SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I

INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni diplomski studij

BOJANJE CRNO-BIJELIH SLIKA DUBOKIM UČENJEM

Obrada slika i računalni vid

Andreja Nađ

Osijek, 2022.

##### SADRŽAJ

[1. UVOD 1](#_Toc95047583)

[2. DIGITALNA SLIKE 2](#_Toc95047584)

[2.1. RGB prostor boja 2](#_Toc95047585)

[2.2. Crno-bijele slike 2](#_Toc95047586)

[2.3. Lab prostor boja 2](#_Toc95047587)

[3. DUBOKO UČENJE 3](#_Toc95047588)

[3.1. Konvolucijske neuronske mreže 3](#_Toc95047589)

[3.2. Generativna suparnička mreža 4](#_Toc95047590)

[4. KORIŠTENE TEHNOLOGIJE I ALATI 6](#_Toc95047591)

[4.1. Skup podataka 6](#_Toc95047592)

[4.2. Python 6](#_Toc95047593)

[4.3. Google Colab 6](#_Toc95047594)

[4.1. Potrebne biblioteke 6](#_Toc95047595)

[5. PROGRAMSKO RJEŠENJE 7](#_Toc95047596)

[5.1. Konvolucijska neuronska mreža 7](#_Toc95047597)

[5.1.1. Rezultati 11](#_Toc95047598)

[5.2. Generativna suparnička mreža 12](#_Toc95047599)

[5.2.1. Rezultati 14](#_Toc95047600)

[ZAKLJUČAK 16](#_Toc95047601)

[LITERATURA 17](#_Toc95047602)

# UVOD

Cilj ovog projekta je korištenjem TensorFlow-a izraditi model duboke neuronske mreže za bojanje crno-bijelih slika koristeći Tensorflow. Rezultate mreže prikazati na crno-bijelim slikama uz prikaz slika u boji. Potrebno je objasniti osnovne definicije i prikazati rezultate. Izvedena su dva algoritma koji rade putem neuronskih mreža, a to su: CNN model i GAN model koji će kasnije biti objašnjeni i uspoređeni.

## Strojno učenje

Strojno učenje je područje računalne znanosti koje je nastalo iz raspoznavanja uzoraka i računalne teorije učenja u području umjetne inteligencije. Strojno učenje je zapravo programiranje računala na način da optimiziraju neki kriterij uspješnosti temeljem podatkovnih primjera ili prethodnog iskustva [1]. Algoritmi strojnog učenja izgrađuju model preko kojeg se izvode predikcije ili odluke na temelju podataka bez potrebe izvođenja statičkih programskih instrukcija.

Vrste strojnog učenja su: nadzirano učenje, nenadzirano učenje i podržano učenje. Nadzirano učenje je vrsta strojnog učenja gdje je cilj odrediti nepoznatu funkcionalnu ovisnost između ulaznih veličina i izlazne veličine na temelju podatkovnih primjera. Pri tome su podatkovni primjeri parovi koji se sastoji od vektora ulaznih veličina i vrijednosti pripadne izlazne veličine. Postoje dvije osnovne namjene modela dobivenih nadziranim učenjem, a to su: predikcija i zaključivanje. Predikcija predstavlja proces u kojemu se na temelju modela i vektora ulaznih veličina može procjeniti vrijednost izlazne veličine. Predikcija je obično glavna svrha modela u problemima strojnog učenja. Zaključivanje predstavlja proces u kojemu se temeljem modela pokušava saznati više o procesu generiranja podataka. Zaključivanje je češće u području statističkog učenja.

Nadzirano učenje možemo podijeliti na dvije vrste problema, a to su regresija i klasifikacija. Ako je izlazna veličina iz skupa realnih brojeva, tj. ako je kontinuirana, tada se ovaj problem naziva regresija. Primjeri regresije su: procjena cijena kuća, procjena potrošnje plinova i sl. Ako je izlazna veličina diskretna, onda se ovakav problem naziva klasifikacija. Primjeri klasifikacije su: prepoznavanje lica, prepoznavanje pisanih brojeva, medicinska dijagnostika i sl. U ovom projektu potrebno je koristiti regresiju.

## Regresija

Cilj regresije je pronaći aproksimacijsku funkciju koja minimizira empirijsku pogrešku.

Regresija je problem nadgledanog učenja gdje je cilj odrediti nepoznatu funkcionalnu ovisnost između ulaznih veličina i izlazne veličine na temeljnu podatkovnih primjera. Formula za izlaznu veličinu se pretpostavlja kao zbroj nepoznate funkcionalne ovisnotsi i pogreške koja je nezavisna od ulaznih podataka i treba imati srednju vrijednost 0. Tada je cilj nadgledanog učenja da odredi aproksimaciju te nepoznate funkcionalne ovisnosti koja se onda koristi za predikciju izlazne veličine za novi uzorak ulaznih veličina. Dobivena aproksimacija se naziva i modelom, a on se može definirati kao funkcija s konačnim brojem parametara koji se određuju tijekom učenja.

# DIGITALNA SLIKE

Digitalna slika je slika sastavljena od piksela, od kojih svaki ima konačnu, diskretnu količinu numeričkog prikaza za svoj intenzitet ili razinu sive koja je rezultat njezinih dvodimenzionalnih funkcija spojenih kao ulaz pomoću prostornih koordinata.

## RGB prostor boja

Svaki piksel slike je označen kao brojčana vrijednost od 0 do 255. Svaki piksel ima tri vrijednosti koje predstavljaju tri kanala RGB prostora boja koji su: crveni, zeleni i plavi. To znači da se jedna slika sastoji od tri matrice od koje svaka predstavlja sve vrijednosti piksela jednog kanala.

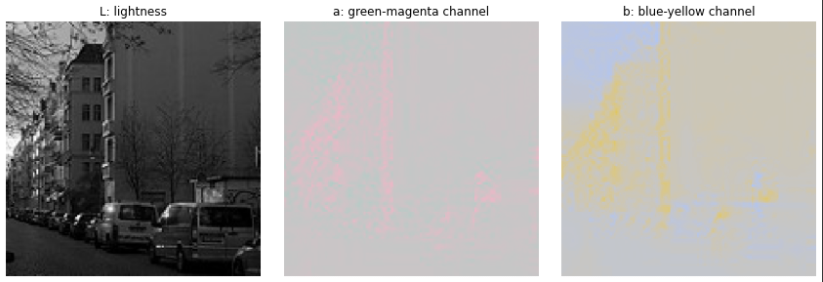
## Crno-bijele slike

Za slike u crno-bijelim nijansama (nijanse sive), vrijednosti piksela se nalaze u rasponu od 0 do 255. Manji brojevi bliže nuli predstavljaju tamniju nijansu, dok veći brojevi bliže 255 predstavljaju svijetliju ili bijelu nijansu. Svaka slika u računalu je spremljena u ovakvom obliku kao matrica brojeva koja je poznata kao i L („lightness“) kanal koji predstavlja svijetlinu slike.

## Lab prostor boja

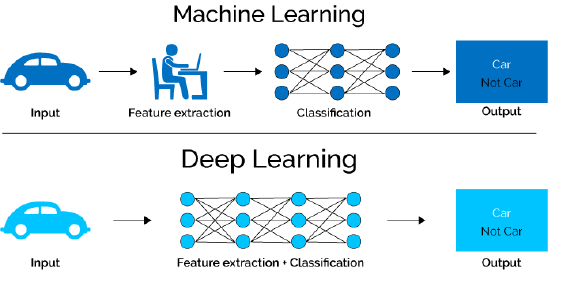
Slično RGB prostoru boja, Lab prostor boja ima tri kanala, no za razliku od RGB prostora boja, Lab drugačije kodira informacije boja. L kanal kodira samo intenzitet svjetlosti što bi bila crno-bijela slika, kanal 'a' kodira zeleno-crveno i 'b' kanal kodira plavo-žuto. Sve boje koje se nalaze između (uključujući) magente i zelene boje se nalaze u 'a' kanalu, dok sve boje između (uključujući) žute i plave se nalaze u 'b' kanalu.

U projektnom zadatku potrebno je za ulaznu crno-bijelu sliku tj. L kanal, vratiti RGB sliku. Koristeći Lab prostor boja, za poznati L kanal, neuronska mreža mora predvidjeti a i b kanale.



# DUBOKO UČENJE

Duboko učenje je dio strojnog učenja temeljno na umjetnim neuronskim mrežama s učenjem značajki. Učenje značajki je skup tehnika koje omogućuju sustavu automatsko otkrivanje značajki potrebnih za otkrivanje ili klasifikaciju značajki iz neobrađenih podataka. Slika XXXX prikazuje razliku strojnog učenja i dubokog učenja. Kod strojnog učenja je izvdajanje značajki bilo potrebno napraviti samostalno te se tek onda događa klasifikacija, no duboko učenje je to sve automatiziralo.



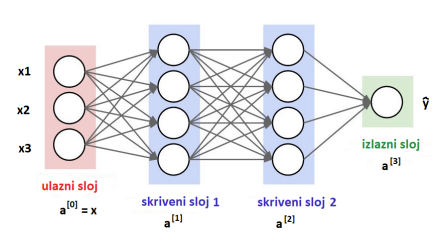
Duboko učenje je postalo korisnije kako se povećala količina dostupnih podataka za obuku. S vremenom, modeli dubokog učenja su postali boli kako se poboljšala programska i sklopovska infrastruktura za duboko učenje. Isto tako, duboko učenje rješava sve kompliciranije primjene sa sve većom preciznošću tijekom vremena. Ta preciznost rezultata u nekim slučajevima je ili usporediva ili čak nadmašuje rezultate koje mogu postići ljudi.

Duboko učenje koristi se u rješavanju mnogih problema kao što su:

* robotika
* umjetna inteligencija u računalnim igrama
* obrada prirodnog jezika
* obrada slike i računalni vid – detekcija objekata, uklanjanje pozadine slike...
* autonomna vožnja – detekcija objekata, upravljanje vozilom, praćenje stanja vozača.

## Konvolucijske neuronske mreže

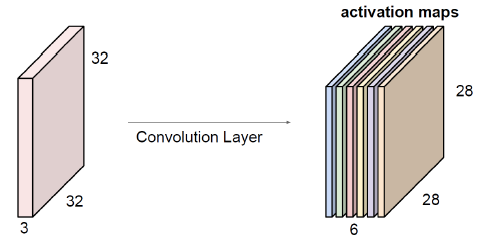
Konvolucijske neuronske mreže ili CNN se sastoje od neurona koji se samo-optimiziraju kroz učenje. Svaki neuron prima ulaz i izvodi operaciju. Za razliku od umjetne neuronske mreže, konvolucijske neuronske mreže lakše provode složene operacije potrebne za izračunavanje slikovnih podataka većih dimenzija.



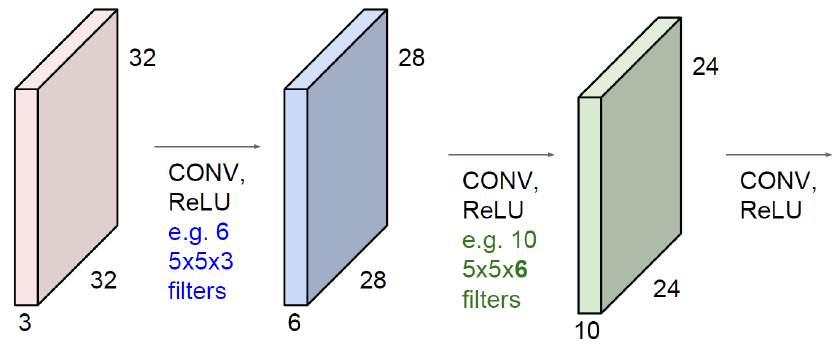
Slika 1. Konvolucijska neuronska mreža

Osnova konvolucijskih neuronskih mreža je konvolucijski sloj čijom primjenom se zadržava prostorna struktura ulaznih podataka. Svaki neuron povezan je samo s malim područjem u ulaznoj slici ili u prethodnom sloju. Primjena filtara na određenom dijelu slike rezultira u skalarnoj vrijednosti. Filtar pokriva malo prostorno područje, ali se prostire preko sva tri kanala ulazne slike. Težine filtra i bias mogu se podešavati. Oni zapravo predstavljaju parametre mreže koji se mijenjaju tijekom učenja.

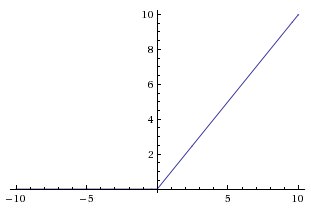
Primjenom istog filtra na različite pozicije dobiva se aktivacijska mapa što je dvodimenzionalna matrica koja sadrži odziv filtra na pojedinom dijelu ulazne slike. Svi neuroni iste aktivacijske mape imaju zajedničke parametre te je moguće primjeniti više filtara. Time se dobivaju zasebne aktivacijske mape koje se slažu kedna pored druge.



CNN temelji se na sekvenci slojeva između kojih se nalaze aktivacijske funkcije od kojih je najčešća ReLU funkcija. Sekvencijalnom se prijmenom konvolucijskih filtara gubi prostorna veličina kao što vidimo na slici XXX (32 🡪 28 🡪 24).

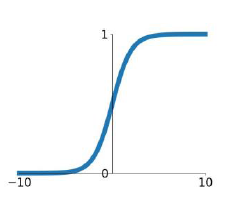
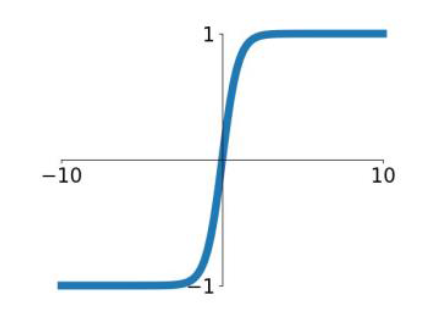


ReLU (Rectified Linear Unit) je aktivacijska funkcija ispravljača koja se koristi u aktivacijskom sloju CNN-a. ReLU je računalno efikasna i dobro oponaša biološku neuronsku aktivnost.

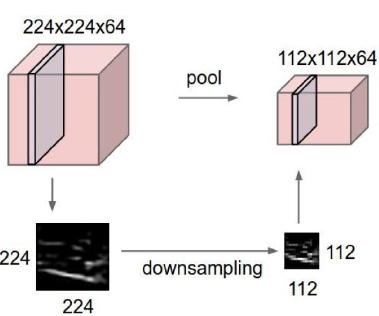


Slika 2. Funkcija ispravljača

Sigmoid aktivacijska funkcija ima izlaz u rasponu od 0 do 1 te je među popularnijim za korištenje, no problem nastaje kada je neuron u zasićenju jer tada ima gradijent 0. Tanh aktivacijska funkcija koja je korištena u modelu generatora je slična, no ona je centrirana oko 0. Čak i sa tim centriranjem, postoji problem da nestaje gradijent kada je neuron u zasićenju. Za razliku od njih, ReLU nema problema s nestajanjem gradijenta i računalno je efikasno. Brži je u konvergiranju nego sigmoidna ili tanh funkcija. No problem je što svejedno ima gradijent 0 kada je u negativnom dijelu.

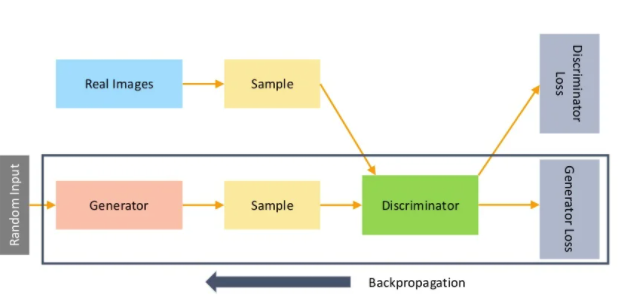
Kako bi se reprezentacija slike držala što manjom, uvode se slojevi sažimanja (eng. *pooling* *layers*). Ovaj sloj se primjenjuje na svaku aktivacijsku mapu zasebno i nema parametre koji se određuju postupkom učenja. Ovime se provodi smanjivanje rezolucije pojedine aktivacijske mape.



Na kraju postoji izlazni sloj u kojemu se koristi neuron s linearnom aktivacijskom funkcijom.

## Generativna suparnička mreža

Generativna suparnička mreža ili GAN (Generatice adversarial network) je obrazac za strojno učenje kojeg su dizanirali Ian Goodfellow i njegovi kolege u lipnju 2014. godine. GAN se sastoji od dvije neuronske mreže koje se međusobno natječu u analizi, hvatanju i kopiranju varijacija unutar skupa podataka. Te dvije mreže su diskriminator i generator. Generator uči stvarati lažne podatke uključivanjem povratnih informacija od diskriminatora koji identificira stvarne podatke od lažnih podataka koje stvara generator. Generator uči distribuciju podataka i osposobljen je za povećanje vjerojatnosti da će mreža diskriminatora napraviti pogreške. Diskriminator procjenjuje vjerojatnost da je uzorak koji je dobio iz podataka o obuci, a ne iz generatora. Prema tome, koraci za obuku GAN-a su: definirati problem, odabrati arhitekturu GAN-a, trenirati diskriminator na stvarnim podacima, generirati lažne podatke za generator, trenirati diskriminator na lažnim podacima te generator trenirati sa izlazom diskriminatora



# KORIŠTENE TEHNOLOGIJE I ALATI

## Skup podataka

Skup podataka preuzet je sa Kaggle stranice koja omogućuje korisnicima da pronađu i objavljuju skupove podataka, istražuju i grade modele u okruženju podatkovne znanosti temeljenom na webu. Skup podataka sastoji se od preko sedam tisuća krajolika u boji i njihovih kopija bez boje.

## Python

Korišteni programski jezik je Python, jedan od najpopularnijih programskih jezika. Python je interpretirani programski jezik opće namjene visoke razine čiji je cilj pomoći programerima da napišu jasan, logičan kod za projekte bilo koje veličine. Glavna stavka ovog programskog jezika je jednostavnost i čitljivost koda uz korištenje značajnog uvlačenja.

## Google Colab

Google Colaboratory je besplatno Jupyter okruženje koje koristi Python programerski jezik. Projekt Jupyter je projekt i zajednica čiji je cilj "razviti softver otvorenog koda, otvorene standarde i usluge za interaktivno računanje u desecima programskih jezika". Colab svakome omogućuje pisanje i izvršavanje proizvoljnog python koda putem preglednika, a posebno je prikladan za strojno učenje, analizu podataka i obrazovanje. Tehnički gledano, Colab je temeljen na Jupyter usluzi prijenosnih računala koja ne zahtijeva postavljanje za korištenje, a pruža besplatan pristup računalnim resursima uključujući GPU-ove. Ovo je odlično rješenje ako korisnik nema dovoljno dobro sklopovlje za duboko učenje. Colab ne zahtjeva dodatno preuzimanje, instaliranje ili pokretanje bilo čega drugoga jer Google Colab ima to sve ugrađeno samo se treba pozvati biblioteka. Potrebni podaci se mogu uvesti sa Google Drive-a, Github-a i drugih omogućenih izvora. Skup podataka koji se sastoji od slika se može uvesti, istrenirati i evaluirati na Googleovim serverima na oblaku koristeći njihove grafičke procesorske jedinice pa tako izrada modela više nije ograničena.

Google Colab posjeduje mogućnost korištenja raznih Pzthon biblioteka, Kerasa i Tensorflow-a bez potrebe instaliranja, uz dostupnost, laku implementaciju i Google-ove GPU-ove kako bi se ubrzalo treniranje modela, Google Colab se pokazao kao idelno rješenje pravljenja modela strojnog učenja.

## Keras

Keras je najčešće korišteni okvir za duboko učenje. Izgrađen na vrhu TensorFlowa 2, Keras je okvir koji je jak u industriji koji se može proširiti na velike skupine GPU-a. Keras sadrži brojne implementacije najčešće korištenih stavki neuronske mreže kao što su slojevi, ciljevi, aktivacijske funkcije, optimizatori i niz alata koji olakšavaju rad sa slikovnim i tekstualnim podacima kako bi se pojednostavilo kodiranje potrebno za pisanje koda duboke neuronske mreže. Uz standardne neuronske mreže, Keras ima podršku za konvolucijske neuronske mreže. Podržava i druge uobičajene slojeve kao što su izbacivanje, normalizacija serije i sažimanje.

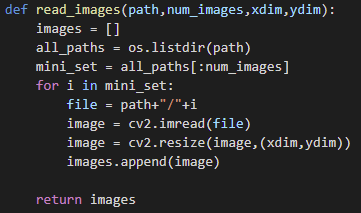
## TensorFlow

TensorFlow je besplatna softverska knjižnica otvorenog koda za strojno učenje i umjetnu inteligenciju. Može se koristiti u nizu zadataka, ali ima poseban fokus na obučavanju i zaključivanju dubokih neuronskih mreža. TensorFlow ima fleksibilnu arhitekturu koja omogućuje jednostavnu implementaciju na različitim platformama (CPU, GPU, TPU) kao i stolnim računalima, mobitelima i sl. TensorFlow se može izvoditi na više CPU-a i GPU-ova. iZgradnja i učenje mreže zasniva se na „dataflow graphs“ tj. grafovi protoka podataka. U ovim grafovima, čvorovi predstavljaju jedinice računanja, a rubovi podatke koji su potrošeni ili proizvedeni izračunima. Naziv TensorFlow potječe od operacija koje takve neuronske mreže izvode na višedimenzionalnim nizovima podataka, koji se nazivaju tenzorima.

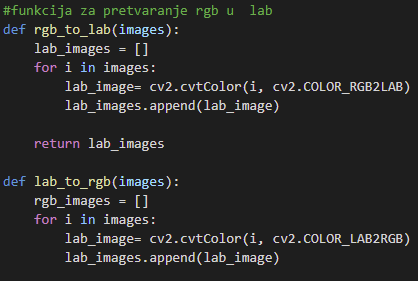
# PROGRAMSKO RJEŠENJE

## Konvolucijska neuronska mreža

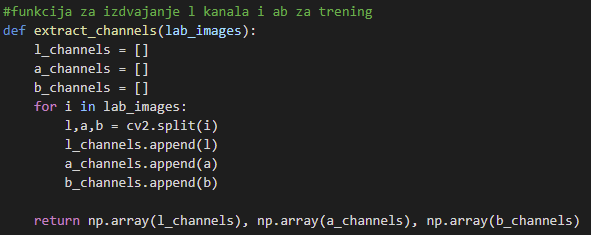
Funkcija *read\_images* učitava sve slike sa zadanog direktorija te ih postavlja sve u jednu listu.



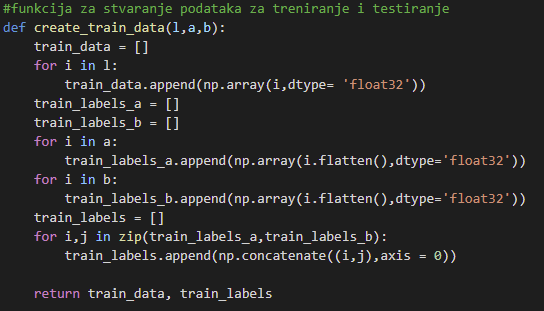
Za obradu slike potrebne su nam funkcije koje pretvraju slike iz RGB prostor boja u Lab prostor boja i obrnuto.



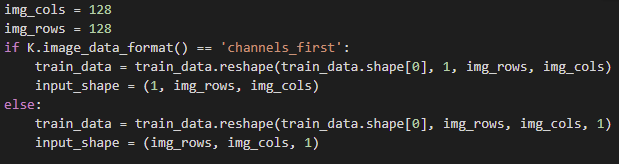
Lab prostor boja sastoji se od tri kanala od kojih je L kanal crno-bijela slika dok kanali a i b predstavljaju kanale boja stoga nam je potrebno odvojiti crno-bijele slike od boja za potrebe treninga mreže.



Funkcija *create\_train\_data* je funkcija koja uzima prethodne odvojene kanale Lab spektra te ih postavlja unutar dvije odvojene varijable. Ova ista funkcija može se koristiti za podatke za trening i testiranje.



Provjera ulaznog oblika podataka provjerava se uz funkciju image\_data\_format() gdje provjeravamo idu li kanali prvi ili zadnji. Po tome pravimo ulazni oblik koji se daje u pravljenje modela dubokog učenja.



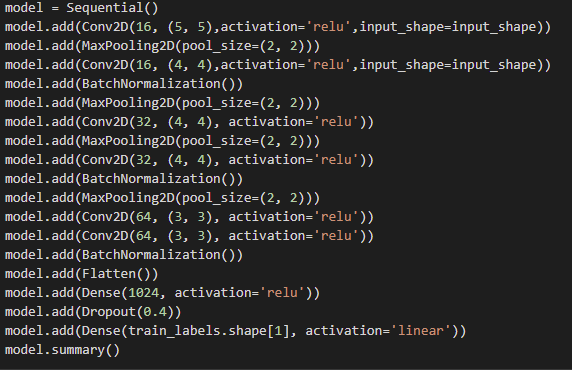
Sekvencijski model prikladan je za običan snop slojeva gdje svaki sloj ima točno jedan ulazni tenzor i jedan izlazni tenzor. U prvom konvolucijskom sloju učimo 16 filtara te se nakon toga u sloju maksimalnog sažimanja događa se smanjenje prostornih dimenzija izlaznog volumena. S ovim ponavljanjem, iako stavimo više filtara u konvolucijskim slojevima, naš izlazni prostorni volumen se smanjuje dok se broj naučenih filtara povećava.

Skupna normalizacija ili *BatchNormalization()* je funkcija koja standardizira ulaz u sloju neuronske mreže dubokog učenja. Jednom implementirana, skupna normalizacija ima učinak dramatičnog ubrzanja procesa treninga neuronske mreže, a u nekim slučajevima poboljšava performanse modela putem umjerenog učinka regularizacije. Sloj će transformirati ulaze tako da budu standardizirani, što znači da će imati srednju vrijednost od nule i standardnu ​​devijaciju od jedan. Tijekom treninga, sloj će pratiti statistiku za svaku ulaznu varijablu i koristiti je za standardizaciju podataka.

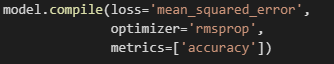
Funkcija *Flatten()* koristi se za dobivanje kopije danog niza sažete u jednu dimenziju. Gusti sloj ili *Dense layer* je sloj koji je duboko povezan sa svojim prethodnim slojem tj. neuron gustog sloja prima izlaz svakog neurona prethodnog sloja. Ovaj sloj je najčešće korišten sloj u mrežama.

*Dropout* sloj nasumično izbacuje neurone tijekom treniranja kako bi se smanjila mogućnost prekomjernog prilagođavanja trening podatcima.

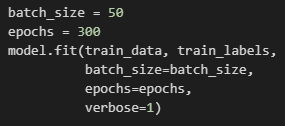
Izlazni sloj je linearna funkcija.



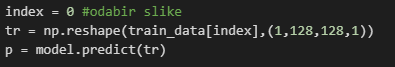
Nakon što smo završili s postavljanjem modela, potrebno ga je kompajlirati što znači definiranje funkcije gubitka, optimizatora i metriku.



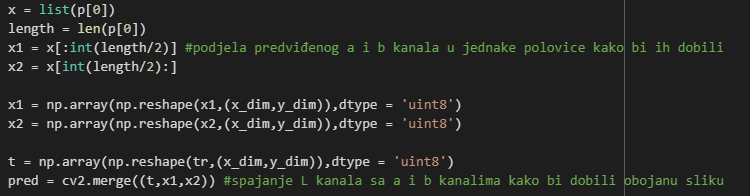
Kada je cijeli model definiran potrebno ga je trenirati. Treniranje ovisi o broju epoha tj. iteracija na skupu podataka i ukupnom broju koraka koji se događaju tijekom trajanja jedne epohe. Broj koraka svake epohe inače se gleda kao ukupan broj ulaznih podataka podjeljen sa brojem uzoraka što je u našem slučaju 7000/50 = 140.



Istrenirani model potrebno je samo pozvati s funkcijom *predict* kako bi model vratio izlaznu vrijednost.

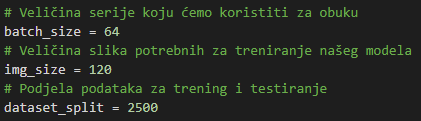


Tu izlaznu vrijednost tada trebamo podjeliti na jednake polovice za a i b kanal kojima moramo ponovno namjestiti oblik kao i ulaznoj sivoj slici. Zbrojem njih tj. spajanjem njih, dobivamo obojenu sliku.

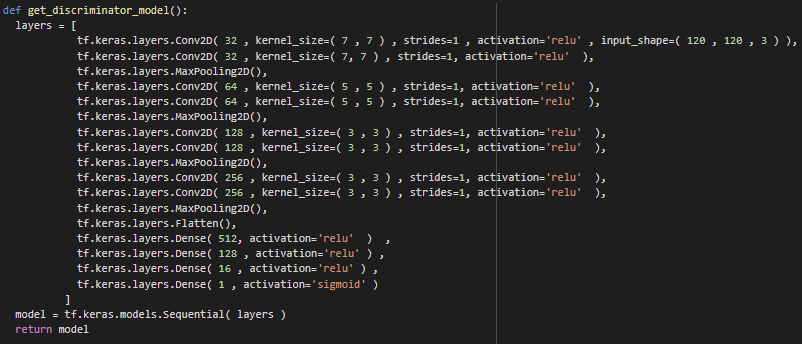


## Generativna suparnička mreža

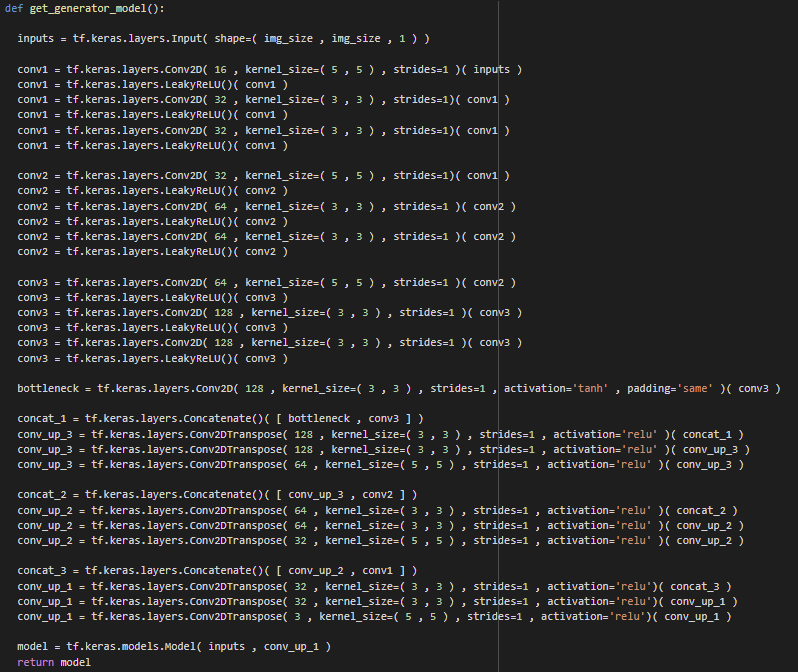
Prvo deklariramo potrebne varijable koje ćemo koristiti kasnije. Varijabla *dataset\_split* predstavlja broj ulaznih podataka.

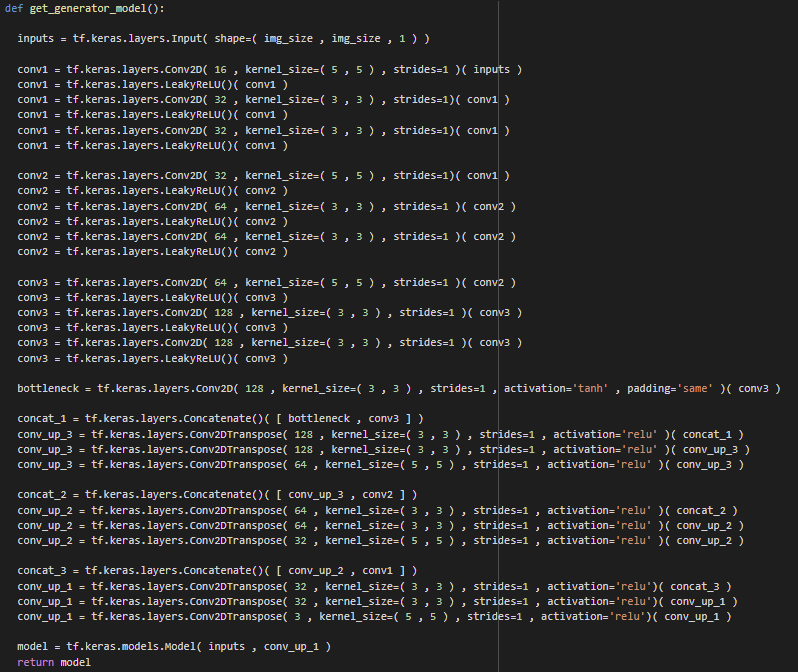


Model diskriminatora je sličan modelu CNN-a, ali izlazni sloj ima sigmoid funkciju.

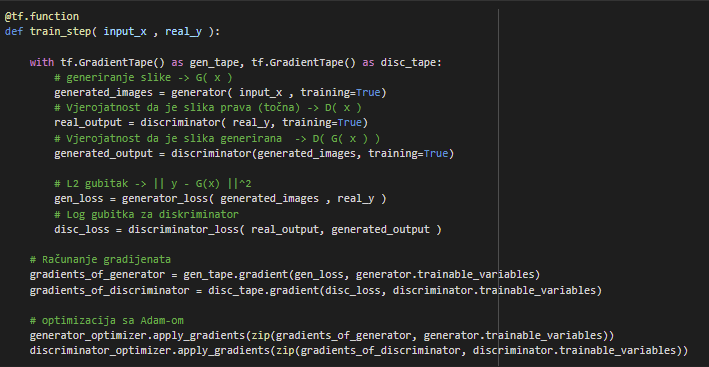


Model generatora mješavina je između konvolucijskog sloja i aktivacijskog sloja gdje se primjenjuje funkcija ReLU.





Treniranje podataka izvodi se tako da se predaje funkciji siva slika i njena RGB slika. Nakon toga događa se generiranje slike putem generatora za zadanu sivu sliku. Ta generirana slika i prava RGB slika testiraju se putem diskriminatora koji vraća vjerojatnost je li slika generirana ili ne.



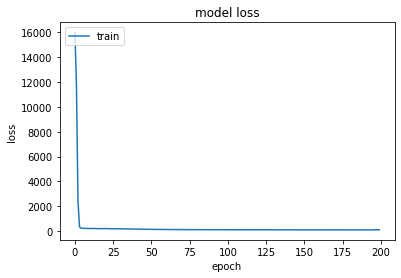
Testiranje podataka izvodimo tako da ih predamo istreniranom generatoru.



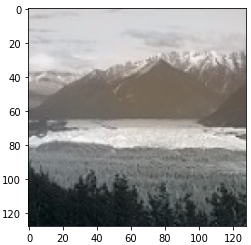
# REZULTATI

## Konvolucijska neuronska mreža

Očekivani gubitka pri treniranju je pravilno se smanji sa većim brojem epoha.

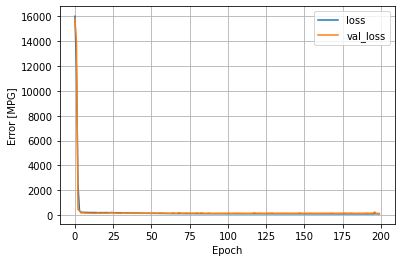


Rezultati nisu oni očekivani iako je testirano sa različitim parametrima modela. Potrebno je model trenirati sa puno više podataka i epohama kako bi se prikazalo pravilno rješenje.



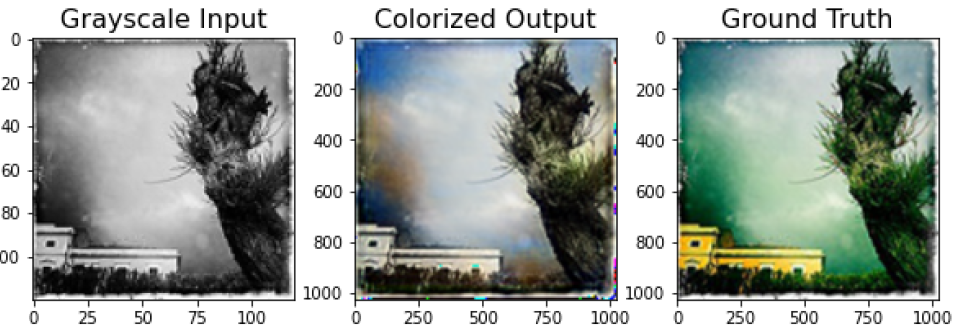
U ovom primjeru slika se zapravo vidi djelovanje modela gdje su boje točno pogođene ali ne na odgovarajućim mjestima.

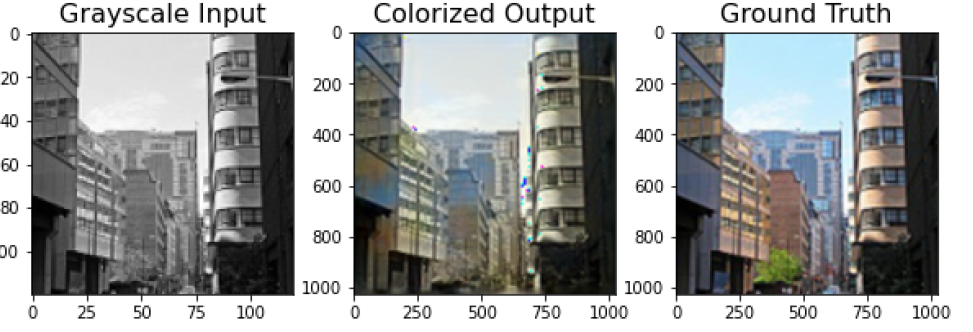




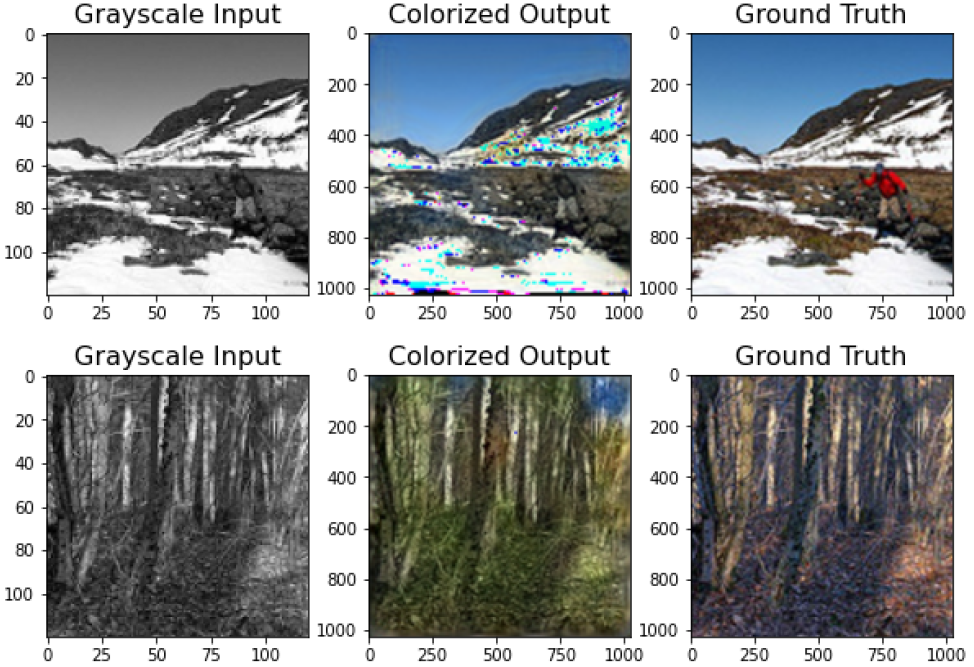
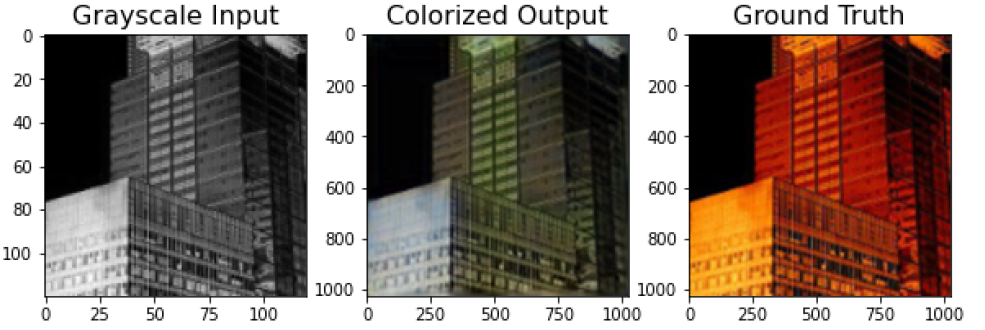
## Generativna suparnička mreža

Nažalost Google Colab nije mogao dovršiti učenje iznad 2500 ulaznih podataka zbog oduzimanja resursa. Iako je Google Colab besplatan, ima limit sa resursima koji nije vidljiv dok Colab ne javi da ste istrošili previše resursa. Naravno, resursi se vrate, no Google Colab ima razne načine odabira tko u kojem trenutku ima prednost s resursima. No, i sa tim manjim brojem podataka, dobili smo puno bolje rezultate nego u prethodnoj verziji.

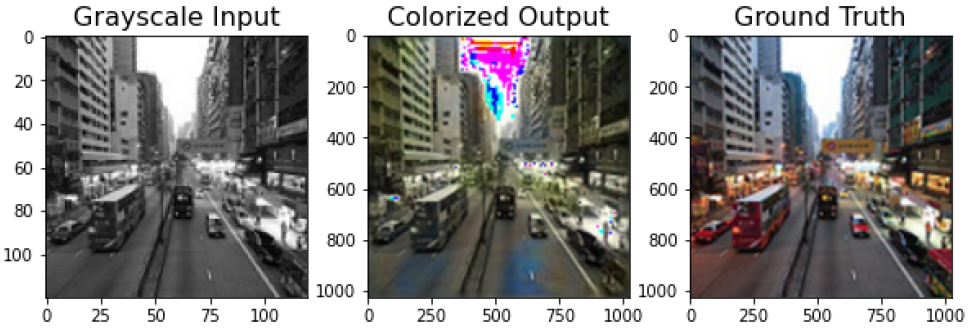
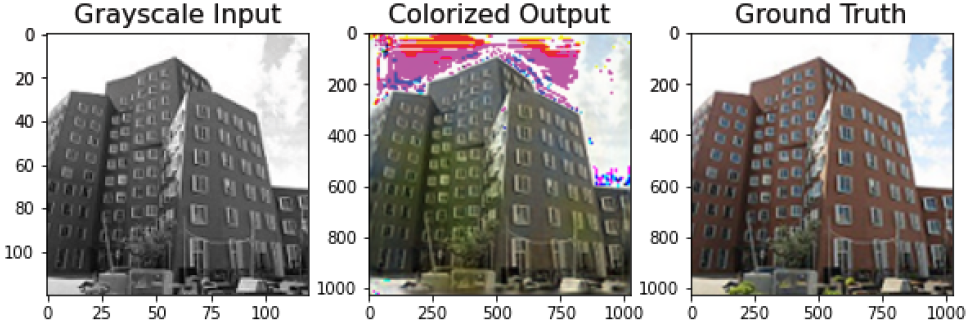




Slike prikazane dolje ne odgovaraju orginalima u boji, no ček izgledaju normalno.



Slike su vidljivo bolje nego kod obične CNN mreže. Zbog manjih podataka pri ulazu neke od slika imaju krive predviđene boje gdje bijela boja je predviđena u spektru između magente i plave.



#### ZAKLJUČAK

GAN model dao je bolje rezultate čak i za manji broj ulaznih podataka zbog svoje kompleksnosti tj. korištenja dvije mreže koje međusobno komuniciraju. GAN model isto tako daje i neke lošije rezultate koje smatram bi se mogli popraviti s većim brojem ulaznih podataka. Lakša verzija ovoga rada koja bi po mogućnosti bila točnija nego ova trenutna, bila bi kada bi se zadao jedan od a ili b kanala te prema tome bi CNN mreža predviđala samo jedan kanal. Isto tako postoji mogućnost da bi minimalne promjene pomogle u samom učenju kao što su možda veći konvolucijski filteri.

#### LITERATURA

[1] Tensorflow, https://www.tensorflow.org/

[2] Deep Learning, https://www.mathworks.com/discovery/deep-learning.html

[3] A. Tulenkov, Y. Yaramchenko, Anzhelika Parkhomenko, Y. Zalzubovskiy, Andriy Parkhomenko, M. Kalinina, Adaption of Smart House System for People with Special Needs Based on Wireless Technologies, Međunarodna konferencija o inteligentnom prikupljanju podataka i naprednim računalnim sustavima, Dortmund, Njemačka, rujan 2020. [30. lipnja 2021.]

[4] M. Li, W. Gu, W. Chen, Y. He, Y. Wu, Y. Zhang, Smart Home: Architecture, Technologies and Systems, Elsevier, 2018. [20. kolovoza 2021.]

[5] S. J. Darby, Smart technology in the home: time for some clarity, Building Research & Information, ožujak 2017. 7 [30. lipnja 2021.]

[6] S. Balakrishnan, H. Vasudavan, R. Kumar Murugesan, Smart Home Technologies: A Preliminary Review, Zbornik radova 6. međunarodne konferencije o informacijskim tehnologijama: IoT i pametni grad, prosinac, 2018. [20. kolovoza 2021.]

[7] D. Spivey, Home Automation for Dummies, John Wiley & Sons, New Jersey, 2015. [30. lipnja 2021.]

[8] K. Gill, S.-H. H. Yang, F. Yao, X. Lu, A ZigBee-Based Home Automation System, Transakcije o potrošačkoj elektronici, svibanj 2009. [30. lipnja 2021.]

[9] S. Park, S. Won, J. Lee, Smart home – digitally engineered domestic life, srpanj 2003. [30. lipnja 2021.]