SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 220

AUTOMATSKO BOJANJE CRNO-BIJELIH FOTOGRAFIJA PUTEM DUBOKOG UČENJA

Filip Anđel

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 220

AUTOMATSKO BOJANJE CRNO-BIJELIH FOTOGRAFIJA PUTEM DUBOKOG UČENJA

Filip Anđel

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 10. ožujka 2023.

DIPLOMSKI ZADATAK br. 220

Pristupnik:	**Filip Anđel (0036507792)**
Studij:	Računarstvo
Profil:	Računarska znanost
Mentor:	izv. prof. dr. sc. Vladimir Čeperić

Zadatak: **Automatsko bojanje crno-bijelih fotografija putem dubokog učenja**

Opis zadatka:

U ovom radu opisuje se problem automatskog bojanja crno-bijelih fotografija te se daje pregled tehnika dubokog učenja koje se koriste za rješavanje tog problema. Predstavljaju se dosadašnja istraživanja na tu temu te se odabire prikladan model dubokog učenja za implementaciju. Opisuje se postupak pripreme skupa podataka za učenje, validaciju i ispitivanje te se analizira utjecaj veličine mreže, različitih metoda regularizacije, aktivacijskih funkcija te tehnika učenja. Postupak učenja i validiranja hiperparametara se optimizira, a valjanost modela se testira minimalno na setu crno-bijelih fotografija slika Gustava Klimta uz uobičajene mjere kvalitete. Dobiveni rezultati se interpretiraju te se ukazuje na moguće probleme, nedostatke i ograničenja. Na kraju, predlažu se moguće buduće nadogradnje modela, te se razvija programska rješenja za automatsko bojanje crno-bijelih fotografija. U radu se prilažu izvorni kodovi programa, dobiveni rezultati uz potrebna pojašnjenja i korištena literatura.

Rok za predaju rada: 23. lipnja 2023.

Sadržaj

Uvod

Razvoj dubokog učenja, a posebno područja računalnog vida, otvorio je mnoga vrata za spoj umjetnosti i tehnologije u svrhu očuvanja i revitalizacije kulturne baštine. Ovaj rad bavi se istraživanjem područja kolorizacije slika korištenjem tehnika dubokog učenja, fokusirajući se na umjetnička djela poznatog slikara Gustava Klimta. Cilj je pronaći prikladan model kolorizacije i naučiti ga da u crno-bijele fotografije Klimtovih remek-djela unese živopisne nijanse čime će se obogatiti iskustvo promatrača i pridonijeti očuvanju umjetničke baštine.

Klimtova umjetnost, poznata po svojim složenim uzorcima, simboličkim motivima i prepoznatljivoj paleti boja, predstavlja jedinstven izazov za kolorizaciju. Nadalje, rijetkost dostupnih podataka, otežana gubitkom mnogih Klimtovih originalnih umjetničkih djela, naglašava važnost korištenja računalnih metoda za rekonstrukciju i reinterpretaciju tih povijesnih blaga.

Kroz sveobuhvatan pregled postojeće literature istražit će se evolucija tehnika kolorizacije počevši od ručnih pristupa pa sve do suvremenih metoda dubokog učenja. Ističu se prednosti, mane, ograničenja i potencijal tradicionalnih i modernih metoda, sa posebnim naglaskom na modelu transformera koji predstavlja vrhunac napretka u području računalnog vida. Konačno, predstavit će se odabrani transformerski model i rezultati dobiveni učenjem istoga na kolekciji djela Gustava Klimta, kao i mogućnosti za unapređenje.

1. Kolorizacija slika

Unošenje boje u crno-bijele slike izazovan je proces koji je nastao u prvoj polovici 19. stoljeća zajedno s fotografijom [1]. Sve su fotografije bile monokromatske (sadržale su samo nijanse sive boje) čak do 1950-ih, pa su tadašnji fotografi i umjetnici ručno bojali svoje monokromatske fotografije kako bi izgledale realističnije. Jedan od prvih umjetnika koji je primijenio ovakav proces je švicarski slikar i grafičar Johann Baptist Isenring koji je koristio mješavinu arapske gume i pigmenata za bojanje crno-bijelih fotografija. Ovaj je proces podrazumijevao ručno bojanje fotografije tako da izgleda što sličnije onome kako bi izgledala da je slikana fotografskom tehnikom koja hvata boje. Drugim riječima, cilj je bio obojati sliku što realističnije. To nije jednostavan zadatak, jer se umjetnik mora sjećati kako je prizor izgledao u trenutku fotografiranja, a u slučaju da se ne sjeća ili boja sliku koju nije sam uslikao, mora pretpostaviti kako bi fotografija mogla biti obojana. Osim toga, sam čin bojanja ovakvih fotografija nije bio jednostavan i zahtijevao je dobro poznavanje crtanja i slikanja.

Još jedna primjena kolorizacije crno-bijelih slika je restauracija umjetničkih djela. Umjetnička djela, ovisno o korištenoj tehnici slikanja, mogu s vremenom izgubiti boju i izblijediti. Osim toga, umjetnici ponekad žele retuširati ili dotjerati postojeće boje na nekoj slici, ili dodati nedostajući dio slike u slučaju fizičkih oštećenja. Iako takav pothvat isto zahtijeva vještinu slikanja i poznavanje originala, relativno je jednostavan jer umjetnik ima ideju kako je slika prije izgledala.

Nešto zahtjevniji aspekt kolorizacije slika je bojanje bez ikakvog predznanja o slici. Najčešća primjena je ponovo na starim crno-bijelim fotografijama, međutim u ovom slučaju kolorizaciju radimo godinama ili čak stoljećima kasnije, tako da su sve potencijalne informacije o bojama koje bi originalna fotografija trebala sadržavati davno izgubljene. Često se koloriziraju fotografije koje imaju neki povijesni ili umjetnički značaj kako bi se približile suvremenoj publici i modernizirale.

Slika 1. Primjer monokromatske i ručno obojane fotografije u autorstvu studija Stillfried & Andersen, druga polovica 19. stoljeća

Iako autor tijekom koloriziranja fotografija ne zna kako je scena stvarno izgledala u trenutku fotografiranja, može s dosta dobrom preciznosti pretpostaviti boje na temelju konteksta i zdravog razuma. Primjerice, na fotografijama nebo nikada neće biti zelene boje, niti će travnjak biti plav. Ako je na fotografiji neka poznata osoba, boja kose i boja očiju te osobe može se saznati iz povijesnih izvora. Ipak, može se dogoditi da je nemoguće iz konteksta zaključiti kako obojati neki dio slike. To je ustvari temeljni problem kolorizacije – kada fotografiju slikamo samo u nijansama sive, u potpunosti gubimo sve informacije o originalnim bojama. Jedina informacija koja ostaje je intenzitet sive u pojedinoj točci, odnosno koliko je neka točka (piksel) tamna ili svijetla.

Ovaj problem najviše dolazi do izražaja kod bojanja umjetničkih djela koja su originalno bila u boji. Ovisno o stilu, slike može biti vrlo lako ili gotovo nemoguće precizno obojati. Primjerice, koloriziranje slika u stilu realizma vrlo je slično bojanju fotografija, dok je bojanje apstraktnijih stilova subjektivno i otvoreno interpretaciji. Primjerice, kod nekih autora nebo može biti zlatno, stabla ne moraju biti zelena i slično. Poznavanje stila pojedinog autora čiju se sliku kolorizira može biti od velike pomoći u ovom procesu, kao i perioda i umjetničkog pokreta koji se veže uz pojedinu sliku. Upravo iz tog razloga, restauraciju i kolorizaciju crno-bijelih slika uglavnom su ručno radili obrazovani umjetnici s dubokim poznavanjem originalnog autora.

S razvojem tehnologije kolorizacija slika postala je znatno pristupačnija. Digitalne tehnike kolorizacije znatno olakšavaju sam proces bojanja, iako je problem odabira boja i

nijansi i dalje prisutan. Najpoznatiji primjer digitalnog uređivanja, stvaranja i bojanja dvodimenzionalnih slika je program Photoshop, koji omogućava praktički bilo kome da se okuša u kolorizaciji slika bez prethodnog poznavanja tehnika crtanja i slikanja. Iako je kolorizacija ovim putem postala jednostavnija, i dalje zahtijeva doprinos osobe koja mora imati određenu razinu vještine i predznanja.

2. Automatska kolorizacija

Iako ručna kolorizacija funkcionira sasvim dobro, bojanje velikog broja slika ili fotografija je vrlo vremenski zahtjevan proces. Osim toga, ručna kolorizacija (čak i uz alate za digitalnu obradu slika) zahtijeva dobro poznavanje slika ili fotografija koje se bojaju, kao i određenu vještinu. Počevši u ranim 2000-ima, kreću se razvijati automatske metode kolorizacije slika čiji je cilj ubrzati i pojednostaviti proces, a u konačnici i potpuno ga automatizirati. Kroz slijedeća poglavlja predstavit će se neki od modela automatske kolorizacije. Iako se uz riječ „model“ obično veže arhitektura neuronske mreže u području dubokog učenja, zbog jednostavnosti će se i metode koje ne uključuju duboko učenje a bave se automatskom kolorizacijom nazivati modelima, na temelju činjenice da primaju neku ulaznu sliku, a na izlazu daju pretpostavku o tome kako bi original trebao izgledati u boji.

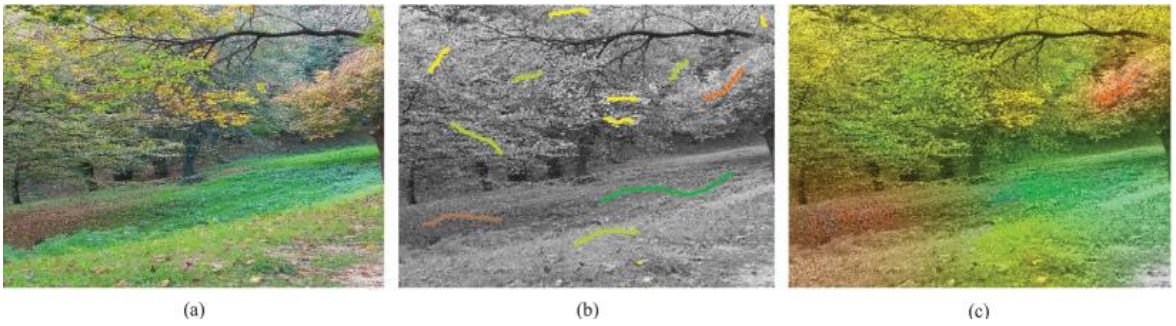
2.1. Problem kolorizacije

Problem kolorizacije je što je često nemoguće iz crno-bijele slike predvidjeti koje je točno boje neki objekt na slici. Na primjeru (slika 2) prikazani su automobili u raznim bojama, a pored ista fotografija ali crno-bijela. Vidljivo je da je praktički nemoguće iz crno-bijele slike razlučiti koje bi boje automobili stvarno trebali biti. Jedina informacija koja ostaje sadržana u slici je intenzitet boje, odnosno koji je automobil svjetliji a koji tamniji. Ta informacija nije dovoljna za donijeti zaključak o stvarnoj boji automobila, što čini bilo kakav pokušaj bojanja crno-bijele slike bez znanja o originalu samo pretpostavkom. Drugim riječima, rješenje problema kolorizacije nije jedinstveno niti jednoznačno, pogotovo u slučaju kolorizacije slike koja niti nema original u boji. Ovaj problem otežava ručno bojanje slika ali i predstavlja glavnu prepreku automatizaciji ovog postupka. Glavne modele automatske kolorizacije slika može se podijeliti po tome kako rješavaju ovaj problem.

Slika 2. a) Fotografija nekoliko automobila preuzeta s Interneta b) ista fotografija bez boja

2.2. Modeli navođeni dodatnim kontekstom

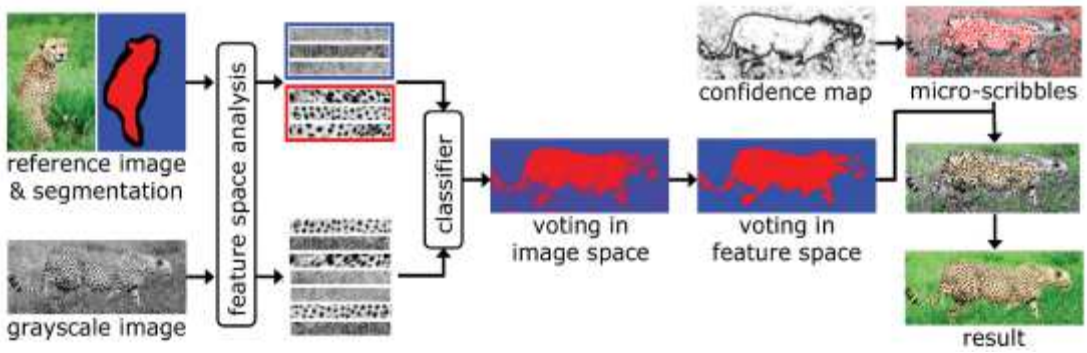
Jedna od ideja kako riješiti problem nedostajućih informacija je jednostavno na neki način unijeti te informacije nazad u sliku. Primjerice, to može biti ručno unošenje željene boje na specifične dijelove crno-bijele slike. U radu [3] predstavljena je ovakva metoda. Temelji se na pretpostavci da su regije piksela sličnog intenziteta (slične nijanse sive) obojani sličnom bojom. Ako korisnik ručno predloži koje boje koristiti za koju regiju, model može odraditi ostatak posla kolorizacije i dati solidan rezultat. Konačna se boja slike računa kao linearna funkcija intenziteta Y u YUV prostoru boja. Razni drugi modeli razvijeni su na sličnom principu, a primjer ovakve kolorizacije vidljiv je na slici.



Slika 3. a) originalna slika b) prijedlozi boja c) konačni rezultat

Prednosti ovakvog pristupa su jednostavnost i funkcionalnost bez potrebe za originalnom slikom, te mogućnost proizvoljnog izbora boje i namještanja boja sve dok korisnik nije zadovoljan konačnim rezultatom. Najveći nedostatak je što za svaku sliku korisnik treba pojedinačno predložiti boje, što je neprimjenjivo na veliki skup slika. Boje trebaju biti pomno izabrane kako bi rezultat bio dobar, a konačne slike bez obzira na izbor boja mogu izgledati nerealistično (primjer vidljiv na slici).

Slijedeća metoda kolorizira sliku na temelju referentne slike, što znatno smanjuje ljudski faktor u postupku. Za bojanje neke crno-bijele slike, potrebno je prvo pronaći što sličniju sliku. Ovaj zadatak znatno je olakšan poboljšanjem i razvojem internetskih tražilica, pomoću kojih je lako pronaći slike sličnog sadržaja, boja i tekstura. Model zatim koristi sličnosti u teksturama i intenzitetima piksela između referentne slike i crno-bijele slike kako bi obojao crno-bijelu sliku po uzoru na referentnu. Primjer ovakve metode koja koristi strojno učenje za prepoznavanje značajki prikazan je na slici. Model koristi algoritam k-najbližih susjeda (k-NN) kako bi grupirao slične piksele u regije te zatim slične regije na crno-bijeloj slici obojao kao što su obojane na referentnoj ([4]).



Slika 4. Princip rada kolorizacije na temelju primjera

Ovakav pristup brz je i jednostavan, ali također ima svojih nedostataka. Referentne slike treba ručno pronaći, a moguće je da za neku vrlo specifičnu sliku i ne postoji dovoljno dobra referentna slika. Osim toga, referentna slika mora biti vrlo slična onoj koju treba obojati kako bi rezultati bili zadovoljavajući.

2.3. Modeli dubokog učenja

Pristup modelima dubokog učenja u svrhu kolorizacije crno-bijelih slika relativno je jednostavan i sličan ostalim problemima iz područja računalnog vida. Općenito, postupak je sljedeći: potrebno je izgraditi model koji je sposoban prepoznati značajke i objekte na slici, odnosno koji ima ugrađeno semantičko razumijevanje, te koji prepoznate značajke može asocirati s određenim bojama. Takav model se trenira (uči) na velikom skupu slika na način da na ulaz prima crno-bijele slike, a na izlazu daje svoju pretpostavku kako bi ciljna obojana slika trebala izgledati. Rezultat koji je model dao uspoređuje se s pravom ciljnom slikom, te se računa gubitak (eng. *loss*) pomoću neke odabrane funkcije. Funkcija gubitka modelu

govori koliko treba promijeniti svoje unutarnje parametre da bi izlazna slika više ličila originalnoj ciljnoj slici. Nakon dovoljno viđenih primjera, model bi trebao dobro naučiti bojati i slike koje nije do tada vidio.

U ovom pristupu, bojanje slike obavlja se na temelju prepoznanih objekata i značajki. Dobro naučen model „znati“ će prepoznati automobil, čovjeka, nebo, stablo… i obojati ih u odgovarajuće boje. Cilj korištenja dubokog učenja u svrhu kolorizacije (i općenito) je automatizirati proces što je više moguće i izbaciti potrebu za ljudskim doprinosom kako bi proces postao skalabilan, brz i neovisan o ljudskom faktoru. U suštini automatizacija je postignuta tako da model, umjesto da dobiva dodatan kontekst o slici iz ručno predloženih boja ili iz referentne slike, informaciju o tome kako obojati određeni dio slike izvlači iz kolekcije već viđenih primjera na kojima je naučen. Naravno, to znači da ni modeli dubokog učenja ne daju jednoznačno i potpuno točno rješenje za danu sliku. Na primjeru slike 2, ovisno o tome koje je slike model vidio tijekom učenja, različito će obojati automobile. Primjerice ako je u skupu za učenje 100 slika koje sadrže automobil marke Chevrolet (na slici plavi automobil) i svaki od tih automobila je crvene boje, model će gotovo sigurno odlučiti da je i ovaj Chevrolet prikazan na slici 2 crvene boje iako to nije točno.

Prvi rad koji nastoji kolorizaciju prenijeti u svijet dubokog učenja je [5] koji koristi jednostavnu unaprijednu neuronsku mrežu kao funkciju aproksimacije obojane slike iz skupa značajki. Nakon toga, modeli većinom koriste arhitekturu konvolucijske neuronske mreže (CNN). Ova se arhitektura pokazala kao optimalna za većinu problema iz područja računalnog vida zbog svojih prednosti, a to su manji broj parametara (brže učenje), manja sklonost prenaučenosti i prostorna ekvivarijantnost.

CNN modeli primijenjeni na problem kolorizacije mogu se podijeliti na samostalne i one koji zahtijevaju neki doprinos korisnika. Primjerice to može biti kratki tekst koji opisuje boje nekih dijelova slike (žuti auto, zelena košulja…) ili primjer boje zadan na nekoj regiji slike. Primjer ovakve arhitekture predstavljen je u radu [6] a model se sastoji od konvolucijske neuronske mreže koja prima natuknice od strane korisnika i na temelju njih boja slike.

3. Transformeri

Konvolucijske neuronske mreže vladale su područjem računalnog vida kao optimalan model za većinu zadataka sve do pojave vizualnih transformera. Transformer je vrsta arhitekture dubokog učenja temeljena na konceptu samo-pozornosti i enkodersko-dekoderskoj strukturi, a prvi put je predstavljen u radu „Attention Is All You Need“ ([7]). Ova je arhitektura osmišljena kao model optimiziran za rad sa dugim sekvencama podataka, pa je prva primjena bila u području obrade prirodnog jezika (NLP – eng. *natural language processing*), točnije prevođenja niza riječi iz jednog jezika u drugi. U tom su se području transformeri brzo pokazali boljima nego povratne neuronske mreže i njihove varijante kao što je LSTM (eng. *long short-term memory*) zbog mogućnosti transformera da cijelu ulaznu sekvencu obrađuje istovremeno i paralelno neovisno o duljini. Kroz slijedećih nekoliko poglavlja detaljno će se objasniti princip rada i arhitektura transformera predstavljenog u spomenutom radu.

3.1. Arhitektura

Transformer, kako je definiran u prethodno spomenutom radu „Attention Is All You Need“, sastoji se od enkodera i dekodera. Arhitektura modela prikazana je na slici 5 preuzetoj iz spomenutog rada. Većina elemenata od kojih se sastoje enkoder i dekoder nisu ništa do sad neviđeno u dubokom učenju (*feed-forward* neuronska mreža, linearni sloj, *softmax* sloj itd.), a jedino što je novo su elementi *multi-head attention* i *masked multi-head attention*. Ovi će elementi kao i ostatak arhitekture biti objašnjeni u slijedećim poglavljima, a za početak potrebno je razumjeti što model prima kao ulaz.

Na ulazu u model nalaze se riječi odnosno tokeni (nizovi znakova). S obzirom da nije moguće matematički manipulirati nizovima znakova, potrebno je tokene pretvoriti u vektorske reprezentacije. Svaka se riječ pretvara u reprezentacijski vektor (eng. *embedding*) na način da su riječi koje se više pojavljuju u istim kontekstima sličniji vektori (odnosno bliži po euklidskoj udaljenosti) nego riječi koje su nepovezane. Ovaj postupak je već postojeća praksa koja se koristi u području NLP-a. Ovime model dobiva informaciju koje riječi su asocirane i koliko jako, odnosno dobiva svojevrsno semantičko razumijevanje riječi.

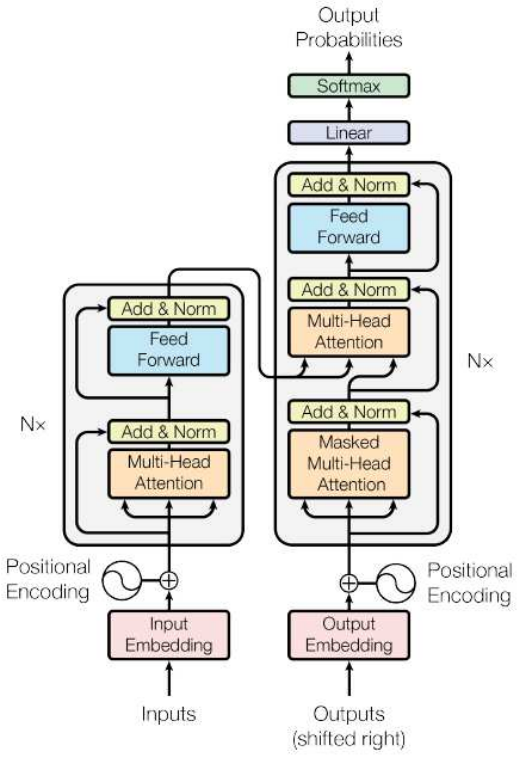
Ono što nedostaje je informacija o tome gdje se koja riječ nalazi u rečenici. Iako to ponekad ne mijenja značenje rečenice, u nekim slučajevima je dosta bitno, a pogotovo za

jezike gdje je redoslijed riječi u rečenici relativno fiksan. Iz tog razloga na *embeddinge* tokena dodaje se još i pozicijski vektor kako bi se ugradila informacija o poziciji tokena u nizu. Ovaj se vektor može računati na razne načine, a autori rada koristili su slijedeće sinusoidne funkcije:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right)$$

U ovim funkcijama *pos* označava poziciju tokena, a *i* označava dimenziju. Pozicijski vektori jednake su veličine kao i reprezentacijski tako da se lako mogu zbrojiti. Na taj način na ulazu modela dobiva se matrica koja sadrži informacije o semantičkoj sličnosti riječi kao i njihovoj poziciji u rečenici. Ova matrica zatim ulazi u enkoder, odnosno u *multi-head attention* blok.



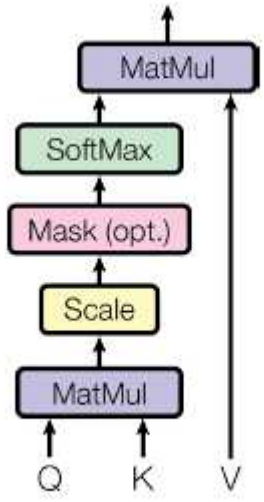
Slika 5. Arhitektura transformera

3.2. Pozornost

Pozornost (eng. *attention*) je mehanizam prethodno ugrađivan u povratne neuronske mreže. Predstavljen je u radu [8] a osmišljen je s ciljem da riješi probleme povratnih neuronskih mreža (RNN – eng. *recurrent neural network*). Problemi RNN-a u kontekstu NLP-a koje pozornost popravlja su slijedeći ([9]):

- dugačke sekvence – ako se na ulaz ovakve mreže stavi dugačak niz riječi, može se dogoditi da model „zaboravi“ na nešto s početka teksta jer ima ograničeno pamćenje
- veliki broj slojeva koji vodi velikom broju parametara, što znači dugotrajno učenje i optimiziranje u puno koraka
- nestajući i eksplodirajući gradijenti kao posljedica prethodnog problema
- RNN mreže ne mogu se paralelizirati jer se učenje vodi korak po korak

Mehanizam pozornosti inspiriran je biološkim procesom razumijevanja jezika, odnosno time kako ljudski mozak interpretira rečenice i razlučuje kako su riječi u rečenici povezane i koje su riječi više ili manje relevantne za semantičko značenje rečenice. Na ovaj način postignuto je neko općenito razumijevanje koje su riječi povezane, a zatim je potrebno povezati pojedine riječi u specifičnom ulazu koji je dan modelu. Pozornost to radi na način da međusobno pomnoži vektor svakog tokena sa vektorima svih ostalih tokena, te tako dobije matricu koja govori u kojoj je mjeri neki token povezan s ostalim tokenima u nizu.



Slika 6. Mehanizam pozornosti - skalirani skalarni produkt vektora

Matrica pozornosti računa se slijedećom formulom, čija je skica prikazana na slici 6:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

U slučaju samo-pozornosti, *Q*, *K* i *V* iz gornje formule su ili matrice jednake ulaznoj. Rezultantna matrica pozornosti jednake je veličine kao i ulazni *embedding*. Kod problema prevođenja teksta, ulazni *embedding* je matrica gdje je u svakom redu *embedding* jedne riječi, pa matrica ima *N* redaka i *d* stupaca. *N* označava veličinu ulazne sekvence a *d* (ili d_{model}) veličinu *embeddinga*. Matrica pozornosti sada sadrži još jednu informaciju, točnije o povezanosti specifičnih tokena u nizu, a zadržava informacije o semantičkom značenju riječi i njihovoj poziciji koje nasljeđuje od *embeddinga*. Ta nova informacija proizlazi upravo iz skalarnog produkta matrica *Q* i *K* koji po svojoj prirodi daje veći rezultat što su sličniji vektori koje množi. Produkt se skalira s korijenom veličine *embeddinga* kako ne bi bili preveliki, a zatim se nad skaliranim produktom vrši funkcija *softmax* koja pretvara produkt u težine jer će suma svakog retka biti 1. Konačno, dobivene težine se množe s matricom *V*.

3.3. *Multi-head attention*

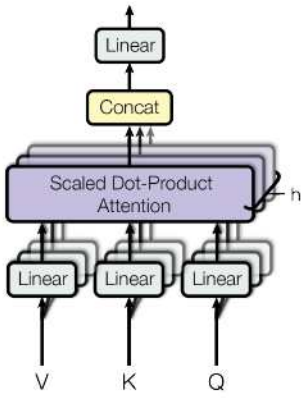
U modelu transformera koristi se malo složeniji oblik samo-pozornosti koji nadograđuje prethodno opisani mehanizam. Formula je slijedeća, a skica je na slici 7:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O$$

$$where\ head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

Matrice *Q, K, V* sada više nisu jednake kao ulaz u model, nego se na ulazu vrši linearna transformacija koja će biti trenirana. Takvih linearnih transformacija ima *h* za svaku od tri ulazne matrice, a zatim se na njima vrši funkcija pozornosti definirana ranije. Hiperparametar *h* označava broj glava (*head*) po čemu ovaj blok dobiva svoje ime. Rezultatne vrijednosti pozornosti zatim se spajaju nazad u rezultantnu matricu iste veličine kao i prije (*N* puta *d*) i množe s još jednom matricom težina koje se također treniraju. Na ovaj način u mehanizam pozornosti uvedena je mogućnost učenja, a cilj proširenja mehanizma na više glava je dozvoliti modelu da nauči više različitih interpretacija riječi. Primjerice, neke riječi mogu poprimiti oblik glagola, imenice ili pridjeva ovisno o kontekstu

rečenice, a različite će glave u ovakvom modelu naučiti interpretirati više mogućih značenja za istu riječ.



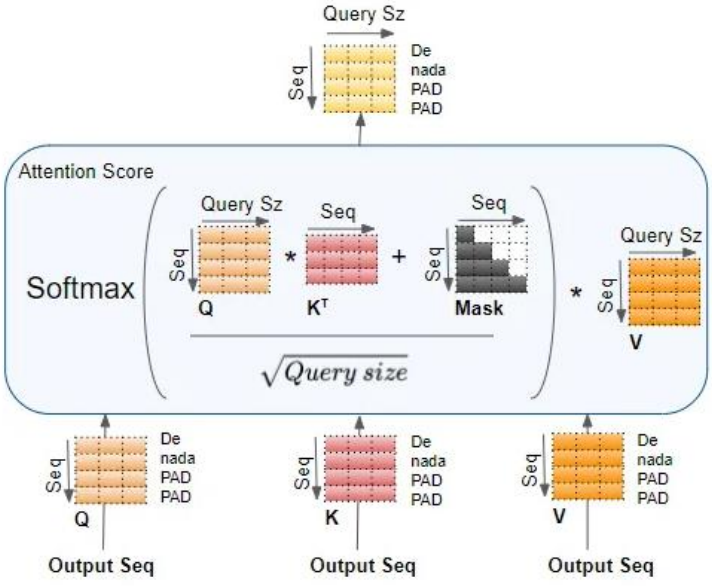
Slika 7. Skica *multi-head attention* bloka

3.4. Enkoder

Do sada je objašnjeno što se događa s ulaznim nizom tokena prije ulaza u sami enkoder i u *multi-head attention* bloku. Izlaz iz tog bloka je matrica jednake veličine kao ulazna, ali u nju je pomoći pozornosti ugrađeno (enkodirano) još informacija o povezanosti tokena. Prema slici 5, vidljivo je da izlaz ulazi u normalizacijski sloj, a zatim u običnu potpuno povezanu (*feed forward*) neuronsku mrežu. Ova se mreža ponekad označava i kao MLP (*multi-layer perceptron*) ili *fully connected* (potpuno povezana) mreža. Sastoji se od dvije linearne transformacije i nelinearne aktivacijske funkcije (obično ReLU) između njih. Svrha ovih transformacija je dodati još parametara koji se mogu učiti te uvesti nelinearnost i dodatnu složenost u model. Samo-pozornost uči veze između tokena u ulaznom nizu, a ovaj sloj dodaje još fleksibilnosti kako bi model mogao prepoznati složenije uzorke i reprezentacije iz dobivenog ulaza. Smatra se da je u ovoj mreži „zapamćeno“ najviše informacija te se zato povećanje kapaciteta transformerskih modela obično vrši povećanjem upravo ovog sloja. ([10]). Nakon još jednog sloja normalizacije, izlaz potpuno povezane mreže šalje se u dekoder.

3.5. Dekoder

Dekoder na ulazu prima *embedding* izlaznih tokena (prevedena rečenica) s dodanim pozicijskim kodiranjem kao i enkoder. Također, ovi vektori/matrice ulaze u blok vrlo sličan *multi-head attention* bloku iz enkodera, osim što su neki tokeni „maskirani“ odnosno skriveni. Točnije, maskirani su tokeni koje dekoder ne smije još vidjeti. Ideja je da, s obzirom da bi transformer trebao prevoditi riječ po riječ iz ulaznog niza u riječi izlaznog niza, dekoder ne smije znati pozornost između prve riječi i zadnje sve dok nije stvarno došao do zadnje riječi. Ilustracija je na slici 8, a prikazana je maskirajuća matrica koja skriva dio matrice pozornosti na način da pretvara te vrijednosti u -∞ kako bi ih *softmax* funkcija pretvorila u nule. U ilustraciji je dan primjer izlaznog niza „De nada PAD PAD“ što je prijevod ulaznog niza „You are welcome PAD“. Tokeni „PAD“ označavaju *padding* odnosno nadopunjavanje koje služi da bi ulazni i izlazni niz bili fiksne duljine, a dodaju se u slučaju da se prevode rečenice manje od te duljine. *Padding* tokeni također dobivaju svoj *embedding* (obično nule) pa ih model nauči ignorirati. Maskirajuća matrica skriva sve iznad glavne dijagonale matrice pozornosti, što ostvaruje sljedeći učinak: za token „De“ (prvi redak skalarnog produkta matrica Q i K), pozornost se računa samo za token „De“ jer je to jedini do tada viđeni token, a ostali se produkti postavljaju u $-\infty$. U drugom redu, pozornost se računa između tokena „nada“ i tokena „De“, te tokena „nada“ i samog sebe, a ostatak niza je skriven.

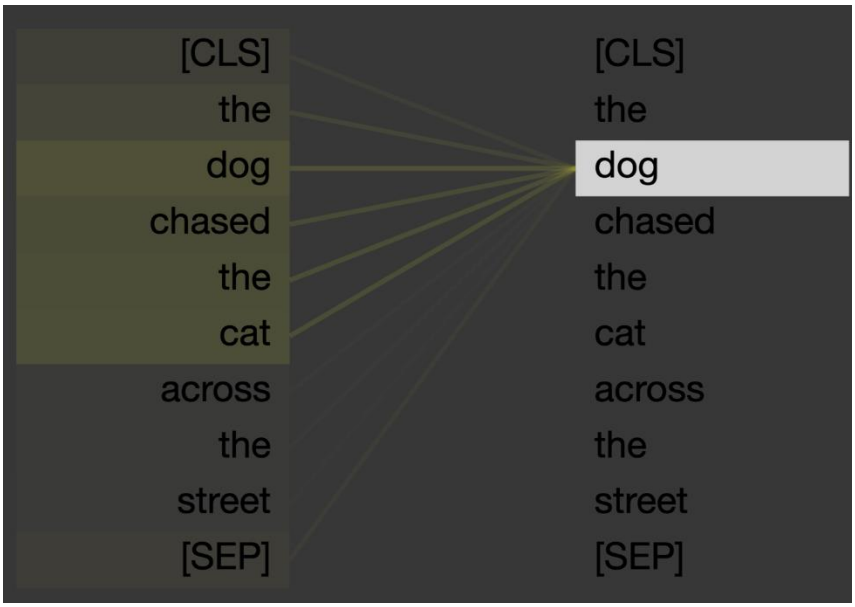


Slika 8. *Masked multi-head attention* blok [11]

3.6. Enkoder-dekoder pozornost (*cross-attention*)

Izlaz maskirane pozornosti prolazi sloj normalizacije, a zatim ulazi u još jedan *multi-head attention* blok koji se često naziva *encoder-decoder attention* ili *cross-attention*. Ovaj blok funkcionira kao i onaj u enkoderu s jednom razlikom – matrice *K* i *V* dolaze iz enkodera, a *Q* dolazi iz izlaznog niza provedenog kroz maskiranu pozornost. U ovom elementu modela konačno dolazi do veze između ulaznog i izlaznog niza, i upravo je on zaslužan za rješavanje prevođenja rečenica. Računanjem pozornosti u ovom bloku dobiva se matrica koja prolazi normalizaciju i ponovo potpuno povezanu mrežu, a konačno prolazi još jednu linearnu transformaciju i *softmax* funkciju kako bi dala prijedlog sljedeće prevedene riječi. Kako točno funkcionira izlaz dekodera bit će objašnjeno zajedno s treniranjem i predviđanjem.

Prije toga, za intuitivno razumijevanje kako dolazi do prevođenja u ovom bloku bilo bi dobro spomenuti zašto se ulazi u funkciju pozornosti označavaju baš sa *Q, K i V*. Ovo imenovanje potiče iz baza podataka, gdje se glavni elementi zovu upit (*query*), ključ (*key*) i vrijednost (*value*). Primjerice, ako se u nekoj bazi podataka obavlja neko pretraživanje, unešeni tekst koji se traži je *query*, skup svih unosa u bazi čini ključeve (*keys*) a rezultat pretraživanja trebali bi biti unosi u bazi najsličniji upitu (*values*). Analogno vrijedi za ove pojmove u transformerskoj arhitekturi, a odličan je primjer opisan u izvoru [12]. Primjer je dan na rečenici „the dog chased the cat across the street“. Ako je cilj prevesti specifično riječ „dog“, ta riječ postaje upit (*Q*) koji želi saznati koje druge riječi u rečenici su mu bitne. Skup svih ostalih riječi u rečenici čini ključeve (*K*), a te riječi na upit odgovaraju sa svojom važnosti za riječ „dog“. Ovo proizlazi iz prethodno opisanog skalarnog produkta matrica *Q* i *K*. Pretpostavka je da će riječi “chased“ i „cat“ biti više povezane s riječi „dog“ (skalarni produkt će biti veći jer se te riječi češće pojavljuju u sličnim kontekstima) nego riječi „across“ i „street“, odnosno biti će mu od veće važnosti. Matrica *V* predstavlja općenite semantičke informacije o svim ovim riječi neovisno o ulaznoj rečenici. Konačno, pozornost (skalarni produkt *Q* i *K*) provodi se kroz *softmax* i množi s vrijednostima (*V*) kako bi se dobio ponderirani zbroj (eng. *weighted sum*) *embeddinga* temeljen na relevantnosti. Vizualizacija ovih vrijednosti prikaza je na slici 9.



Slika 9. Vizualizacija pozornosti za riječ "dog". Svjetlija žuta boja označava veći rezultat

3.7. Predviđanje

Nakon prethode analogije i primjera lakše je intuitivno razumjeti kako model prevodi riječi iz ulaznog niza u izlazni. Prevođenje se izvršava u više vremenskih koraka, a ne odjednom, za razliku od treniranja. Kako bi model odradio predviđanje (eng. *inference*), potrebno je na ulaz enkodera postaviti rečenicu koju model treba prevesti. Enkoder u svoju izlaznu matricu ugrađuje informacije o semantičkom značenju, povezanosti i važnosti riječi u ulaznoj rečenici, kao i o njihovim pozicijama. Na ulaz dekodera potrebno je postaviti token koji označava početak rečenice, često označavan kao „<SOS>“ (eng. *start of sentence*). Ovaj token, slično kao i *padding* token, ima svoj predodređen i fiksan *embedding*. Kada na enkoder-dekoder blok dođe <SOS> token, on kombinira semantičko značenje tog tokena s kontekstom ulaznog niza te izbacuje matricu pozornosti. Nakon sloja normalizacije, ta matrica ulazi u potpuno povezanu mrežu kojoj je zadatak interpretirati matricu pozornosti i dati sljedeću riječ u nizu, a u ovom slučaju to je prva riječ prevedene rečenice. Nakon toga u sljedećem vremenskom koraku na ulaz dekoder postavlja se niz od dva tokena: <SOS> i riječ koju je model odabrao u prošlom vremenskom koraku. Postupak traje sve dok dekoder ne previdi token <EOS> (eng. *end of sentence*) koji označava da je izlazna rečenica gotova. Kroz sve te vremenske korake, ulaz i izlaz enkodera se ne mijenjaju jer se ulazni niz nije promijenio, pa tako matrice K i V na ulazu enkoder-dekoder bloka ostaju konstantne, a mijenja se upit Q.

3.8. Treniranje

Treniranje transformerskog modela vrši se u jednom vremenskom koraku (za svaki primjer), što je i bio jedan od razloga uvođenja pozornosti kao temeljne arhitekture umjesto RNN-a. Na ulaz enkodera dolazi ulazna rečenica kako je opisano u prethodnim poglavljima, a na izlaz (ulaz dekodera) dolazi ciljna prevedena rečenica. *Masked multi-head attention* blok pobrine se za to da se pozornost ne računa između tokena koji dolaze nakon trenutnog. Funkcija gubitka računa se nad vrijednostima *embeddinga* izlaza modela i stvarnog prijevoda (eng. *label* ili *target*) i unazadnom propagacijom računaju se gradijenti i ažuriraju parametri kao i kod bilo kojeg drugog modela dubokog učenja.

Predviđanje i učenje transformerskih modela objašnjeni su na temelju izvora [13].

4. Vizualni transformer (*vision transformer*)

Ubrzo nakon pojavljivanja transformera u području obrade prirodnog jezika, taj je model primijenjen i na problem prepoznavanja slika. U radu [14] prvi put je iskorištena arhitektura transformera u području računalnog vida i to sa vrlo dobrim rezultatima. Autori rada nastojali su što manje modificirati originalni transformer iz rada „Attention Is All You Need" pa su praktički jedino promijenili ulazne tokene iz riječi u djeliće slike veličine 16x16 piksela. Ovakav transformer nazvali su ViT (Vision Transformer) i naučili su ga na ImageNet skupu podataka (skup od 1.3 milijuna raznovrsnih slika). Ovako naučen model davao je nešto lošije rezultate od tadašnjeg najboljeg modela ResNet koji se temeljio na CNN arhitekturi. Autori rada zaključili su da inherentna induktivna pristranost konvolucijskih mreža, točnije prostorna i translacijska ekvivarijantnost, omogućava takvim modelima učenje na manjim skupovima podataka s dobrim rezultatima. Međutim, učenjem transformera na još većem skupu podataka od nekoliko stotina milijuna slika, pokazalo se da dovoljno velik skup podataka znači više nego induktivna pristranost CNN-a. Konačni rezultati i usporedba s ResNet konvolucijskom mrežom prikazani su na slici 8.

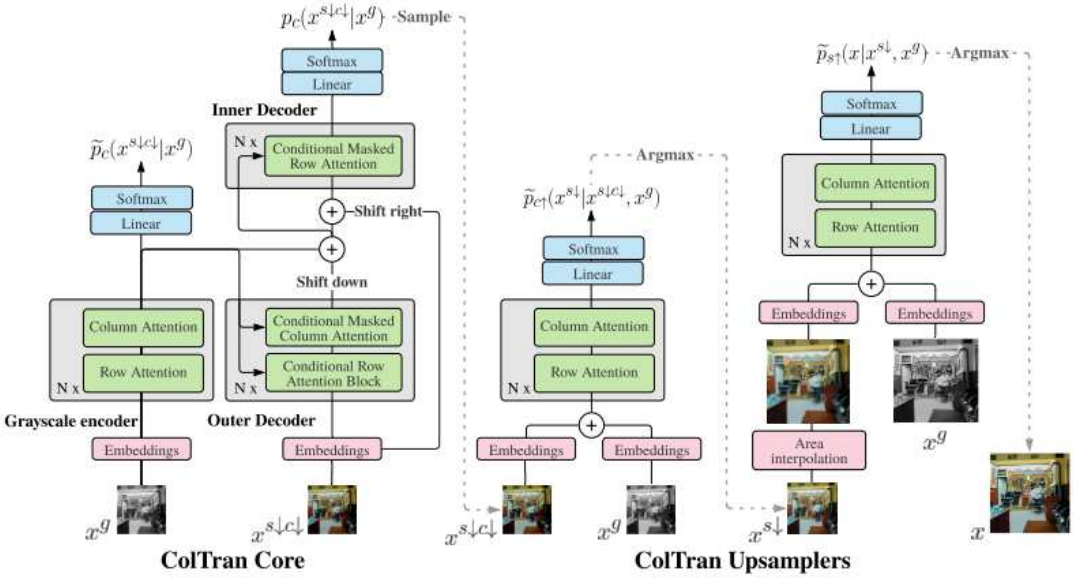
	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Slika 10. Tablica rezultata ViT i ResNet modela preuzeta iz rada

Pokazalo se da je mehanizam samo-pozornosti dovoljno snažan da zamijeni čak i konvolucijske neuronske mreže u području računalnog vida na problemu prepoznavanja slika, a slične rezultate postiže i u kolorizaciji. Slijede objašnjenja dva takva transformerska modela.

4.1. ColTran (Colorization Transformer)

Model ColTran predstavljen je u radu „Colorization Transformer“ ([16]). Ovaj model koristi pozornost na malo drugačiji način pomoću *axial transformera* (transformer po osima) predstavljenim u radu [17]. Kao što sam naziv kaže, pozornost se u ovoj vrsti transformera računa po osima, odnosno po redcima i stupcima. Glavna prednost ovakvog računanja pozornosti je manja složenost ($O\left(D\sqrt{D}\right)$ umjesto $O(D^2)$ gdje je D dimenzija ulaznog *embeddinga*) i manje slojeva uz očuvano globalno receptivno polje.



Slika 11. Arhitektura ColTran modela preuzeta iz rada

Arhitektura modela prikazana je na slici 11. Zadatak kolorizacije podijeljen je na tri manja zadatka odnosno tri neovisno učena modela. Prvi je zadatak postići grubu autoregresivnu kolorizaciju niske rezolucije, drugi je naduzorkovanje (eng. *upsampling*) boje a treći *upsampling* prostora. U ovom kontekstu *upsampling* označava povećanje rezolucije slike. Prvi model sastoji se od *axial* transformera, a druga dva sastoje se od blokova pozornosti koji deterministički i paralelno obavljaju *upsampling* boje i prostora grubo obojane slike. Drugim riječima, prvi model daje grubu skicu kako bi obojana slika trebala izgledati. Ova skica je mutna (niske rezolucije) pa *upsampleri* imaju zadatak obnoviti sliku u rezoluciju jednaku ulaznoj.

Model je učen na slikama rezolucije 256x256 piksela iz ImageNet skupa, a rezultati su bolji od dotadašnjih transformerskih rješenja za kolorizaciju. Rezultati su prikazani na slici 12, a posebno je zanimljivo da ColTran postiže preko 60% na ljudskoj evaluaciji (*AMT Fooling Rate*) što znači da se prosječnoj osobi predstavila stvarna slika i ista ta slika koju je obojao ColTran, te je ColTranova slika u 60% slučajeva zavarala čovjeka da je originalna.

Models	FID
ColTran	**19.37 ± 0.09**
ColTran-B	19.98 ± 0.20
ColTran-S	22.06 ± 0.13
PixColor [16]	24.32 ± 0.21
cGAN [3]	24.41 ± 0.27
cINN [1]	25.13 ± 0.3
VAE-MDN [11]	25.98 ± 0.28
Ground truth	14.68 ± 0.15
Grayscale	30.19 ± 0.1

Models	AMT Fooling rate
ColTran (Oracle)	62.0 % ± 0.99
ColTran (Seed 1)	40.5 % ± 0.81
ColTran (Seed 2)	**42.3 % ± 0.76**
ColTran (Seed 3)	41.7 % ± 0.83
PixColor [16] (Oracle)	38.3 % ± 0.98
PixColor (Seed 1)	33.3 % ± 1.04
PixColor (Seed 2)	35.4 % ± 1.01
PixColor (Seed 3)	33.2 % ± 1.03
CIC [56]	29.2 % ± 0.98
LRAC [27]	30.9 % ± 1.02
LTBC [22]	25.8 % ± 0.97

Slika 12. Evaluacija ColTran modela preuzeta iz rada

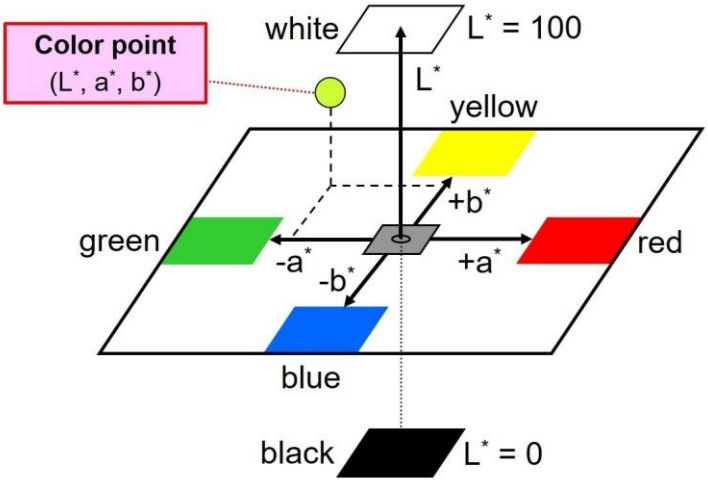
4.2. DDColor (Dual Decoder Color)

DDColor je transformerski model za kolorizaciju predstavljen u radu „DDColor: Towards Photo-Realistic Image Colorization via Dual Decoders“ ([17]). U trenutku pisanja ovog rada jedan je od najaktualnijih i najefektivnijih modela za problem kolorizacije slika. Kao što naziv sugerira, model se sastoji od dva dekodera od kojih je jedan zadužen za piksele a drugi za boje. Dekoderi rade zajedno kako bi pronašli korelacije među bojama i semantičkim reprezentacijama značajki pomoću mehanizma pozornosti.

Zadatak modela je za ulaznu crno-bijelu sliku u obliku matrice x_L dimenzije $HxWx1$ predvidjeti dva nedostajuća kanala boje, odnosno matricu $\hat{y}_{AB}$ dimenzija $HxWx2$. Vrijednost *L* predstavlja svjetlinu ili intenzitet (nijansa sive) piksela, a kanali *AB* predstavljaju koloritnost (eng. *chrominance*) piksela. Prikazivanje boja pomoću vrijednosti *L, A i B* potiče iz CIELAB krominantnog dijagrama ([18]).

Ovaj dijagram cijeli spektar boja vidljivih ljudskom oku pretvara u trodimenzionalni prostor kojemu su *A, B* i *L* koordinate. Vizualizacija dijagrama prikazana je na slici 13 ([18]).

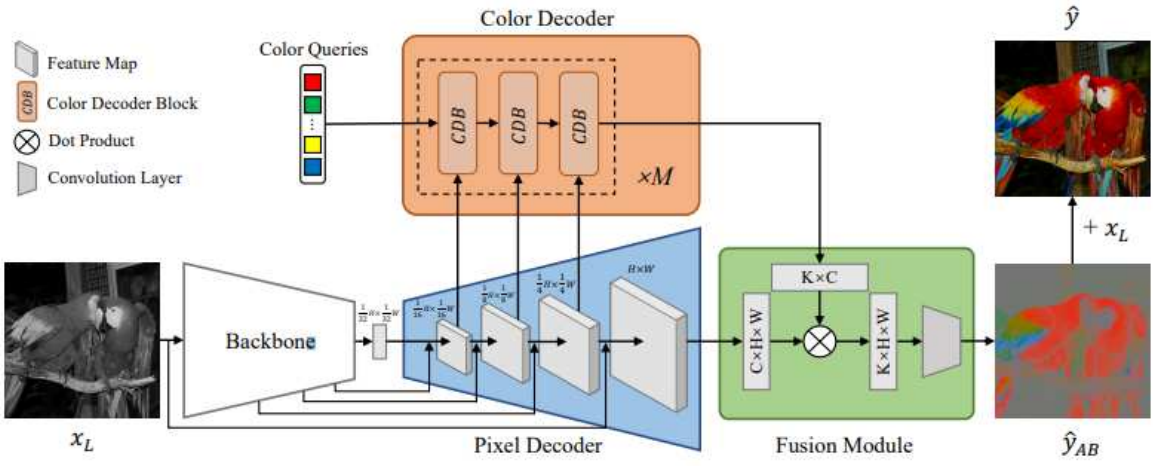
Sa slike je vidljivo da ako je poznata samo vrijednost *L*, spektar boja ograničen je na crnu (*L* = 0), bijelu (*L* = 100) i nijanse sive između njih.



Slika 13. CIELAB *color space* (koloritetni dijagram)

4.2.1. Arhitektura

Arhitektura modela sastoji se od enkodera čiji izlaz ulazi paralelno u dva dekodera, čime se model razlikuje od standardne implementacije transformerske arhitekture. Dijagram arhitekture prikazan je na slici 14.



Slika 14. Dijagram arhitekture DDColor modela preuzet iz rada

Slijedeća razlika je u enkoderu koji umjesto mehanizma pozornosti koristi proizvoljnu mrežu za koju je jedino bitno da ima sposobnost izlučivanja semantičkih značajki. Autori rada koriste mrežu ConvNeXt koja je tom trenutku jedna od najefektivnijih konvolucijskih mreža za problem klasifikacije slika. ConvNeXt je predstavljen u radu [19] a cilj rada bio je dizajnirati konvolucijsku mrežu koja može konkurirati transformerima u klasifikaciji slika.

4.2.2. Enkoder

Zadatak enkodera je pripremiti 4 mape značajki na različitim skalama. Preciznije, rezolucije tih matrica su $\frac{H}{4} \times \frac{W}{4}, \frac{H}{8} \times \frac{W}{8}, \frac{H}{16} \times \frac{W}{16}$ i $\frac{H}{32} \times \frac{W}{32}$ gdje su H i W visina i širina ulazne slike. Mapa značajki (eng. *feature map*) može se definirati kao semantička interpretacija niza piksela, odnosno sadrži informacije o tome što se gdje nalazi na slici. Koriste se 4 različite mape na različitim skalama s ciljem da se uhvate manje (eng. *low level*) i veće (eng. *high level*) značajke. Primjerice, na fotografiji krajolika, *high level* značajke mogu biti nebo, šuma i livada, a *low level* značajke bile bi pojedino stablo, grm na livadi, ili oblak na nebu. *High level* značajke proizlaze iz mapa značajki manjih rezolucija, a *low level* značajke iz matrica značajki većih rezolucija.

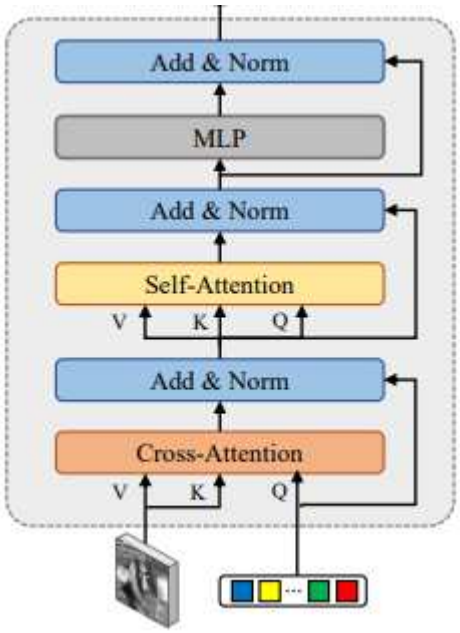
4.2.3. Dekoder piksela

Dekoder piksela sastoji se od 4 *upsampling* sloja koji primaju mape značajki iz enkodera počevši od najmanje rezolucije. Zadatak ovih slojeva je povećati razinu detalja u mapama značajki povećanjem rezolucije sve dok rezolucija ne bude jednaka ulaznoj ($H \times W$). U prijašnjim modelima koji se bave sličnom problematikom za *upsampling* se najčešće koristila interpolacija ili dekonvolucija, međutim autori ovog modela odabrali su PixelShuffle. PixelShuffle je *upsampling* metoda koja je predstavljena u radu [20] a temelji se na konvolucijskoj mreži s dodatkom periodičkog premještanja (eng. *shuffling*) piksela iz tenzora (matrice) niske rezolucije u tenzor visoke rezolucije.

4.2.4. Dekoder boja

Dekoder boja još je jedan element modela kroz koji autori rada unose inovaciju u području kolorizacije. Većina prijašnjih modela koristi neku vrstu predznanja (eng. *prior*) kako bi postigla vividne rezultate. Primjerice, to može biti generativno predznanje iz predtreniranog GAN modela ([21]) koji uče distribuciju boja u prirodnim slikama koja se zatim može koristiti kao *prior* za druge modele. Još jedna vrsta predznanja je statistička distribucija, a za nju je potrebno izračunati frekvencije pojavljivanja određenih boja i kombinacija boja na nekom velikom skupu slika. Konačno, moguće je i pripremiti predznanje za model stvaranjem parova značajka-boja iz skupa ručno opisanih slika. Očito je da svaka od ovih vrsta predznanja zahtijeva puno pripreme i truda, a i smanjuje sposobnost modela da generalizira s obzirom da su sve tri vrste predznanja naučene na određenom

skupu. Upravo zato autori DDColor modela ne koriste nikakvo predznanje u dekoderu boja nego se oslanjaju na izlaz enkodera piksela, *query* boja kao ulaz, te mehanizam pozornosti.



Slika 15. Arhitektura dekodera boja (CDB – *color decoder block*)

4.2.5. CDB (blok dekodera boje)

Arhitektura jednog bloka dekodera boja prikazan je na slici 15. Dekoder boja sastoji se od više ovakvih blokova gdje izlaz prvog postaje ulaz drugog sve do zadnjeg bloka. Dekoder je sličan prethodno opisanom dekoderu iz standardnog transformera. Prva razlika vidljiva je na ulazu, gdje se enkoder-dekoder pozornost (eng. *cross-attention*) primjenjuje već u prvom sloju. Matrica *Q* sadrži *embedding* boja, a *K* i *V* sadrže semantičke informacije o značajkama na slici. *Embedding* boja autori izražavaju na slijedeći način:

$$Z_0 = [Z_0^1, Z_0^2, \dots, Z_0^K] \in \mathbb{R}^{K \times C}$$

Vrijednosti *embeddinga* boja prilikom treniranja postavljaju se u nule i šalju u prvi blok dekodera boja, koji primjenjuje funkciju:

$$Z_l' = softmax(Q_l K_l^T) V_l + Z_{l-1}$$

Varijabla *l* označava indeks sloja, pa će prvi blok računati s *embeddingom* Z_0. Matrica *Q* računa se linearnom transformacijom *embeddinga* boja:

$$Q_l = f_Q(Z_{l-1})$$

Isto vrijedi i za *K* i *V* koji se dobiju linearnim transformacijama nad izlazom enkodera piksela. Ovim operacijama *embedding* boja obogaćen je semantičkim informacijama odnosno značajkama. Nakon toga, *embedding* prolazi još nekoliko slojeva:

$$Z_l'' = MSA\big(LN(Z_l')\big) + Z_l'$$

$$Z_l''' = MLP\big(LN(Z_l'')\big) + Z''_l$$

$$Z_l = LN(Z_l''')$$

Ovo su već prethodno objašnjeni *multi-head attention* (MSA), potpuno povezani sloj (MLP) i normalizacija sloja (LN).

Kao što je spomenuto, dekoder boja sastoji se od više ovakvih blokova. Blokovi su grupirani u grupe po 3 bloka, a dekoder piksela šalje grupi blokova mape značajki skalirane u 3 različite rezolucije što se može vidjeti na slici 14. Dekoder boja koristi različite razine detalja u mapama značajki kako bi još preciznije naučio bojati pojedine značajke, što je različito od sličnih prijašnjih modela koji uče na samo jednoj rezoluciji mape značajki.

Konačno, izlaz dekodera piksela i dekodera boja kombinira se u elementu koji autori nazivaju *fusion module*. Zadatak ovog elementa je iz kombinacije ova dva izlaza predvidjeti konačnu rezultantnu kolorizaciju slike. Ulazi u element su semantički *embedding* značajki $E_i \in \mathbb{R}^{C\times H\times W}$ što odgovara izlazu enkodera piksela i *embedding* boja obogaćen skaliranim mapama značajka $E_c \in \mathbb{R}^{K\times C}$ što odgovara izlazu enkodera boja. *C* označava dimenziju *embeddinga* a *K* broj *embeddinga* boja. *Fusion module* ove *embeddinge* kombinira u značajku $\hat{F} \in \mathbb{R}^{K\times H\times W}$ koristeći skalarni produkt:

$$\hat{F} = E_c \cdot E\_i$$

Na prethodni rezultat primjenjuje konvolucijski sloj veličine 1×1 kako bi se generirali kanali boja:

$$\hat{y}_{ab} = Conv(\hat{F}).$$

Konačno, obojena slika dobiva se konkateniranjem kanala boja *AB* na crno-bijelu sliku x_L.

4.2.6. Treniranje

Tijekom treniranja modela, primijenjuju se četiri različite funkcije gubitka. Prva od njih je *pixel loss* odnosno gubitak po pikselu. Računa se kao L1 udaljenost između izlaza modela

$\hat{y}$ i stvarne slike y što navodi model da rezultantna slika bude što bliža stvarnoj. Slijedeći gubitak je *perceptual loss* koji nastoji minizirati semantičku razliku između slika $\hat{y}$ i y. U svrhu računanja ovog gubitka koristi se predtrenirani VGG16 model ([22]) koji razlučuje značajke iz obje slike, koje se zatim uspoređuju kako bi se navelo model da rezultantna slika ne gubi semantičko značenje stvarne slike. Slijedeći gubitak je *adversarial loss* koji dolazi iz diskriminatorske mreže čiji je zadatak raspoznati sliku $\hat{y}$ i y. Ovaj gubitak navodi model da razlika između izlazne i stvarne slike bude neprepoznatljiva. Konačno, primijenjuje se i *colorfulness loss* koji navodi model da slike boja živim i šarenim bojama, a računa se pomoću prosjeka i standardne devijacije boje piksela što znači da je gubitak manji što je slika šarenija:

$$\mathcal{L}_{col} = 1 - [\sigma_{rgyb}(\hat{y}) + 0.3 \cdot \mu_{rgyb}(\hat{y})]/100$$

Ovaj je mehanizam detaljnije opisan u radu [23].

Ukupni gubitak izražava se na slijedeći način (lambda vrijednosti označavaju težine različitih funkcija gubitaka):

$$\mathcal{L}_{\theta} = \lambda_{pix}\mathcal{L}_{pix} + \lambda_{per}\mathcal{L}_{per} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{col}\mathcal{L}_{col}$$

Model je treniran i testiran na tri različita skupa podataka. Točnije, to su *ImageNet* (1.3 milijuna slika za učenje i 50000 za testiranje), *COCO-Stuff* koji se sastoji od raznih slika prirode i testira na 5000 slika, i *ADE20K* koji se testira na 2000 slika. Performanse modela evaluiraju se slijedećim mjerama:

- FID (*Fréchet inception distance*) je metrika za kvantificiranje realizma i raznolikosti slika, a računa se uspoređivanjem sličnosti u distribuciji boja između kolorizirane i originalne slike [24]
- CF (*Colorfulness score*) je metrika koja mjeri šarenost slike i odražava vividnost koloriziranih slika
- PSNR (*Peak Signal-to-Noise Ratio*) je mjera koja računa šum unesen u sliku kolorizacijom u odnosu na originalnu sliku, a smatra se da nije adekvatan pokazatelj kvalitete kolorizacije

Autori model treniraju koristeći navedene 4 funkcije gubitka i 3 evaluacijske metrice na 3 skupa podataka uz slijedeće postavke. Koristi se optimizator *AdamW* sa vrijednostima $\beta_1 = 0.9, \beta_2 = 0.99,\ weight_{decay} = 0.01$. Stopa učenja postavlja se na $1e^{-4}$. Težine

funkcija gubitaka su $\lambda_{pix} = 0.1, \lambda_{per} = 5.0, \lambda_{adv} = 1.0, \lambda_{col} = 0.5$. Kao enkoder koristi se spomenuta mreža ConvNeXt-L. U dekoderu piksela, koriste se dimenzije značajki 512, 512, 256 i 256. Dekoder boja sastoji se od ukupno devet CDB-ova, a broj *embeddinga* boja je 100. Treniranje se obavlja u 400000 iteracija, a *batch size* je 16. Stopa učenja se smanjuje za pola nakon 80000 iteracija i zatim svakih 40000 iteracija. Veličina slika za učenje postavlja se na 256×256 piksela. Učenje se obavljalo paralelno na četiri Tesla V100 grafičke kartice.

4.2.7. Rezultati

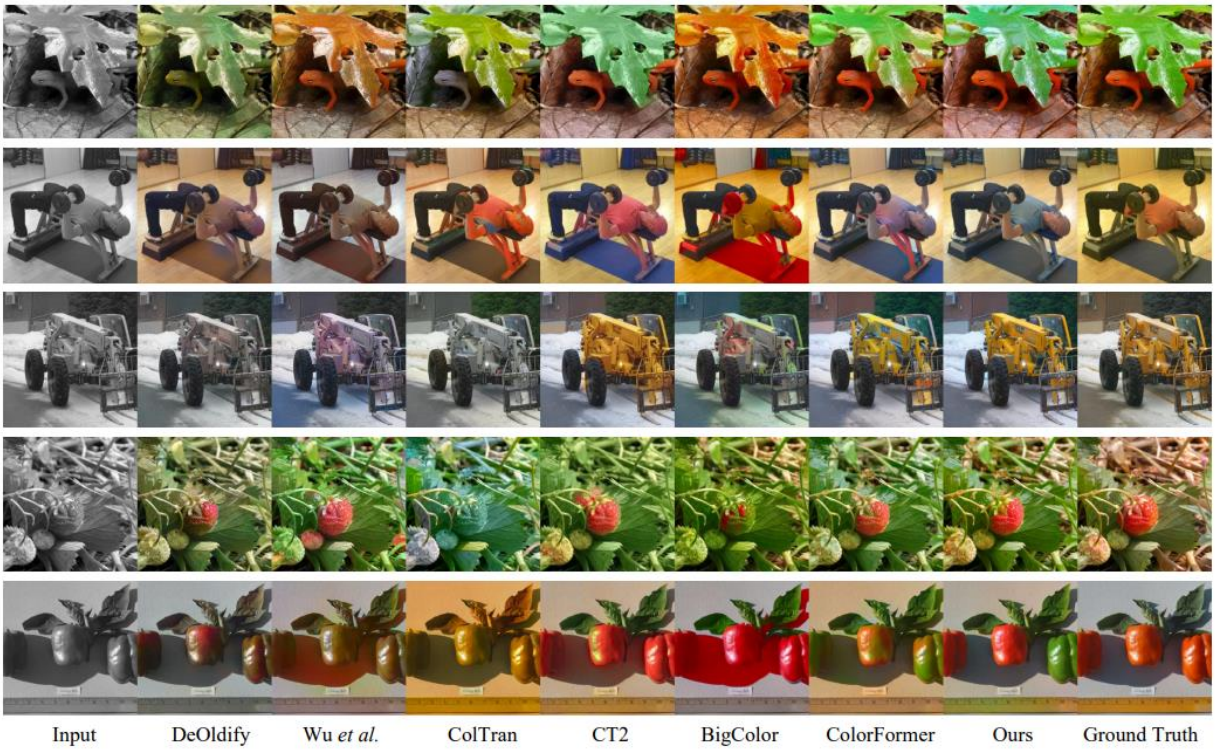
Vizualizacija ovako naučenog modela u usporedbi s konkurentnim modelima prikazana je na slici 16. Vidljivo je da DDColor generira realistične i vividne boje, a slike u nekim slučaevima izgledaju ugodnije oku od originalnih (eng. *ground truth*).

Na slici 17 prikazana je detaljna evaluacija modela koristeći spomenute metrike u usporedbi s konkurentnim modelima. Vidljivo je da DDColor postiže najbolje iznose u FID metrici (niži broj je bolji) na svim testnim skupovima. Osim toga, iako neki modeli slike bojaju šarenije (veći CF iznos), to ne znači nužno da je slika bolje obojana. Zato autori rada računaju ΔCF koji govori kolika je razlika u šarenosti originalne i kolorizirane slike. U ovoj metrici DDColor je uvjerljivo na prvom mjestu (manja razlika je bolja).

Razlozi uspjeha ovog modela su u mnogim implementiranim unapređenjima. Kao što je prethodno spomenuto, veliki utjecaj imaju *multi-scale* mape značajki u dekoderu piksela koje omogućavaju modelu da bolje odredi točne granice semantičkih značajki na slici što umanjuje učinak prelijevanja boja (eng. *color bleeding*) koji je prisutan u nekim konkurentnim modelima. Do prelijevanja boja dolazi kada model nema dovoljno jasnu sliku o tome gdje je točno koja značajka i gdje počinje i završava. S obzirom na to, kada ide značajku obojati, nije neobično da se dio te boje „prelije“ izvan granica same značajke. Na slici 16, *color bleeding* se može najviše primijetiti kod BigColor modela.

Još jedan problem s kojim se modeli kolorizacije susreću su isprane, blijede slike. Prekomjerno oslanjanje na gubitak po pikselu vodi rezultatima koji su slabo obojani, a primjer je vidljiv kod modela DeOldify. Razlog tome je što model učen na velikom broju slika može minimizirati funkciju gubitka po pikselu tako da vrlo konzervativno i slabo boja slike, jer će gubitak po pikselu biti veći ako nešto oboja prejako nego preslabo. S ciljem dobivanja šarenijih i vividnijih rezultata DDColor, a i neki drugi modeli, uvode *colorfulness*

loss kako bi naveli modele da teže šarenijim slikama. Dodatno, dekoder boja koji je također učen na *multi-scale* značajkama kako je prethodno opisano doprinosi šarenosti slika zbog dodatnog semantičkog razumijevanja odnosa između značajki i njihovih prirodnih boja.



Slika 16. Vizualna usporedba rezultata DDColora i prijašnjih modela

Method	#Params.	ImageNet (val5k)				ImageNet (val50k)				COCO-Stuff				ADE20K*			
		FID↓	CF↑	ΔCF↓	PSNR↑	FID↓	CF↑	ΔCF↓	PSNR↑	FID↓	CF↑	ΔCF↓	PSNR↑	FID↓	CF↑	ΔCF↓	PSNR↑
CIC[49]	32.2M	8.72	31.60	6.61	22.64	19.17	**43.92**	4.83	20.86	27.88	33.84	4.40	22.73	15.31	31.92	3.12	23.14
InstColor[39]	69.4M	8.06	24.87	13.34	23.28	7.36	27.05	12.04	22.91	13.09	27.45	10.79	23.38	15.44	23.54	11.50	24.27
DeOldify[1]	63.6M	6.59	21.29	16.92	**24.11**	3.87	22.83	16.26	22.97	13.86	24.99	13.25	**24.19**	12.41	17.98	17.06	**24.40**
Wu *et al.* [46]	310.9M	5.95	32.98	5.23	21.68	3.62	35.13	3.96	21.81	-	-	-	-	13.27	27.57	7.47	22.03
ColTran [24]	74.0M	6.44	34.50	3.71	20.95	6.14	35.50	3.59	22.30	14.94	36.27	1.97	21.72	12.03	34.58	0.46	21.86
CT2 [45]	463.0M	5.51	38.48	0.27	23.50	4.95	39.96	0.87	22.93	-	-	-	-	11.42	**35.95**	0.91	23.90
BigColor [13]	105.2M	5.36	**39.74**	1.53	21.24	1.24	40.01	0.92	21.24	-	-	-	-	11.23	35.85	0.81	21.33
ColorFormer [47]	44.8M	4.91	38.00	0.21	23.10	1.71	39.76	0.67	23.00	8.68	36.34	1.90	23.91	8.83	32.27	2.77	23.97
DDColor-tiny	55.0M	4.38	37.66	0.55	23.54	1.23	37.72	1.37	23.63	7.24	**38.48**	**0.24**	23.45	10.03	35.27	**0.23**	24.39
DDColor-large	227.9M	**3.92**	38.26	**0.05**	23.85	**0.96**	38.65	**0.44**	**23.74**	**5.18**	**38.48**	**0.24**	22.85	**8.21**	34.80	**0.24**	24.13

Slika 17. Usporedba performansi s konkurentnim modelima

5. Kolorizacija crno-bijelih slika Gustava Klimta

Zadatak ovog rada je pronaći prikladan model za kolorizaciju crno-bijelih slika, istrenirati model na proizvoljnom skupu podataka i konačno dobiti prihvatljive rezultate na skupu slika umjetnika Gustava Klimta. Nakon prethodnog pregleda postojećih modela za kolorizaciju crno-bijelih slika, zaključeno je da će se u svrhu zadatka koristiti model DDColor zbog toga što predstavlja vrhunac postignuća u ovom području.

5.1. Gustav Klimt

Gustav Klimt bio je najvažniji slikar austrijske secesije ([25]). Rođen je 14. srpnja 1862. u Beču a preminuo je 6. veljače 1918. godine. Poznat je po svojem vrlo specifičnom stilu koji se može svrstati u umjetnički pokret art nouveau (nova umjetnost). Njegove slike pretežito prikazuju žene, a često su opisane i kao pornografske.

Slika 18. Najpoznatija slika Gustava Klimta, Poljubac (1907/08.)

5.2. Skup podataka

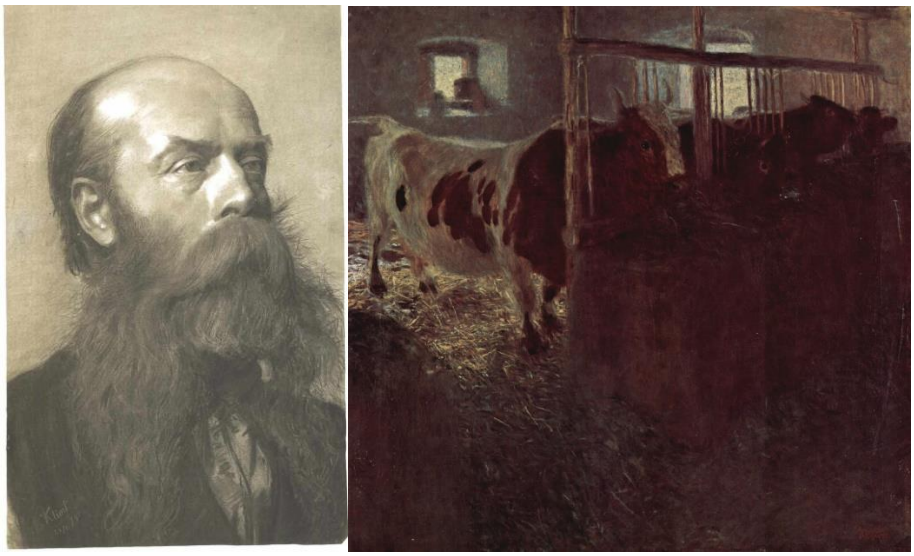
Skup podataka sastoji se od 171 slike Gustava Klimta u JPG formatu. Skup nije podijeljen po razdobljima ili stilovima, tako da se sastoji od gotovo svih njegovih slika i crteža koji su dostupni u digitalnom obliku, od najapstraktnijih do najrealističnih. Slike su

prikupljene s Interneta. Zatim, programski je nasumice odabrano 30 slika za testnu skupinu, a ostatak je ostavljen za treniranje modela.

Skup za treniranje od samo 141 slike nekoliko je redova veličine manji nego što model kao DDColor zahtijeva za dobre rezultate. Zato je za svrhu zadatka preuzet još jedan znatno veći skup podataka pod imenom WikiArt. WikiArt je skup podataka koji se sastoji od ukupno 52757 slika od 195 različitih autora. Također, slike su podijeljene u kategorije po odgovarajućim umjetničkim stilovima. Ovaj je skup podataka skupljen sa istoimene internetske enciklopedije vizualne umjetnosti, a napravljen je u svrhu treniranja ArtGAN generativnog modela u radu [26]. Ovaj će se skup koristiti za treniranje modela, a zatim će se za *fine-tuning* koristiti manji skup probranih slika kako bi se ostvarili što bolji rezultati na testnom skupu slika Gustava Klimta.

Testni skup sastoji se specifično od slika Gustava Klimta iz nekoliko razloga. Neke od originalnih slika Gustava Klimta su uništene ili izgubljene, a preostale su samo crno-bijele fotografije. Model koji može automatski predložiti kako su te slike originalno izgledale mogao bi biti od umjetničke važnosti. Osim toga, Gustav Klimt slikao je u raznim stilovima od kojih je većina apstraktna, pa je dodatan izazov naučiti model da pokrije sve testne slučajeve i dobro ih oboja. Kao što je spomenuto ranije, bojanje fotografija ili vrlo realističnih slika je gotovo trivijalno uz kvalitetno dizajniran i naučen model, dok je ovako specifičan zadatak znatno zahtjevniji. Neke od testnih slika prikazane su na slikama 8 i 9, a vidljivo je koliko su različiti njegovi stilovi, od crno-bijelih skica do apstraktnih, šarenih prikaza prirode pa do realističnog portreta.

Slika 19. a) Adele Bloch Bauer, 1912. b) Stablo jabuke, 1912.



Slika 20. a) Čovjek s bradom iz profila, 1879. b) Krave u štali, 1901.

5.3. Treniranje modela

Model je treniran u nekoliko navrata na različitim skupovima podataka dok nisu postignuti zadovoljavajući rezultati. Prvi pokušaj bio je provjera kako model s predtreniranim težinama boja slike Gustava Klimta, a rezultati su nezadovoljavajući što je i očekivano. Model je u sklopu rada [17] treniran na ImageNet skupu podataka koji se sastoji od 1.3 milijuna slika. Svaka od tih slika je fotografija nekog prizora iz stvarnog svijeta, stoga nije neobično da tako naučen model boja i slike Gustava Klimta na realističan način, dok ga apstraktnije slike poprilično zbune. Primjeri su vidljivi na slikama 21, 22 i 23 gdje lijeva slika prikazuje rezultat modela predtreniranog na fotografijama, a desna Klimtov original.

Slika 21. Kammer Chateau near Attersee I, 1908.

Slika 22. Study of the Head of a Blind Man, 1896.

Slika 23. Očekivanje, 1909.

Zaključak testiranja predtreniranog modela na testnim slikama je da učenje na realističnim fotografijama nije dovoljno za uspješnu kolorizaciju apstraktnijih slika, te je stoga sljedeća ideja naučiti model na skupu koji se sastoji isključivo od umjetničkih djela kako bi model naučio kako umjetnici obično bojaju određene značajke. Taj skup je prethodno opisani WikiArt, a sastoji se od skoro 80 tisuća slika nakon što su neke slike izuzete.

Učenje je odrađeno sa većinom jednakim postavkama kao u originalnom radu. Zbog manjeg skupa podataka i ograničenosti tehničke opreme korištene za učenje modela (grafička kartica GTX 1080) broj iteracija smanjen je na 40000 a *batch size* na 4. Stopa učenja inicijalizirana je na $1e^{-4}$ ali se smanjuje nakon 4000 iteracija, a zatim svakih 2000 iteracija. Skup podataka augmentiran je udvostručavanjem skupa s horizontalno zrcaljenim slikama, što dovodi veličinu skupa na 160 tisuća. To znači da model svaku sliku vidi dvaput, jednom u originalnom obliku i jednom u zrcaljenom (160 tisuća slika = 40000 iteracija * 4 *batch size*). Performanse ovako učenog modela testirane su na testnom skupu koji se sastoji od 150 Klimtovih slika kako bi se procijenilo postoji li poboljšanje u odnosu na originalni predtrenirani model. Rezultat ovog učenja je slijedeći: FID iznos od 43.4943, CF od 46.3912. Neki primjeri kolorizacije Klimtovih slika ovim modelom prikazani su na slikama 24, 25 i 26. Iako su rezultati nešto bolji od prethodnog modela vidi se prevladavajući utjecaj žute boje čak i na slikama gdje je ne bi trebalo biti toliko. Pretpostavka je da je žuta boja prezasićena u skupu WikiArt kao i slikarstvu općenito, što navodi model da ju prečesto koristi.

Slika 24. Močvara, 1900.

Slika 25. Breza u šumi, 1903.

Slika 26. Nada II, 1908.

Ipak, iz rezultata je očito da je model naučio odnos značajki i boja, te smisleno odabire bojati različite dijelove slika. Slijedeća je ideja, koristeći ovaj model naučen na WikiArt skupu kao temelj, napraviti *fine-tuning* (fino podešavanje) modela na manjem skupu slika kako bi se približilo Klimtovom stilu bojanja. Kao što je prethodno spomenuto Gustav Klimt svrstava se u umjetnički pokret pod imenom Art nouveau ([27]), pa je iz WikiArt skupa izdvojen skup svih slika iz tog pokreta. Takvih slika ima 3807, a ubačene su i neke Klimtove slike, od kojih je 30 ostavljeno kao testni skup.

Postavke učenja su iste osim što je broj iteracija sada 5000, a stopa učenja se ne smanjuje i inicijalizirana je na $5e^{-4}$ s ciljem bržeg učenja zbog manje iteracija. Ovo znači da će model vidjeti svaku sliku iz skupa više puta i jače ažurirati težine za svaku od njih.

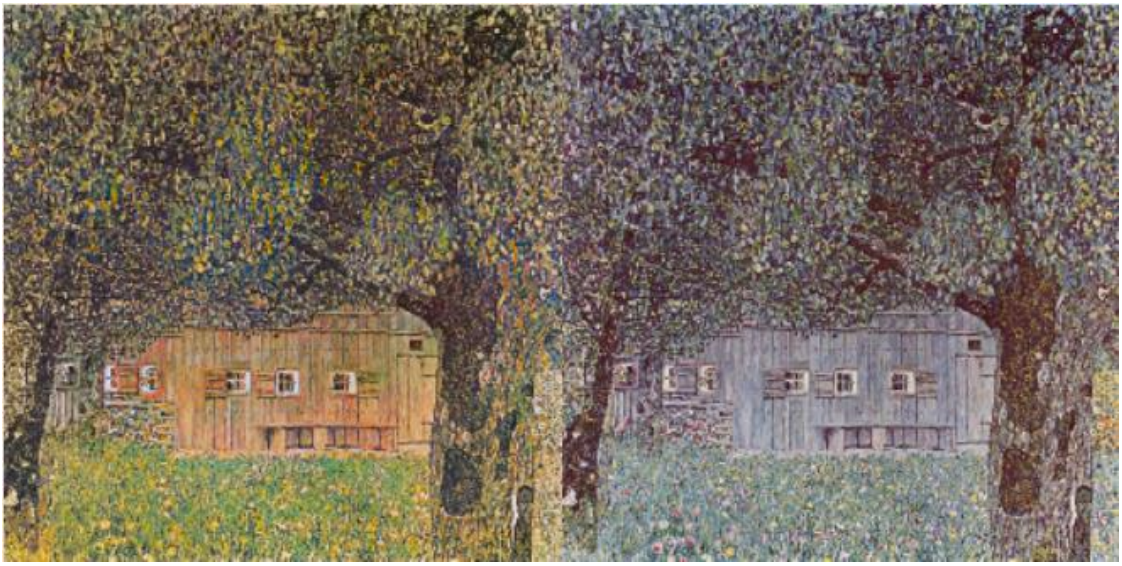
Ovakav se pristup nije ispostavio dobar, a FID mjera ispala je 60.5810, dok je CF 50.0941. Detaljnijom kvalitativnom analizom rezultantnih slika primijećeno je da je žuta boja još dominantnija, što je vjerojatno posljedica povećane stope učenja i prirode Art nouveau stila.

Nakon ovog neuspjeha ideja je bila ponovo uzeti temeljni WikiArt model i pripremiti skup slika sličniji Klimtu. Iako bi za neke druge umjetničke stilove prethodni pristup možda funkcionirao zbog sličnosti stila bojanja i raznolikosti i šarenosti boja, Art nouveau po svojoj prirodi nema neki zadani stil i sastoji se od gomilu različitih tehnika, simbolika i motiva. Što se tiče izbora boja, žuta odnosno zlatna boja dominira u ovom periodu još više nego što je uobičajeno. Međutim, postoji podskupina Art nouveau pokreta koja je vrlo bliska Klimtu, a to je takozvana bečka secesija ([28]). U ovaj se pokret osim Gustava Klimta ubrajaju još i Josef Hoffman, Koloman Moser i Otto Wagner. Daljnjim istraživanjem pronađeno je nekoliko umjetnika koje je Gustav Klimt inspirirao ili koji slikaju sličnim stilom, a to su Alphonse Mucha, Egon Schiele, Ferdinand Hodler, Marc Chagall, Oskar Kokoschka i Sonia Delaunay. Iz WikiArt skupa i s Interneta okupljeno je što više slika pod autorstvom imenovanih autora, a rezultira skupom od 1364 slika što uključuje i Klimtov skup za učenje (141 slika). Učenje je odrađeno u 15000 iteracija sa stopom učenja $1e^{-4}$ koja se smanjuje nakon 7500 i nakon 10000 iteracija, a zatim svakih 1000 iteracija. S obzirom da je *batch size* 4, to znači da će model vidjeti 60000 slika iz skupa koji, nakon augmentacije, broji 2728 slika. To znači da će model svaku sliku vidjeti ukupno 22 puta, s ciljem da se *overfita* (prenauči) na skupu slika sličnim Klimtu kako bi bolje kolorizirao neviđene slike Klimta iz testnog skupa.

6. Rezultati

Rezultati prethodno opisanog učenja su zadovoljavajući. Neki primjeri prikazani su na slikama 27, 28 i 29.

Slika 27. Crni pernati šešir, 1910.



Slika 28. Kuća u gornjoj Austriji, 1911.

Slika 29. Schloss Kammer am Attersee III, 1909/10.

Na testnom skupu vidljiv je smanjen utjecaj žute boje, što je postignuto pomnim odabirom umjetnika u skupu za učenje, čiji stilovi uključuju sličan odabir boja kao i Klimt. Osim toga, semantičke značajke obojane su vrlo smisleno i u stilu Klimta, iako odabir boja nije savršen. Međutim kao što je opisano ranije, odabir boja ne može biti savršen, a pogotovo na ovako specifičnom problemu. Jedan se dio informacija o originalnoj slici nepovratno gubi, a kontekst i predznanje dobiveni učenjem na skupu slika sličnih Klimtu i Klimtovih ostalih slika ne može se primijeniti na sve testne primjere. S obzirom na ograničenu količinu dostupnih digitaliziranih slika Klimta, vrlo je moguće da među tim slikama postoje potpuno jedinstveni primjeri bojanja određenih značajki koje je zbog jedinstvenosti nemoguće točno predvidjeti. Ipak, rezultati kolorizacije su zadovoljavajući i svakako mogu poslužiti kao prvi korak u kolorizaciji Klimtovih crno-bijelih slika. Što se tiče evaluacijskih metrika, postignut je FID iznos od 40.0640 i CF iznos od 37.2704, a PSNR iznosi 21.8491.

7. Rasprava

Dobiveni rezultati posljedica su detaljnog i pomnog odabira skupa podataka i površnog podešavanja hiperparametara jednog specifičnog modela. Ovi bi se rezultati mogli poboljšati na nekoliko načina.

Primjena drugih vrsta modela s naglaskom na konvolucijske mreže mogla bi poboljšati kolorizaciju s obzirom da konvolucijska arhitektura ima ugrađenu induktivnu pristranost hijerarhijskom prepoznavanju značajki što u teoriji znači da se model može bolje naučiti s manjim skupom podataka. Međutim, model DDColor u enkoderu već koristi konvolucijsku arhitekturu tako da se ne očekuje znatno poboljšanje.

Postprocessing (naknadna obrada) rezultantnih slika mogla bi pomoći dovesti kolorizirane slike Klimta bliže onome kako bi stvarno trebale izgledati. S obzirom da je stil bojanja uspješno naučen, ponekad pogrešan odabir boja mogao bi se popraviti naknadnim učenjem nekog modela dubokog učenja čiji bi zadatak bio korigirati boje podešavanjem kontrasta, topline i sličnih aspekata slike. Međutim ovaj bi pristup uveo još jednu neovisnu mrežu koju bi također trebalo naučiti na vrlo malom skupu podataka za učenje.

Detaljnije podešavanje hiperparametara DDColor modela moglo bi dovesti boljim rezultatima. Zbog ograničenih tehničkih kapaciteta u ovom radu nije bilo moguće testirati sve moguće smislene kombinacije hiperparametara i testirati rezultate kako bi se pronašao najbolji pristup. Model DDColor sastoji se od skoro 240 milijuna parametara što znači da na grafičkoj kartici korištenoj u svrhu ovog rada jedna iteracija učenja traje 5 do 8 minuta. No ovaj je model moguće podesiti na puno načina, uključujući korištenje druge mreže u enkoderu umjesto ConvNeXt-a, ili drugog diskriminatora za računanje $\mathcal{L}_{adv}$ gubitka. Također, moguće je povećati ili smanjiti broj skala mapa značajki u enkoderu, povećati broj blokova dekodera boja, te namještati stopu učenja, težine funkcija gubitaka i slično.

Dodatak ekspertnog znanja mogao bi pomoći u dobivanju boljih rezultata. Iako DDColor nastoji izbjeći korištenje ručno pripremljenog predznanja, u ovako specifičnom problemu bilo bi vrlo korisno pripremiti predznanje u oblikova parova značajki i pripadajućih boja. Na ovaj način stručnjak bi mogao unijeti svoje znanje u model i pomoći modelu da nauči kolorizirati čak i iz vrlo malog skupa podataka. Međutim, ovaj pristup je subjektivan i ovisno o stručnjaku moguće je dobiti različite rezultate, te zahtijeva dodatan ljudski doprinos i predprocesiranje predznanja kako bi model funkcionirao.

Zaključak

Tehnike dubokog učenja u zadnjih se nekoliko godina razvijaju sve bržim tempom, što je pokazano i ovim radom. Prije samo nekoliko godina predstavljen je prvi model transformera i primijenjen na problem prevođenja, a ubrzo nakon toga i na klasifikaciju slika. Iako je transformerski model uvjerljivo istisnuo konvolucijske neuronske mreže u području računalnog vida, kao i LSTM arhitekturu u području obrade prirodnog jezika, u zadnje vrijeme istraživanja na tu temu kreću se u smjeru povezivanja te dvije arhitekture i korištenja prednosti svake od njih za postizanje još boljih rezultata.

Primjena transformera na problem kolorizacije, koji je specifično zahtjevan zbog svoje stohastičke prirode uzrokovane potpunim nedostatkom informacija o bojama u crno-bijelim slikama, pokazala se poprilično uspješnom. Mehanizam pozornosti vrlo je snažan, efikasan i efektivan za razne probleme, a dodatak razrađenih i pouzdanih konvolucijskih metoda stvara impresivne rezultate.

Dokaz tome je i ovaj rad, koji nastoji naučiti jedan takav model kolorizacije na vrlo ograničenom skupu slika. Iako se rezultati mogu poboljšati raznim spomenutim metodama, pokazalo se da su transformerski modeli, a pogotovo DDColor, više nego sposobni uhvatiti se u koštac s problemom kolorizacije slika. Ovaj se model pokazao sposoban naučiti vrlo specifičan i apstraktan stil kolorizacije Gustava Klimta što duguje svojim brojnim unapređenjima u odnosu na prijašnje pristupe problemu kolorizacije.

Literatura

[1] Wikipedia suradnici, Hand-colouring of photographs, Wikipedia, (2024, veljača). Poveznica: https://en.wikipedia.org/wiki/Hand-colouring_of_photographs; pristupljeno 4. veljače 2024.

[2] Žeger, I., Grgic, S., Vuković, J., Šišul, G., "Grayscale Image Colorization Methods: Overview and Evaluation,", IEEE Access, vol. 9, pp. 113326-113346, 2021.

[3] Levin, A., Lischinski, D., Weiss, Y., Colorization using Optimization, ACM Transactions on Graphics, vol. 23, pp. 689-694, 2004.

[4] Irony, R., Cohen-Or, D., Lischinski, D., Colorization by Example, Eurographics Symposium on Rendering (2005), pp. 201-210, 2005.

[5] Cheng, Z., Yang, Q., Sheng, B., Deep Colorization, arXiv, 2016.

[6] Zhang, R., Zhu, J., Isola, P., Geng, X., Lin, A.S., Yu, T., Efros, A.A., Real-Time User-Guided Image Colorization with Learned Deep Priors, arXiv, 2017.

[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin I., Attention Is All You Need, arXiv, 2017.

[8] Bahdanau, D., Cho, K., Bengio, Y., Neural Machine Translation by Jointly Learning to Align and Translate, arXiv, 2014.

[9] Arjun Sarkar, All you need to know about 'Attention' and 'Transformers' - In-depth Understanding - Part 1, Towards Data Science, (2022, veljača). Poveznica: https://towardsdatascience.com/all-you-need-to-know-about-attention-and-transformers-in-depth-understanding-part-1-552f0b41d021; pristupljeno 4. veljače 2024.

[10] Sathya Krishnan Suresh, Understanding Transformers - Encoder, Medium, (2022, listopad). Poveznica: https://medium.com/mlearning-ai/understanding-transformers-encoder-1f269b1cc943; pristupljeno 5. veljače 2024.

[11] Ketan Doshi, Transformers Explained Visually (Part 3): Multi-head Attention, deep dive, Towards Dana Science, (2021, siječanj). Poveznica: https://towardsdatascience.com/transformers-explained-visually-part-3-multi-head-attention-deep-dive-1c1ff1024853; pristupljeno 5. veljače 2024.

[12] Mohamed Nabil, Unpacking the Query, Key, and Value of Transformers: An Analogy to Database Operations, LinkedIn, (2023, travanj). Poveznica: https://www.linkedin.com/pulse/unpacking-query-key-value-transformers-analogy-database-mohamed-nabil/; pristupljeno 6. veljače 2024.

[13] Umar Jamil, Attention is all you need (Transformer) – Model explanation (including math), Inference and Training, YouTube, (2023, svibanj). Poveznica: https://www.youtube.com/watch?v=bCz4OMemCcA; pristupljeno 6. veljače 2024.

[14] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, arXiv, 2020.

[15] Kumar, M., Weissenborn, D., Kalchbrenner N., Colorization Transformer, arXiv, 2021.

[16] Ho, J., Kalchbrenner, N., Weissenborn, D., Salimans, T., Axial Attention in Multidimensional Transformers, arXiv, 2019.

[17] Kang, X., Yang, T., Ouyang, W., Ren, P., Li, L., Xie, X., DDColor: Towards Photo-Realistic Image Colorization via Dual Decoders, arXiv, 2022.

[18] Wikipedia suradnici, CIELAB color space, Wikipedia, (2024, veljača). Poveznica: https://en.wikipedia.org/wiki/CIELAB_color_space; pristupljeno 7. veljače 2024.

[19] Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T., Xie, S., A ConvNet for the 2020s, arXiv, 2022.

[20] W. Shi, J. Caballero, F. Huszár, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, Z. Wang, Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network, arXiv, 2016.

[21] Kim, G., Kang, K., Kim, S., Lee, H., Kim, S., Kim, J., Baek, J.K., Cho, S., BigColor: Colorization using a Generative Color Prior for Natural Images, arXiv, 2022.

[22] Simonyan, K., Zisserman, A., Very Deep Convolutional Networks for Large-Scale Image Recognition, arXiv, 2014.

[23] Hasler, D., Suesstrunk, S., Measuring colorfulness in natural images, Proceedings of SPIE – The International Society for Optical Engineering (2003), vol. 5007, pp. 87-95, 2003.

[24] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S., GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium, arXiv, 2017.

[25] Wikipedia suradnici, Gustav Klimt, Wikipedia, (2024, veljača). Poveznica: https://en.wikipedia.org/wiki/Gustav_klimt; pristupljeno 8. veljače 2024.

[26] Tan, W.R., Chan, C.S., Aguirre, H., Tanaka, K., Improved ArtGAN for Conditional Synthesis of Natural Image and Artwork, arXiv, 2017.

[27] Wikipedia suradnici, Art Nouveau, Wikipedia, (2024, veljača). Poveznica: https://en.wikipedia.org/wiki/Art_Nouveau; pristupljeno 8. veljače 2024.

[28] Wikipedia suradnici, Vienna Secession, Wikipedia, (2024, veljača). Poveznica: https://en.wikipedia.org/wiki/Vienna_Secession; pristupljeno 8. veljače 2024.

Sažetak

Predstavljen je problem kolorizacije slika te razvoj metoda kolorizacije počevši od ručnih metoda pa sve do tehnika dubokog učenja. Tehnike dubokog učenja detaljnije su opisane s posebnim naglaskom na transformerske modele, od kojih je jedan odabran, objašnjen i istreniran s ciljem koloriziranja crno-bijelih slika u stilu slikara Gustava Klimta. Opisan je proces treniranja, postignuti rezultati, ograničenja i moguća poboljšanja.

Ključne riječi: kolorizacija, crno-bijele slike, duboko učenje, konvolucija, transformer, pozornost, Gustav Klimt

Summary

The problem of image colorization and the development of colorization methods, starting with manual methods and ending with deep learning techniques, are presented. Deep learning techniques are described in more detail with special emphasis on transformer models, one of which was selected, explained and trained with the aim of coloring black and white images in the style of the painter Gustav Klimt. The training process, achieved results, limitations and possible improvements are described.

Key words: colorization, grayscale images, deep learning, convolution, transformer, attention, Gustav Klimt