8	003e7e7e3e1e1c1c
slika15_edited.jpg	0038787e3c1c1c1c	98e53e3b32e2c84d	9d70c0e8f8b0b8a8	003e7e7e3e1e1c1c
Razlika:	2	4	8	0
slika16.jpg	0060707078783800	c5733ccc731c1b38	c0c0c0c0c0e0f6e6	387878fc7c7c3a02
slika16_edited.jpg	0060607078383800	c7333ccee31c3f00	40c0c0c0c2e2760e	7078f8f87c783a03
Razlika:	2	10	8	6
slika17.jpg	000000ffffff0000	c5935ee933cd4832	f0f0f0f09fc322c0	00003effffffe000
slika17_edited.jpg	000000ffffff8000	c0177fec934d4932	f0f0f0f09fc323d8	00003cfffffff000
Razlika:	1	12	3	2
slika18.jpg	febd230f3d3f1000	8ef874cd10b4e569	7871e6dcf1e3e4f0	febd230f3d3f1000
slika18_edited.jpg	febd230f3d3f1000	8edc54cb12b4a56d	7871e6dcf1e3e470	febd130f3d3f1000
Razlika:	0	8	1	2
slika19.jpg	3f1f0f0f1f7f7f00	828b55c32fdc7385	fdfffffffcdfeff	1f0f07070f7f3f00
slika19_edited.jpg	1f0f07070f7f3f00	a29b59922ddd7285	fdfffffffcdfeff	1f0f0707077f7f00
Razlika:	6	10	0	2
slika20.jpg	00f4e0c77f700000	c3f2188f95386bc3	c08cc90ec04398f0	e0f4f5e7ff780000
Slika20_edited.jpg	0000e0c77f700000	c7f0388f94786b83	c08c891cc443bc40	e0f0f4effff80000
Razlika:	5	6	9	4
Ukupno razlike:	46 bita	140 bita	127 bita	46 bita

Tokom izrade ove usporedbe A-Hash, P-Hash i D-Hash su bili relativno brzi u izračunavanju hash vrijednosti s obzirom da se radi o 40 slika, dok je W-Hash trajao znatno sporije, ali je dao isti rezultat u ukupnim razikama kao i A-Hash. Razlika je izračunata na temelju hash vrijednosti orginalne slike i normalizirane slike te se tako dobije broj bitova koji se razlikuju, zbog toga što su sva četiri algoritma 64 bitna. P-Hash algoritam za ovu normalizaciju dao je jako velike razlike između slika, što zapravo govori da je na temelju provedene normalizacije jako osjetiljiv na gore prikazanu opciju Charm1.

Na sljedećoj slici može se vidjeti kako se provodi računanje hash vrijednosti za sliku pod nazivom slika1.jpg. Prema primjeru sa stranice na kojoj se nalazi biblioteka ImageHash napravljeno je računanje hash vrijednosti za sve slike [27].

```
#A-hash

slika1Hash = imagehash.average_hash(Image.open("C:\\Users\\Matej\\Desktop\\set_slikadip\\slika1.jpg"))
print('Hash orginalne slike1: ' + str(slika1Hash))

slika1edited = imagehash.average_hash(Image.open("C:\\Users\\Matej\\Desktop\\set_slikadip\\editane Izgledi  Charm  Charm1\\slika1_edited.jpg"))
print('Hash editane slike1: ' + str(slika1edited))

if(slika1Hash == slika1edited):
    print("Hashevi isti!")
else:
    print("Razlika izmedu hasha:" + str(slika1Hash - slika1edited))
```

Slika 15: Prikaz izračunavanja percepralnog A-Hash

Iduća slika prikazuje rezultat koji funkcija *average_hash()* daje.

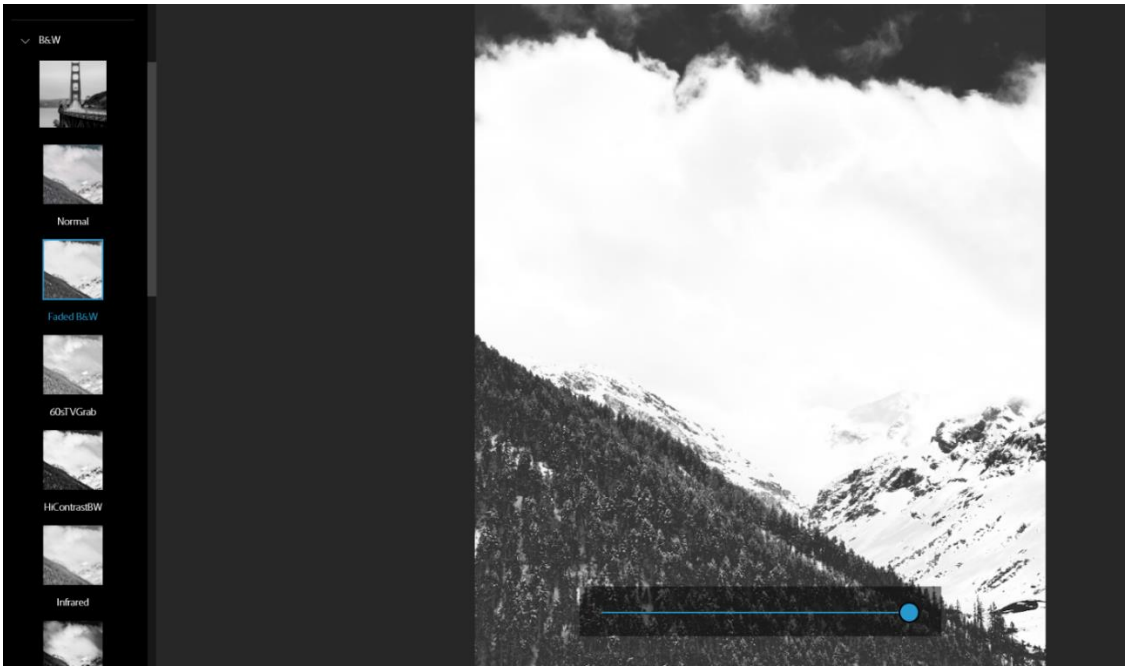
```
PS C:\Users\Matej> & C:/Users/Matej/AppData/Local/Programs/Python/Python37/python.exe c:/Users/Matej/Desktop/dip_primjeri.py
Hash orginalne slike1: fffffec080808080
Hash editane slike1: 20fefec0c0c08080
Razlika izmedu hasha:10
PS C:\Users\Matej>
```

Slika 16: Razultat funkcije average_hash()

Preostali algoritmi se računaju na isti način, samo što se umjesto funkcije *average_hash()* koriste, za P-Hash funkcija *phash()*, za D-Hash funkcija *dhash()* i za W-Hash funkcija *whash()* pa stoga nema potrebe to prikazivati, zato što je postupak isti.

6.2. Usporedba 2

Kod usporedbe 2 radi se normalizacija Izgledi > B&W > Faded B&W.



Slika 17: Opcija Faded B&W

Na prethodnoj slici vidi se primjena opcije Faded B&W gdje se slika iz RGB palete boja pretvara u paletu sivijih tonova. Kod ove opcije se očekuje dosta promjena zbog toga što se mijenja cijela paleta boja.

Tablica 4: Usporedba 2

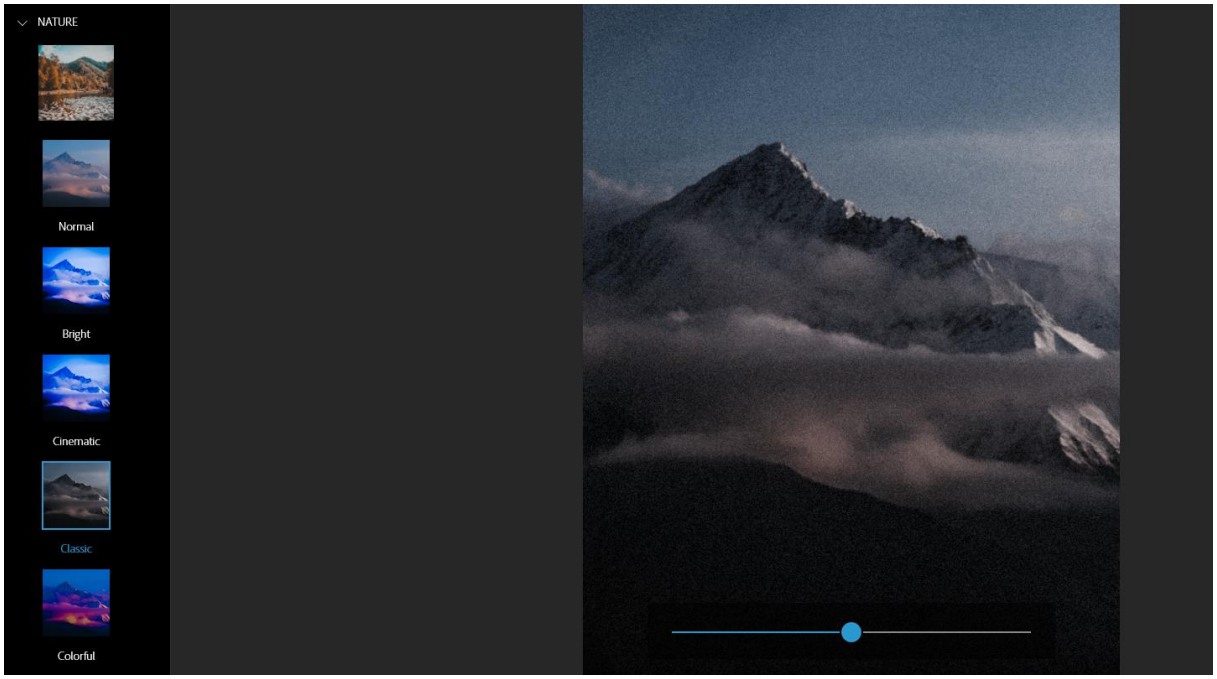
Naziv	A-Hash	P-Hash	D-Hash	W-Hash
slika21.jpg – slika21_edited.jpg	0	8	12	2
slika22.jpg – slika22_edited.jpg	17	12	10	20
slika23.jpg – slika23_edited.jpg	31	26	29	28
slika24.jpg – slika24_edited.jpg	3	2	6	4
slika25.jpg – slika25_edited.jpg	1	6	6	0
slika26.jpg – slika26_edited.jpg	5	4	5	6
slika27.jpg – slika27_edited.jpg	4	4	15	0
slika28.jpg – slika28_edited.jpg	1	6	7	2
slika29.jpg – slika29_edited.jpg	1	10	2	0
slika30.jpg – slika30_edited.jpg	1	2	4	2
slika31.jpg – slika31_edited.jpg	17	4	21	2
slika32.jpg – slika32_edited.jpg	3	2	6	2
slika33.jpg – slika33_edited.jpg	5	10	2	2
slika34.jpg – slika34_edited.jpg	1	6	7	0
slika35.jpg – slika35_edited.jpg	0	4	4	2
slika36.jpg – slika36_edited.jpg	0	8	5	0
slika37.jpg – slika37_edited.jpg	9	6	7	6
slika38.jpg – slika38_edited.jpg	1	6	8	2

slika39.jpg – slika39_edited.jpg	6	8	5	8
slika40.jpg – slika40_edited.jpg	1	2	5	2
Ukupno razlike:	107 bitova	136 bitova	166 bitova	88 bitova

Nakon provedene usporedbe rezultati su dosta lošiji od rezultata iz usporedbe 1. U ovoj usporedbi najlošiji je bio D-Hash, a najbolji W-Hash. Može se primjetiti da kada je neka slika dosta promjenila svoj izgled, razlike su vidljive na svim algoritmima, najbolji primjer može biti slika23.jpg.

6.3. Usporedba 3

Kod usporedbe 3 radi se normalizacija Izgledi > Nature > Classic.



Slika 18: Opcija Classic

Ova opcija daje slikama klasičan izgled, dobiva se dojam da se radi o slikama koje su slikane s pravim fotoaparatima, opcija ne mijenja drastično izgled slike, ali daje dojam kao da se malo zamutila i da su neki pikseli postali crni.

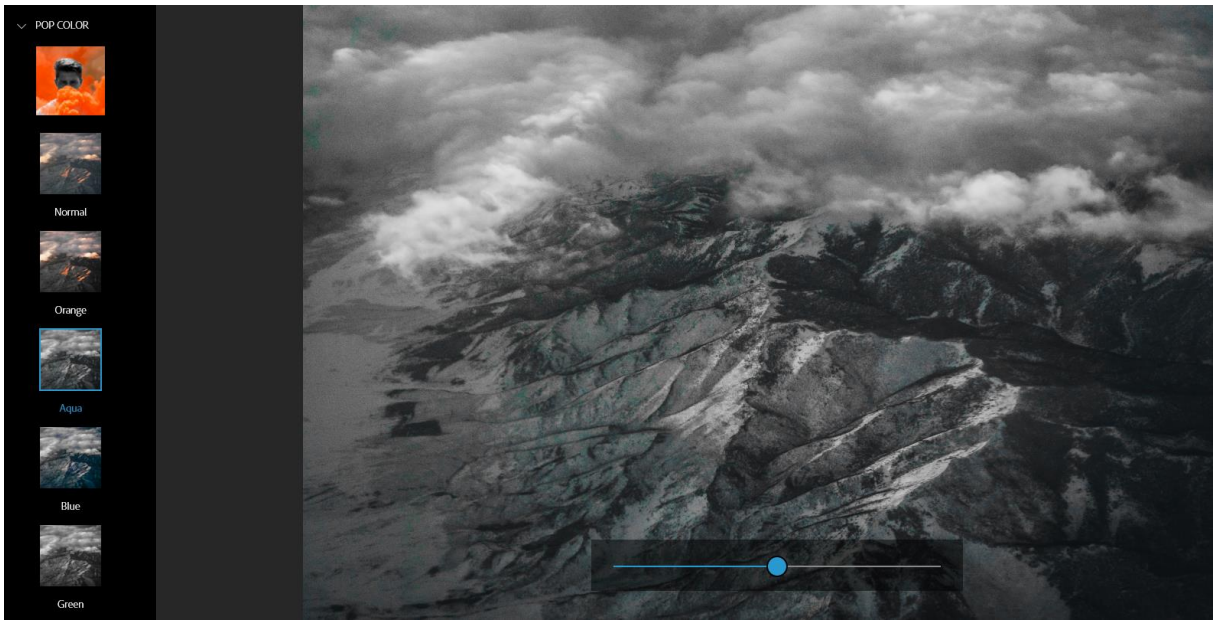
Tablica 5: Usporedba 3

Naziv	A-Hash	P-Hash	D-Hash	W-Hash
slika41.jpg – slika41_edited.jpg	7	4	7	2
slika42.jpg – slika42_edited.jpg	3	16	5	0
slika43.jpg – slika43_edited.jpg	5	6	5	2
slika44.jpg – slika44_edited.jpg	6	10	2	0
slika45.jpg – slika45_edited.jpg	4	8	2	0
slika46.jpg – slika46_edited.jpg	2	2	0	2
slika47.jpg – slika47_edited.jpg	5	10	6	6
slika48.jpg – slika48_edited.jpg	2	8	1	2
slika49.jpg – slika49_edited.jpg	6	4	6	4
slika50.jpg – slika50_edited.jpg	4	16	6	2
slika51.jpg – slika51_edited.jpg	0	10	6	0
slika52.jpg – slika52_edited.jpg	4	10	4	2
slika53.jpg – slika53_edited.jpg	0	12	5	0
slika54.jpg – slika54_edited.jpg	3	8	6	2
slika55.jpg – slika55_edited.jpg	7	8	1	2
slika56.jpg – slika56_edited.jpg	10	8	5	4
slika57.jpg – slika57_edited.jpg	3	6	3	0
slika58.jpg – slika58_edited.jpg	2	2	6	2
slika59.jpg – slika59_edited.jpg	9	8	5	4
slika60.jpg – slika60_edited.jpg	0	6	5	4
Ukupno razlike:	82 bita	162 bita	94 bita	40 bita

Rezultati ove usporedbe pokazali su da je W-Hash imao najmanje razlika korištenjem opcije Classic. P-Hash ima jako puno razlika.

6.4. Usporedba 4

Kod usporedbe 4 radi se normalizacija Izgledi > Pop Color > Aqua.



Slika 19: Opcija Aqua

S prethodne slike može se vidjeti da opcija Aqua reagira na tonove koji su plavije boje te se njih najviše stavlja u fokus.

Tablica 6: Usporedba 4

Naziv	A-Hash	P-Hash	D-Hash	W-Hash
slika61.jpg – slika61_edited.jpg	1	4	0	2
slika62.jpg – slika62_edited.jpg	2	4	11	0
slika63.jpg – slika63_edited.jpg	0	4	9	0
slika64.jpg – slika64_edited.jpg	3	10	3	2
slika65.jpg – slika65_edited.jpg	2	6	4	2

slika66.jpg – slika66_edited.jpg	7	4	1	6
slika67.jpg – slika67_edited.jpg	8	6	5	4
slika68.jpg – slika68_edited.jpg	2	4	6	2
slika69.jpg – slika69_edited.jpg	0	2	7	2
slika70.jpg – slika70_edited.jpg	3	8	6	4
slika71.jpg – slika71_edited.jpg	3	8	8	4
slika72.jpg – slika72_edited.jpg	5	2	6	4
slika73.jpg – slika73_edited.jpg	8	8	11	12
slika74.jpg – slika74_edited.jpg	5	12	5	6
slika75.jpg – slika75_edited.jpg	1	8	8	0
slika76.jpg – slika76_edited.jpg	3	4	0	2
slika77.jpg – slika77_edited.jpg	2	18	17	6
slika78.jpg – slika78_edited.jpg	3	6	1	4
slika79.jpg – slika79_edited.jpg	2	4	3	2
slika80.jpg – slika80_edited.jpg	2	6	3	4
Ukupno razlike:	62 bita	128 bita	114 bita	68 bita

Ovom usporedbom očekivane su velike razlike, zato što opcija daje dosta tamne slike s izraženim plavim tonovima, ako ih ima na slici. A-Hash algoritam ima najmanje razlika, dok P-Hash algoritam ima najviše razlika u bitovima.

6.5. Usporedba 5

Kod usporedbe 5 radi se normalizacija Izgledi > Duo Tone > Blue Tint.

Slika 20: Opcija Blue Tint

S prethodne slike može se vidjeti da opcija Blue Tint daje plavi ton cijelokupnoj slici. Razlog iz kojeg sam odabrao ovu opciju jest taj da pokažem kako se može izmjeniti hash vrijednost samo promjenom boje na slici, a ova usporedba je na većini slika to i pokazala.

Tablica 7: Usporedba 5

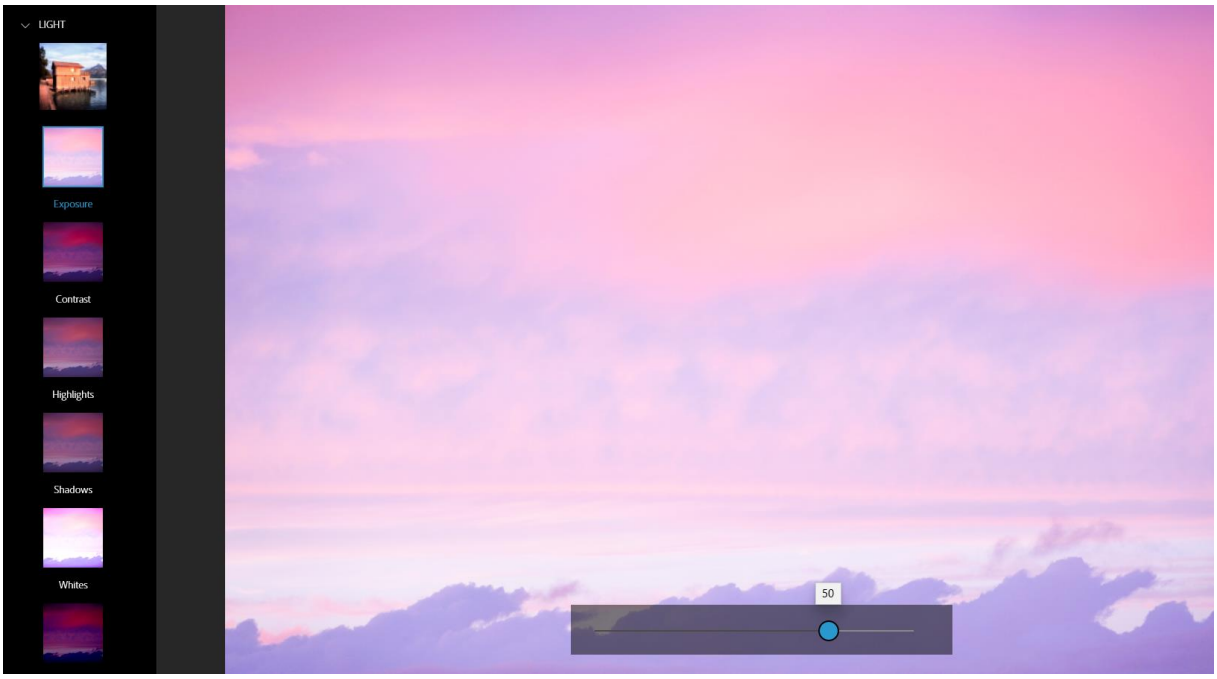
Naziv	A-Hash	P-Hash	D-Hash	W-Hash
slika81.jpg – slika81_edited.jpg	1	2	0	0
slika82.jpg – slika82_edited.jpg	2	4	0	4
slika83.jpg – slika83_edited.jpg	3	2	1	2
slika84.jpg – slika84_edited.jpg	0	2	1	2
slika85.jpg – slika85_edited.jpg	0	2	8	0
slika86.jpg – slika86_edited.jpg	0	0	1	0
slika87.jpg – slika87_edited.jpg	0	0	5	2

slika88.jpg – slika88_edited.jpg	1	2	2	0
slika89.jpg – slika89_edited.jpg	0	4	2	0
slika90.jpg – slika90_edited.jpg	4	6	3	4
slika91.jpg – slika91_edited.jpg	5	4	3	4
slika92.jpg – slika92_edited.jpg	1	2	1	0
slika93.jpg – slika93_edited.jpg	2	0	3	4
slika94.jpg – slika94_edited.jpg	5	6	2	4
slika95.jpg – slika95_edited.jpg	2	2	0	2
slika96.jpg – slika96_edited.jpg	4	0	2	0
slika97.jpg – slika97_edited.jpg	0	2	2	2
slika98.jpg – slika98_edited.jpg	2	2	6	4
slika99.jpg – slika99_edited.jpg	3	4	1	2
slika100.jpg – slika100_edited.jpg	8	10	11	12
Ukupno razlike:	43 bita	56 bita	54 bita	48 bita

Ova opcija je pokazala da su zapravo jako male razlike kada se promjeni boja slike. A-Hash algoritam je najmanje osjetljiv na opciju Blue Tint, dok P-Hash algoritam ima najviše razlike.

6.6. Usporedba 6

Kod usporedbe 6 radi se normalizacija Korekcije > Light > Exposure.



Slika 21: Opcija Exposure

Kod ove opcije Exposure, sama riječ kaže da je to izražavanje pa je to upravo s ovom opcijom i slučaj. Kada se klizač pomakne u desno do izražaja dolaze svijetliji tonovi, a kada se klizač pomakne u lijevo do izražaja dolaze tamniji tonovi. Za ovu usporedbu klizač je postavljen na 50 tako da slike ne budu pre svijetle.

Tablica 8: Usporedba 6

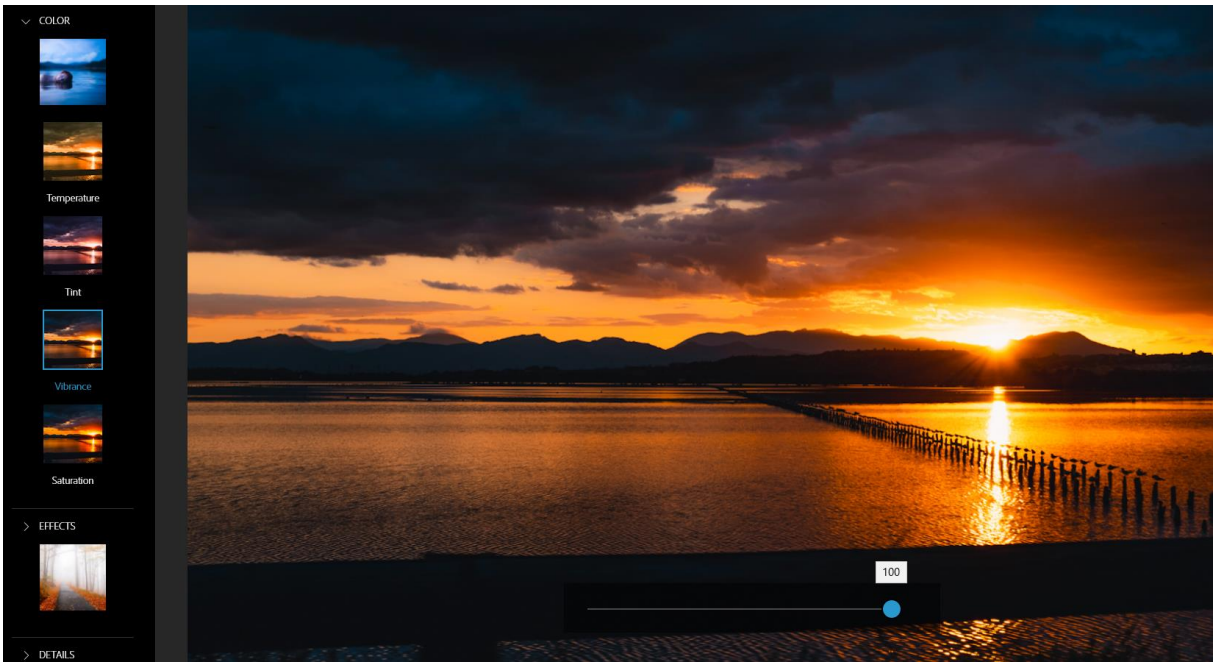
Naziv	A-Hash	P-Hash	D-Hash	W-Hash
slika101.jpg – slika101_edited.jpg	4	12	1	2
slika102.jpg – slika102_edited.jpg	13	6	3	4
slika103.jpg – slika103_edited.jpg	7	12	5	8
slika104.jpg – slika104_edited.jpg	17	22	24	16
slika105.jpg – slika105_edited.jpg	10	10	6	2
slika106.jpg – slika106_edited.jpg	10	10	3	0

slika107.jpg – slika107_edited.jpg	5	24	3	0
slika108.jpg – slika108_edited.jpg	3	12	6	2
slika109.jpg – slika109_edited.jpg	3	10	5	4
slika110.jpg – slika110_edited.jpg	4	8	7	4
slika111.jpg – slika111_edited.jpg	0	8	5	2
slika112.jpg – slika112_edited.jpg	16	22	3	0
slika113.jpg – slika113_edited.jpg	7	8	8	4
slika114.jpg – slika114_edited.jpg	6	20	5	4
slika115.jpg – slika115_edited.jpg	10	8	6	4
slika116.jpg – slika116_edited.jpg	6	6	3	0
slika117.jpg – slika117_edited.jpg	7	14	5	2
slika118.jpg – slika118_edited.jpg	9	14	13	4
slika119.jpg – slika119_edited.jpg	8	16	5	4
slika120.jpg – slika120_edited.jpg	13	22	8	4
Ukupno razlike:	158 bita	264 bita	124 bita	70 bita

Kod ove usporedbi zapravo se najbolje može vidjeti kako se izražavanjem tonova na slici mogu dobiti dosta velike razlike. Klizač je bio postavljen na 50, a zapravo bi i jako malim pomakom došlo do promjena, možda ne ovakvih proporcija već malo manjih. W-Hash imao je najmanje razlika, dok je P-Hash opet imao najviše razlika.

6.7. Usporedba 7

Kod usporedbe 7 radi se normalizacija Korekcije > Color > Vibrance.



Slika 22: Opcija Vibrance

Opcija Vibrance daje sliki prirodniji izgled, kao što se sa slike može vidjeti slika ima prirodan izgled i dobiva se osjećaj kao da smo u slici. Očekivano je da ne bi trebalo biti velikih razlika u hash vrijednostima, ali ova usporedba pokazuje drugačije, kao što možemo vidjeti u sljedećoj tablici.

Tablica 9: Usporedba 7

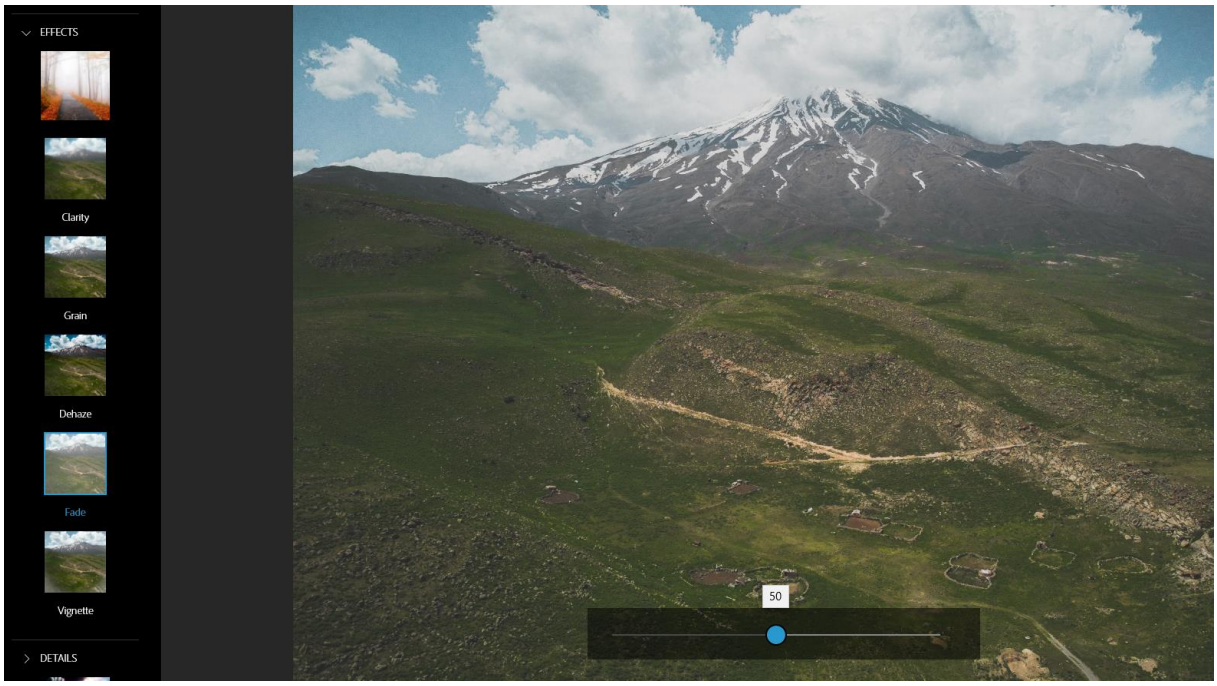
Naziv	A-Hash	P-Hash	D-Hash	W-Hash
slika121.jpg – slika121_edited.jpg	0	4	1	0
slika122.jpg – slika122_edited.jpg	1	2	3	0
slika123.jpg – slika123_edited.jpg	2	8	3	4
slika124.jpg – slika124_edited.jpg	4	2	2	8
slika125.jpg – slika125_edited.jpg	3	4	1	2
slika126.jpg – slika126_edited.jpg	1	4	1	2

slika127.jpg – slika127_edited.jpg	0	4	6	2
slika128.jpg – slika128_edited.jpg	1	0	0	0
slika129.jpg – slika129_edited.jpg	5	2	2	4
slika130.jpg – slika130_edited.jpg	1	0	3	0
slika131.jpg – slika131_edited.jpg	2	2	6	2
slika132.jpg – slika132_edited.jpg	5	4	2	2
slika133.jpg – slika133_edited.jpg	1	4	2	0
slika134.jpg – slika134_edited.jpg	0	4	2	0
slika135.jpg – slika135_edited.jpg	0	0	1	0
slika136.jpg – slika136_edited.jpg	1	6	7	10
slika137.jpg – slika137_edited.jpg	3	4	3	2
slika138.jpg – slika138_edited.jpg	2	4	3	4
slika139.jpg – slika139_edited.jpg	0	6	3	4
slika140.jpg – slika140_edited.jpg	0	2	1	4
Ukupno razlike:	32 bita	66 bita	52 bita	50 bita

Iznenađujuće ova usporedba je dala dosta razlika s obzirom na to kako slike izgledaju nakon normalizacije. Najmanje razlika ima A-Hash algoritam, a najviše P-Hash algoritam.

6.8. Usporedba 8

Kod usporedbe 8 radi se normalizacija Korekcije > Effects > Fade.



Slika 23: Opcija Fade

Kod ove opcije Fade slika se izbljeđuje, ali kako nisam htio pretjerati s izbljeđivanjem klizač je postavljen na 50.

Tablica 10: Usporedba 8

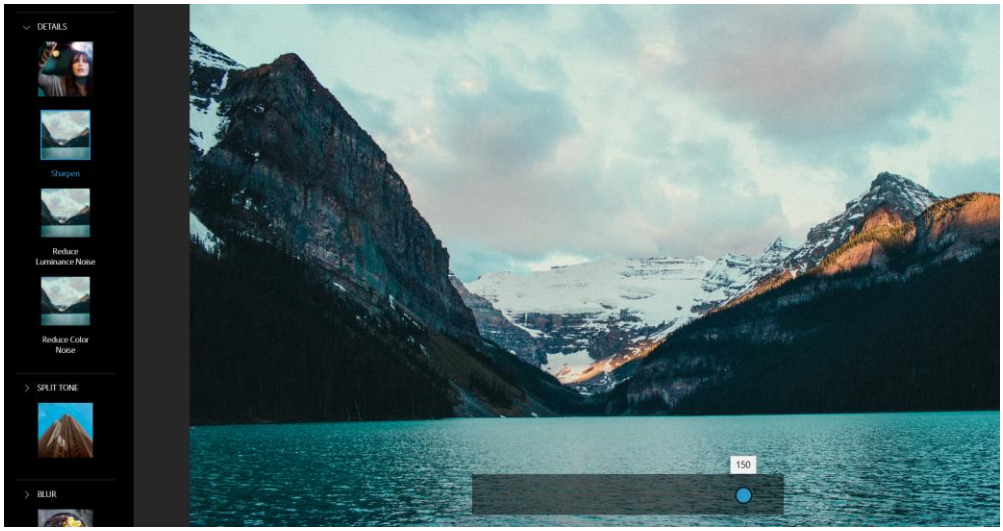
Naziv	A-Hash	P-Hash	D-Hash	W-Hash
slika141.jpg – slika141_edited.jpg	2	4	3	2
slika142.jpg – slika142_edited.jpg	1	6	0	0
slika143.jpg – slika143_edited.jpg	3	0	3	0
slika144.jpg – slika144_edited.jpg	2	4	3	2
slika145.jpg – slika145_edited.jpg	0	4	0	2
slika146.jpg – slika146_edited.jpg	1	2	1	0
slika147.jpg – slika147_edited.jpg	0	4	1	2

slika148.jpg – slika148_edited.jpg	1	2	0	2
slika149.jpg – slika149_edited.jpg	2	0	2	0
slika150.jpg – slika150_edited.jpg	0	0	0	0
slika151.jpg – slika151_edited.jpg	0	2	1	2
slika152.jpg – slika152_edited.jpg	4	4	1	0
slika153.jpg – slika153_edited.jpg	6	2	8	2
slika154.jpg – slika154_edited.jpg	2	2	4	4
slika155.jpg – slika155_edited.jpg	0	2	1	0
slika156.jpg – slika156_edited.jpg	2	2	2	0
slika157.jpg – slika157_edited.jpg	4	2	2	2
slika158.jpg – slika158_edited.jpg	2	2	2	0
slika159.jpg – slika159_edited.jpg	0	2	4	2
slika160.jpg – slika160_edited.jpg	0	2	1	0
Ukupno razlike:	32 bita	48 bita	39 bita	22 bita

Usporedba je pokazala kako izbljeđivanje još uvijek daje dosta velike razlike, W-Hash ima najmanje razlika, dok P-Hash ima najviše razlika.

6.9. Usporedba 9

Kod usporedbe 9 radi se normalizacija Korekcije > Details > Sharpen.



Slika 24: Opcija Sharpen

Opcija Sharpen je najjednostavnija što se tiče promjena na slici, sliku koja je možda malo zamućena ili lošije kvalitete izoštri do te mjere da slika izgleda kao da je slikana profesionalnim fotoaparatom. Opcija zapravo uopće ne bi trebala napraviti promjene u hash vrijednosti, ali pogledajmo tablicu usporedbe 9.

Tablica 11: Usporedba 9

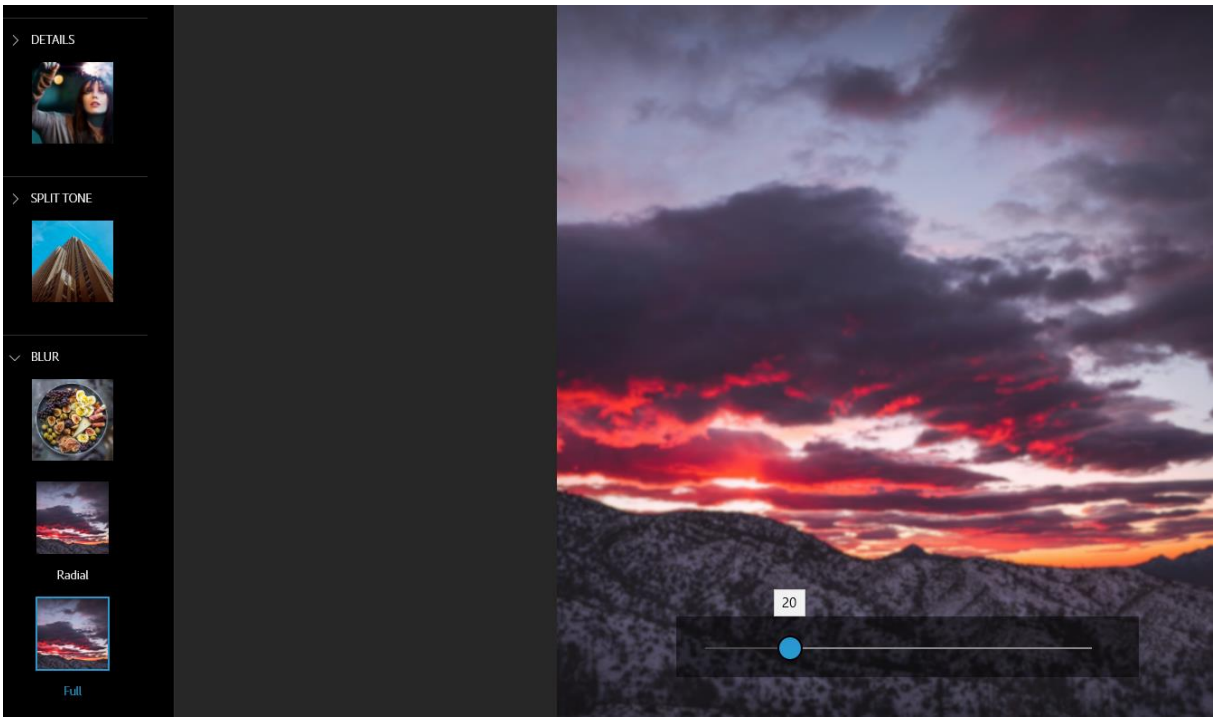
Naziv	A-Hash	P-Hash	D-Hash	W-Hash
slika161.jpg – slika161_edited.jpg	0	0	1	0
slika162.jpg – slika162_edited.jpg	0	0	0	0
slika163.jpg – slika163_edited.jpg	0	0	0	2
slika164.jpg – slika164_edited.jpg	1	0	0	0
slika165.jpg – slika165_edited.jpg	1	2	0	0
slika166.jpg – slika166_edited.jpg	0	2	1	0
slika167.jpg – slika167_edited.jpg	0	2	0	0

slika168.jpg – slika168_edited.jpg	0	2	1	0
slika169.jpg – slika169_edited.jpg	1	0	1	0
slika170.jpg – slika170_edited.jpg	2	2	0	2
slika171.jpg – slika171_edited.jpg	0	0	2	0
slika172.jpg – slika172_edited.jpg	1	0	1	0
slika173.jpg – slika173_edited.jpg	0	2	1	0
slika174.jpg – slika174_edited.jpg	1	0	0	0
slika175.jpg – slika175_edited.jpg	0	2	0	0
slika176.jpg – slika176_edited.jpg	0	2	0	0
slika177.jpg – slika177_edited.jpg	1	0	1	0
slika178.jpg – slika178_edited.jpg	0	2	0	2
slika179.jpg – slika179_edited.jpg	1	2	3	4
slika180.jpg – slika180_edited.jpg	0	0	2	0
Ukupno razlike:	9 bita	20 bita	14 bita	10 bita

Tokom izrade i provedbe dosadašnjih usporedbi, kod ove usporedbe ostao sam najviše iznenađen. Smatrao sam kako izoštravanje slike neće uopće promijeniti hash vrijednost, ali prevario sam se, dakako postoji dosta slika kojima se hash vrijednost nije promjenila, ali izoštravanje koje bi samo po sebi trebalo sliku učiniti boljom, ipak mijenja hash vrijednost. Ovdje se najboljim pokazao A-Hash algoritam koji je kod samo 8 slika imao razliku u hash vrijednosti, i to od 1 bita, dok kod jedne slike 2 bita. Najviše razlika je kod P-Hash algoritma.

6.10. Usporedba 10

Kod usporedbe 10 radi se normalizacija Korekcije > Blur > Full.



Slika 25:Opcija Blur

Opcija Blur je dakle samo zamućivanje slike, slike su zamućene za 20% kako ne bi bile previše mutne i izgubile svoj prirodan izgled. Očekivano je da ova opcija dosta izmjeni hash vrijednosti, ali pogledajmo rezultate u tablici usporedbe 10.

Tablica 12: Usporedba 10

Naziv	A-Hash	P-Hash	D-Hash	W-Hash
slika181.jpg – slika181_edited.jpg	0	0	1	0
slika182.jpg – slika182_edited.jpg	1	0	3	0
slika183.jpg – slika183_edited.jpg	1	2	4	2
slika184.jpg – slika184_edited.jpg	0	0	1	0
slika185.jpg – slika185_edited.jpg	0	0	2	2
slika186.jpg – slika186_edited.jpg	1	4	0	0

slika187.jpg – slika187_edited.jpg	0	2	0	0
slika188.jpg – slika188_edited.jpg	0	0	1	0
slika189.jpg – slika189_edited.jpg	0	2	2	0
slika190.jpg – slika190_edited.jpg	0	0	1	0
slika191.jpg – slika191_edited.jpg	0	0	1	0
slika192.jpg – slika192_edited.jpg	0	2	0	0
slika193.jpg – slika193_edited.jpg	0	2	1	0
slika194.jpg – slika194_edited.jpg	0	0	0	0
slika195.jpg – slika195_edited.jpg	1	2	2	0
slika196.jpg – slika196_edited.jpg	1	2	0	0
slika197.jpg – slika197_edited.jpg	0	0	0	0
slika198.jpg – slika198_edited.jpg	0	2	2	0
slika199.jpg – slika199_edited.jpg	1	0	0	0
slika200.jpg – slika200_edited.jpg	1	0	1	0
Ukupno razlike:	7 bita	20 bita	22 bita	4 bita

Kod ove usporedbe i korištenja ove opcije, rezultati su pokazali da zamućivanje slike ne mijenja hash vrijednosti kako sam mislio. Najmanje razlika ima W-Hash algoritam, dok najviše ima D-Hash algoritam.

6.11. Konačna usporedba rezultata

U ovom podnaslovu prikazani su svi rezultati iz prethodnih usporedbi te su prokomentirani i dani konačni zaključci. Sljedeća tablica prikazuje konačne rezultate usporedbi odnosno ukupne razlike prema odabranim opcijama normalizacije.

Tablica 13: Konačni rezultati usporedbi

Naziv opcije	Naziv usporedbe	A-Hash	P-Hash	D-Hash	W-Hash
Izgledi > Charm > Charm1	Usporedba 1	46	140	127	46
Izgledi > B&W > Faded B&W	Usporedba 2	107	136	166	88
Izgledi > Nature > Classic	Usporedba 3	82	162	94	40
Izgledi > Pop Color > Aqua	Usporedba 4	62	128	114	68
Izgledi > Duo Tone > Blue Tint	Usporedba 5	43	56	54	48
Korekcije > Light > Exposure	Usporedba 6	158	264	124	70
Korekcije > Color > Vibrance	Usporedba 7	32	66	52	50
Korekcije > Effects > Fade	Usporedba 8	32	48	39	22
Korekcije > Details > Sharpen	Usporedba 9	9	20	14	10
Korekcije > Blur > Full	Usporedba 10	7	20	22	4

Na samom kraju važno je dati završnu misao i zaključak. Cilj ove usporedbe je zadovoljen. Određenim normalizacijama slike htjelo se pokazati kako se može promijeniti i utjecati na perceptualni hash. Na korištenim algoritmima, može se vidjeti kako je Usporedba 6 rezultirala najvećim razlikama u bitovima, dok je zadnja Usporedba 10 rezultirala najmanjim razlikama u bitovima. Moglo bi se reći kako su algoritmi P-Hash i D-Hash najosjetljiviji na promjene, dok algoritmi A-Hash i W-Hash nisu tako osjetljivi. Što se tiče samog rada i brzine algoritama, najbrži su bili A-Hash, P-Hash i D-Hash, dok je algoritam W-Hash najsporiji. W-Hash je najsporiji, ali je dao najbolje rezultate u većini provedenih usporedbi. A-Hash algoritam je poprilično blizu W-Hash algoritmu, ali zbog svoje brzine je zapravo i najbolji. U prethodnim poglavljima je zapravo i navedeno kako su preostala tri algoritma temeljeni na A-Hash algoritmu pa nije niti čudo da je dao dobre rezultate. Kada govorimo o normalizaciji, u krivim rukama je to oružje uz pomoć kojeg se zapravo ovi algoritmi osjećaju bespomoćnima. Najgora činjenica je ta da se normalizacija može provesti s najjednostavnijim alatom za uređivanje slika, a da je pri tome alat potpuno besplatan.

7. Zaključak

Ovim diplomskim radom htio se pokazati utjecaj normalizacije slike na perceptualni hash. Kroz teorijski dio došao sam do potrebnih znanja i saznanja kako bih mogao provesti i izraditi praktični dio diplomskog rada. U praktičnom dijelu spomenute su opcije prema kojima su se radile usporedbe kako bi se dokazao utjecaj normalizacije slika na njihov perceptualni hash. Detaljnom porevedbom usporedbi dolazi se do nekoliko zaključaka. Opcije normalizacije koje su se koristile su sljedeće: Charm1, Faded B&W, Classic, Aqua, Blue Tint, Exposure, Vibrance, Fade, Sharpen i Blur. Pojedine usporedbe pokazale su stvarno iznenađujuće rezultate. Na samom kraju usporedbi prikazani su svi rezultati u jednoj tablici iz koje možemo doći do prvog zaključka, a to je da provedbom normalizacije definitivno dolazi do utjecaja na perceptualni hash. Svaki algoritam koji je korišten, pokazao je razlike u bitovima između orginalne slike i slike nad kojom je provedena normalizacija, tako da utjecaj svakako postoji. Sljedeći zaključak je taj da prilikom provedbe normalizacije, ona ne mora biti kompleksna. Naime, opcije ili načini na koji se normalizacija provodi ne mora biti kompleksan, dovoljan je besplatan alat za uređivanje slika kako bi se normalizacija provela. Dakle zaključak iz ovog je sljedeći, normalizacija slika u pogrešnim rukama može rezultirati raznim kriminalnim radnjama te može dovesti u opasnost nevine ljude. Naravno normalizacija slika se ne može ukinuti ili zabraniti, zato što ju je jednostavno napraviti, a i pritom postoje besplatni alati za to, ali je moguće napraviti kompleksnije algoritme koji bi imali bolje razvijenu „percepciju“. Način na koji bi algoritmi bolje mogli percipirati dobivene slike, kao i mi ljudi i zaključiti da se radi o istim ili sličnim slikama, to povlači druga i još kompleksnija pitanja, a to su da li je rješenje korištenje umjetne inteligencije ili bilo kakav način upotrebe novijih tehnologija. Dakle može se zaključiti da postoji jako puno prostora za napredak kod izrade perceptualnih hash algoritama, ali da li je taj napredak isplativ i moguće provesti? Dakako odgovor na ovo pitanje nećemo zasigurno znati u narednom periodu.

Ovi algoritmi koji su se koristili, relativno su zastarjeli, ali su poslužili u cilju ovog diplomskog rada, a to je dokazati utjecaj normalizacije slike na njezin perceptualni hash.

Popis literature

[1] theastrologypage, „Što je hash funkcija? - definicija iz tehopedije“, 2022. [Na internetu]. Dostupno na: https://hr.theastrologypage.com/hash-function [Pristupljeno: 11.07.2022.].

[2] Kurt Mehlhorn, Peter Sanders, „Hash Tables and Associative Arrays“, 2008. [Na internetu]. Dostupno na: https://people.mpi-inf.mpg.de/~mehlhorn/ftp/Toolbox/HashTables.pdf [Pristupljeno: 11.07.2022.].

[3] Charles E. Leiserson, „Amortized Algorithms, Table Doubling, Potential Method“, 2009. [Na internetu]. Dostupno na: http://videolectures.net/mit6046jf05_leiserson_lec13/ [Pristupljeno: 11.07.2022.].

[4] myservername.com, „Hash tablica u C++: Programi za implementaciju Hash Tablice i Hash karte“, (bez dat.). [Na internetu]. Dostupno na: https://hr.myservername.com/hash-table-c-programs-implement-hash-table [Pristupljeno: 11.07.2022.]

[5] Warbletoncouncil, „Hi-kvadrat test (χ^2): što je to i kako se koristi u statistici“, (bez dat.) [Na internetu]. Dostupno na: https://bs.warbletoncouncil.org/prueba-chi-cuadrado-7017#:~:text=Hi-kvadrat%20test%20je%20jedna%20od%20najpoznatijih%20i%20koristi,tome%20jedna%20ne%20ovisi%20o%20drugoj%2C%20niti%20obrnuto [Pristupljeno: 12.07.2022.]

[6] Christoph Zauner, „Implementation and Benchmarking of Perceptual Image Hash Functions“, 2010. [Na internetu]. Dostupno na: https://www.phash.org/docs/pubs/thesis_zauner.pdf [Pristupljeno: 13.07.2022.]

[7] VentureBeat, „What is a perceptual hash function?“, (bez dat.) [Na internetu]. Dostupno na: https://venturebeat.com/ai/what-is-a-perceptual-hash-function/ [Pristupljeno: 16.07.2022.]

[8] Alibaba Cloud, „Google Image Search Explained“, 2017. [Na internetu]. Dostupno na: https://alibaba-cloud.medium.com/google-image-search-explained-30af8ba9cbea [Pristupljeno: 16.07.2022.]

[9] Evan Kilinger, „pHash: The open source perceptual hash libraty“, 2010. [Na internetu]. Dostupno na: http://www.phash.org/ [Pristupljeno: 17.07.2022.]

[10] Raja Halder, Shantanu Pal, Agostino Cortesi, „Watermarking Techniques for Relational Databases: Survey, Classification and Comparison“, 2010. [Na internetu]. Dostupno na: https://www.jucs.org/jucs_16_21/watermarking_techniques_for_relational/jucs_16_21_3164_3190_halder.pdf [Pristupljeno: 17.07.2022.]

[11] Apple Inc., „CSAM Detection: Tehnical Summary“, 2021. [Na internetu]. Dostupno na: https://www.apple.com/child-safety/pdf/CSAM_Detection_Technical_Summary.pdf [Pristupljeno: 17.07.2022.]

[12] Oliver Kuederle, „The Problem With Perceptual Hashes“, (bez dat.) [Na internetu]. Dostupno na: https://rentafounder.com/the-problem-with-perceptual-hashes/ [Pristupljeno: 17.07.2022.]

[13] Lukas Struppek, Dominik Hintersdorf, Daniel Neider, Kristian Kersting, „Learning to Break Deep Perceptual Hashing: The Use Case NeuralHash“, 2022. [Na internetu]. Dostupno na: https://arxiv.org/pdf/2111.06628.pdf [Pristupljeno: 17.07.2022.]

[14] Andrei Z. Broder, „Some applications of Rabin's fingerprinting method“, 1998. [Na internetu]. Dostupno na: https://www.researchgate.net/publication/2688260_Some_applications_of_Rabin's_fingerprinting_method [Pristupljeno: 21.07.2022.]

[15] Sven Taylor, „Browser Fingerprinting Protection: How to Stay Private“, 2022. [Na internetu]. Dostupno na: https://restoreprivacy.com/browser-fingerprinting/ [Pristupljeno: 21.07.2022.]

[16] Shai Halevi, Hugo Krawczyk, „Randomized Hashing and Digital Signatures“, (bez dat.) [Na internetu]. Dostupno na: https://webee.technion.ac.il/~hugo/rhash/ [Pristupljeno: 21.07.2022.]

[17] Saif Al-Kuwari, James H. Davenport, Russell J. Bradford, „Cryptographic Hash Functions: Recent Design Trends and Security Notions“, 2011. [Na internetu]. Dostupno na: https://eprint.iacr.org/2011/565 [Pristupljeno: 22.07.2022.]

[18] Bruce Schneier, „Cryptanalysis of MD5 and SHA: Time for a New Standard“, 2004. [Na internetu]. Dostupno na: https://web.archive.org/web/20160316114109/https:/www.schneier.com/essays/archives/2004/08/cryptanalysis_of_md5.html [Pristupljeno: 22.07.2022.]

[19] Stefan Lucks, „Design Principles for Iterated Hash Functions“, 2004. [Na internetu]. Dostupno na: https://eprint.iacr.org/2004/253 [Pristupljeno: 22.07.2022.]

[20] ScienceDirect, „Cryptographic Hash Algorithm“, (bez dat.). [Na internetu]. Dostupno na: https://www.sciencedirect.com/topics/computer-science/cryptographic-hash-algorithm [Pristupljeno: 22.07.2022.]

[21] Mengjuan Fei, Zhaojie Ju, Xiantong Zhen, Jing Li, „Real-time visual tracking based on improved perceptual hashing“, 2016. [Na internetu]. Dostupno na: https://link.springer.com/article/10.1007/s11042-016-3723-5 [Pristupljeno: 22.07.2022.]

[22] Maamar Hamadouche, Khalil Zebbiche, Mohamed Guerroumi, Hanane Tebbi, Youcef Zafoune, „A comparative study of perceptual hashing algorithms: Application on fingerprint images“, 2021. [Na internetu]. Dostupno na: http://ceur-ws.org/Vol-2904/81.pdf [Pristupljeno: 26.07.2022.]

[23] A. Drmic, M. Silic, G. Delac, K. Vladimir and A. S. Kurdija, „Evaluating robustness of perceptual image hashing algorithms“, 2017. [Na internetu]. Dostupno na: https://ieeexplore.ieee.org/document/7973569 [Pristupljeno: 27.07.2022.]

[24] Ipol, „Image Normalization“, (bez dat.) [Na internetu]. Dostupno na: http://dev.ipol.im/~nmonzon/Normalization.pdf [Pristupljeno: 27.07.2022.]

[25] Hans J. Johnson, Matthew M. McCormick, Luis Iba´nez, *The ITK Software Guide Book 1: Introduction and Development Guidelines and Book 2: Design and Functionality*, 4. izd. ITK Software Guide, 2021.

[26] Shoaib Rashid, „What is image Normalization?“, 2019. [Na internetu]. Dostupno na: https://medium.com/@shoaibrashid/what-is-image-normalization-d8305bf328c0 [Pristupljeno: 27.07.2022.]

[27] PyPI, „ImageHash 4.2.1“, 2021. [Na internetu]. Dostupno na: https://pypi.org/project/ImageHash/ [Pristupljeno: 07.08.2022.]

[28] Adobe, „Adobe Photoshop Express“, 2022. [Na internetu]. Dostupno na: https://www.adobe.com/products/photoshop-express.html [Pristupljeno: 07.08.2022.]

[29] Unsplash, „Nature“, 2022. [Na internetu]. Dostupno na: https://unsplash.com/collections/583204/nature [Pristupljeno: 07.08.2022.]

[30] Drashta Shukla, „Hashing“, 2022. [Na internetu]. Dostupno na: https://thecyberdelta.com/hashing/ [Pristupljeno: 11.07.2022.]

[31] Alex Nadalin, „How to implement a simple hash table in JavaScript“, 2018. [Na internetu]. Dostupno na: https://medium.com/free-code-camp/how-to-implement-a-simple-hash-table-in-javascript-cb3b9c1f2997 [Pristupljeno: 11.07.2022.]

[32] Elliot Williams, „Your Unhashable Fingerprints Secure Nothing“, 2015. [Na internetu]. Dostupno na: https://hackaday.com/wp-content/uploads/2015/11/1024px-avalanche_effect-svg.png [Pristupljeno: 21.07.2022.]

[33] Kate Antonovich, „Man With Open Legs Posture“, (bez dat.) [Na internetu]. Dostupno na: https://www.pngitem.com/middle/hhJobRo_man-with-open-legs-posture-dibujos-con-las/ [Pristupljeno: 21.07.2022.]

[34] Hippopx, „Eiffelov toranj“, 2017. [Na internetu]. Dostupno na: https://www.hippopx.com/hr/paris-france-spring-beauty-the-eiffel-tower-holidays-tree-465712 [Pristupljeno: 21.07.2022.]

[35] Medium, „Hashing and encryption“, 2019. [Na internetu]. Dostupno na: https://fedemcmac.medium.com/hashing-and-encryption-for-dummies-in-case-you-were-wondering-83422b6e7554 [Pristupljeno: 22.07.2022.]

[36] Kimmo Halunen, „Merkle-Damgård hash function construction“, 2018. [Na internetu]. Dostupno na: https://www.researchgate.net/figure/Merkle-Damgard-hash-function-construction_fig1_327000228 [Pristupljeno: 22.07.2022.]

[37] Google Play, „Adobe Photoshop Express“, (bez dat.) [Na internetu]. Dostuno na: https://play-lh.googleusercontent.com/r9zF77jorOmkaRlXnvsLiuVQ3p_gYW8y7x_UL-COoH9PxaTUEMbW1wiwS0z1n1Q31Q=w240-h480-rw [Pristupljeno: 07.08.2022.]

Popis slika

Popis tablica