

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 417

**WEB APLIKACIJA ZA AUTOMATSKO RASPOZNAVANJE
SLIKARA PREMA UMJETNIČKOM DJELU**

Ivana Stilinović

Zagreb, lipanj 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 417

**WEB APLIKACIJA ZA AUTOMATSKO RASPOZNAVANJE
SLIKARA PREMA UMJETNIČKOM DJELU**

Ivana Stilinović

Zagreb, lipanj 2022.

ZAVRŠNI ZADATAK br. 417

Pristupnica: **Ivana Stilinović (0036525591)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: izv. prof. dr. sc. Alan Jović

Zadatak: **Web aplikacija za automatsko raspoznavanje slikara prema umjetničkom djelu**

Opis zadatka:

Cilj završnog rada je napraviti web aplikaciju koja predviđa slikara prema predočenoj umjetnini (slici) korištenjem algoritama strojnog učenja. Aplikacija treba imati mogućnost učitavanja slike koju će onda predobraditi i korištenjem modela dubokog učenja korisniku dojaviti o kojem se slikaru radi i s kojom pouzdanosti je to utvrđeno. Korisnik će se trebati registrirati prije korištenja aplikacije. Model će se učiti na javno dostupnom skupu podataka s velikim brojem umjetnina raznih slikara. Usporedit će se nekoliko modela dubokog učenja s različitim vrijednostima hiperparametara te će se izabrati najbolji model za dani problem. Implementaciju je potrebno ostvariti u programskom jeziku Python.

Rok za predaju rada: 10. lipnja 2022.

Zahvaljujem mentoru izv. prof. dr. sc. Alanu Joviću na pomoći i podršci.

SADRŽAJ

1. Uvod i motivacija	1
1.1. Uvod	1
2. Podaci	2
2.1. Analiza skupa podataka	2
2.2. Priprema podataka	4
2.2.1. Rukovanje podacima	4
2.2.2. Podjela podataka	5
3. Umjetne neuronske mreže	6
3.1. Neuron	6
3.1.1. Biološki neuron	6
3.1.2. Umjetni neuron	6
3.1.3. Neuronske mreže	7
3.2. Arhitektura neuronske mreže	7
3.2.1. Ulazni sloj	8
3.2.2. Skriveni slojevi	8
3.3. Konvolucijske neuronske mreže	8
3.3.1. Konvolucijski sloj	9
3.3.2. Sloj sažimanja	9
3.3.3. Potpuno povezani sloj	9
4. Model	10
4.1. TensorFlow	10
4.2. Izgradnja modela od početka	10
4.2.1. Evaluacija modela	12
5. Izgradnja modela na temelju prethodno naučene mreže	14
5.1. ResNet50	14

5.2. Prijenosno učenje	14
5.3. Izgradnja modela	14
6. Izrada web aplikacije	16
6.1. Flask	16
6.2. Google Colab	16
6.3. Baza podataka	16
6.4. Opis rada aplikacije	17
7. Zaključak	20
Literatura	21

1. Uvod i motivacija

1.1. Uvod

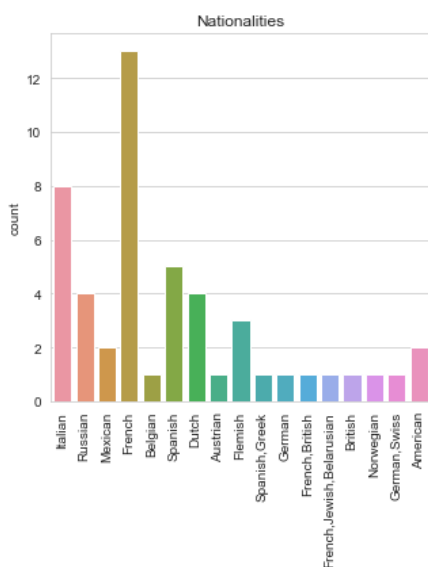
Umjetnost je vrlo subjektivna, svaka osoba sliku može interpretirati na svoj način. Postoji mnogo slikara u puno umjetničkih perioda kao što su: gotika, renesansa, klasicizam, impresionizam i mnogi drugi. U srednjoj školi redom se ide kroz povijest i uči o karakteristikama svakog perioda, pa tako i svakog slikara. Nakon četiri godine likovne umjetnosti i dalje imam problem razaznati o kojem se umjetniku radi. Cilj ove aplikacije je pomoći ljudima, pa tako i meni klasificirati umjetnička djela. Aplikacija će svojim korisnicima omogućiti prijenos slike. Pomoću dubokih neuronskih mreža korisnik će dobiti povratnu informaciju o kojem se umjetniku radi.

Duboko učenje pripada širokom području umjetne inteligencije. Glavni fokus dubokog učenja su neuronske mreže s puno skrivenih slojeva. Cilj ovog rada je upoznati se s dubokim učenjem i izradom modela. Model je algoritam koji pomoću skupa podataka prepoznaje određene obrasce.

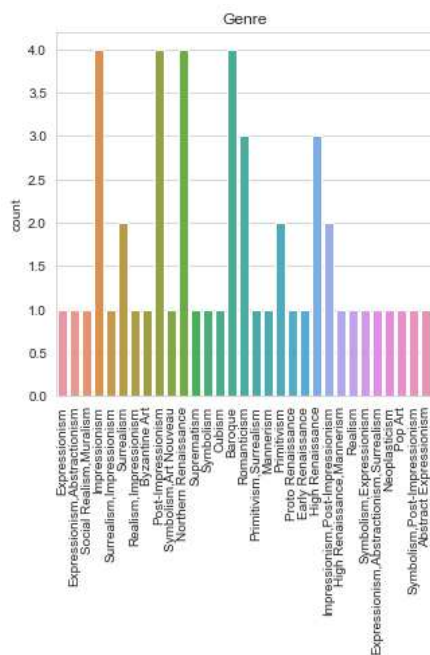
2. Podaci

2.1. Analiza skupa podataka

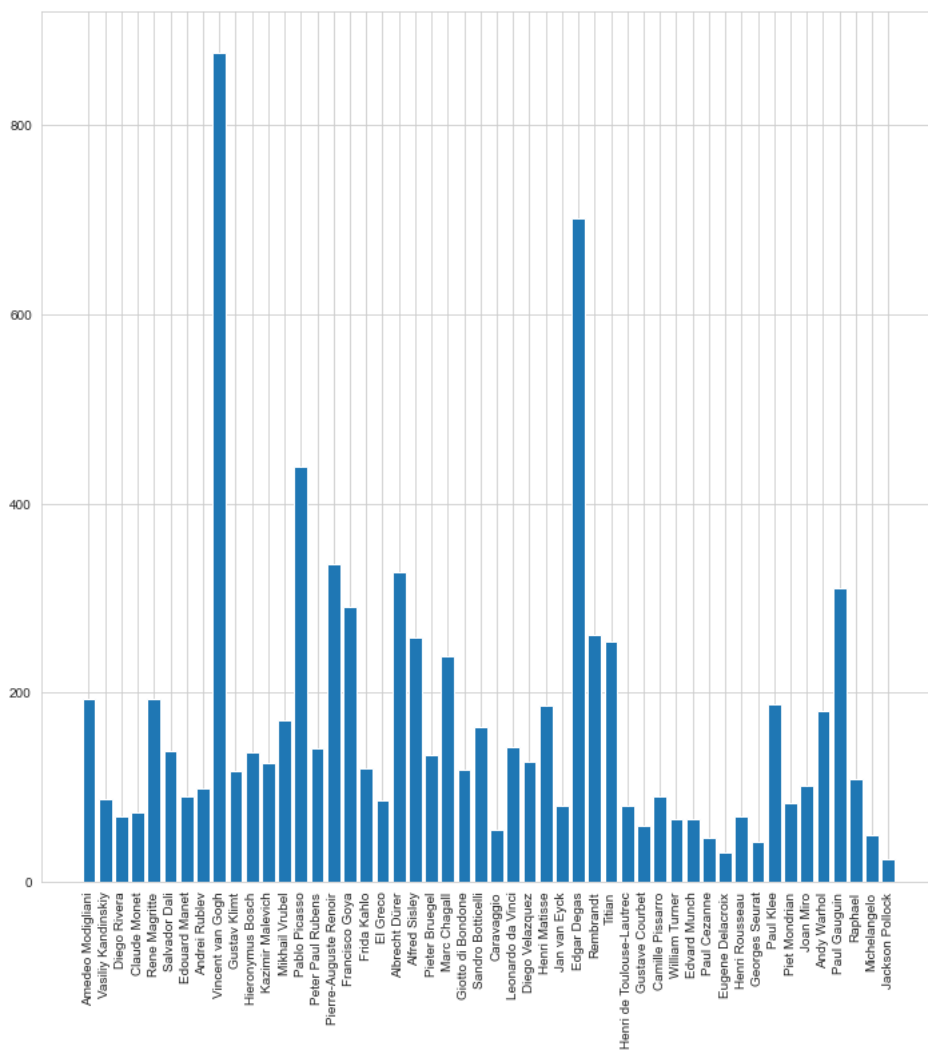
Podaci su preuzeti s interneta. Budući da nam podaci nisu poznati prvo ćemo napraviti njihovu analizu. Skup se sastoji od 50 umjetnika i ukupno 8466 slika. Svaki umjetnik ima osam značajki: id, name, years, genre, nationality, bio, wikipedia i paintings. Umjetnike možemo klasificirati po nacionalnosti (Slika 2.1) ili umjetničkom periodu u kojem su slikali (Slika 2.2). Napraviti ćemo stupčasti dijagram da vidimo koliko koji umjetnik ima slika u ovom skupu. (Slika 2.3)



Slika 2.1: Prema nacionalnosti



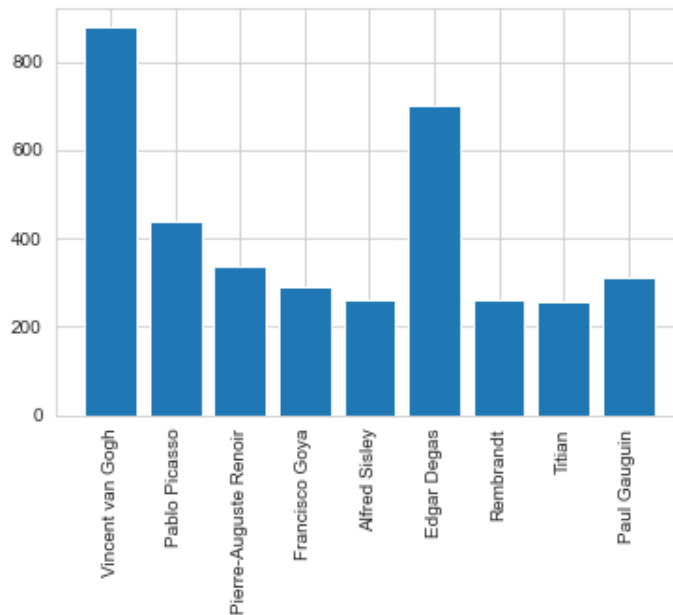
Slika 2.2: Prema žanru



Slika 2.3: Broj slika po slikaru, prikazani su svi slikari u skupu

2.2. Priprema podataka

Za naš daljnji rad uzet ćemo samo umjetnike koji imaju više od 250 slika. Takvih umjetnika ima devet. U podacima postoji problem s imenom slika i direktorija umjetnika Alberta Dürera zbog njemačkih dijakritičkih znakova, stoga ćemo njega izbaciti iz naših podataka kako bi olakšali daljnji rad. Na Slici 2.4 vidimo da su sada omjeri bolji, ali kako bi osigurali što veću točnost uvesti ćemo još jednu varijablu - *class weight*.



Slika 2.4: Slikari s više od 250 slika

Class weight pridružuje veću vrijednost umjetniku s manje slika u skupu, a manju onima s većom vrijednosti. Služi nam za ravnotežu podataka.

$$classWeight_j = nSamples / (nClasses * nSamples_j) \quad (2.1)$$

2.2.1. Rukovanje podacima

Točnost našeg modela ovisi o količini i kvalitete podataka. Poboljšanje podataka (engl. *Data augmentation*) su tehnike generiranja novih podataka iz postojećih kako bi se povećala njihova količina.[1] Traženje novih podataka i njihovo označavanje je skupo, a čak i sitne promjene na postojećim podacima poboljšat će naš model. Primjer originalne slike i njezine poboljšane verzije možemo vidjeti na Slici 2.6.

Originalna slika Pablo Picasso



Transformirana slika Pablo Picasso



Slika 2.5: Primjer poboljšanja podataka transformacijom

2.2.2. Podjela podataka

Podatke ćemo razdvojiti u dva skupa u omjeru 80:20. Model učimo na skupu za učenje, a zatim pomoću tog modela radimo predikciju na do tada neviđenom skupu za testiranje i na tom skupu računamo točnost.

3. Umjetne neuronske mreže

3.1. Neuron

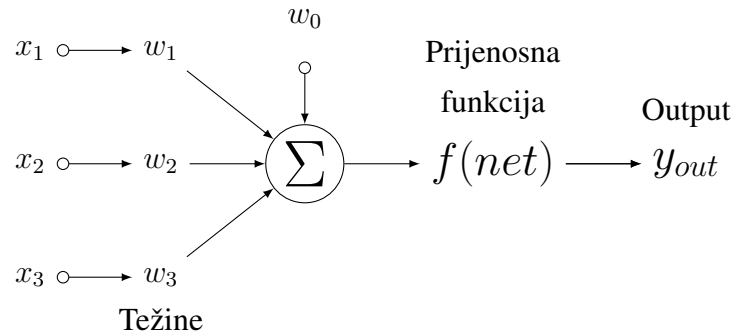
3.1.1. Biološki neuron

Neuron se smatra osnovnom jedinicom živčanog sustava i jedna je od najsloženijih jedinica u tijelu. Živčani se sustav sastoji od 86 milijardi međusobno povezanih neurona.[2] Oni međusobno komuniciraju putem specijaliziranih veza zvanih sinapse. Tipičan neuron sastoji se od tijela stanice (soma), dendrita, jednog aksona te završnih članaka.

Neuroni su električni podražljivi zbog održavanja gradijenta napona na njihovim membranama. Ako se napon dovoljno promijeni u nekom kratkom intervalu, neuron generira akcijski potencijal koji putuje duž aksona i aktivira sinaptičke veze kako do njih dođe. Akcijski potencijal mogu pokrenuti i kemijske i električne sinapse.

3.1.2. Umjetni neuron

Umjetni neuron matematička je funkcija osmišljena na temelju biološkog neurona. Ulogu akcijskog potencijala ovdje igra prijenosna funkcija f . Prijenosna funkcija je funkcija odluke, ako suma ulaza u neuronu pomnožena s težinom dođe do određene vrijednosti, specifične za tu funkciju, ona se propusti kroz prijenosnu granicu. Često korištene prijenosne funkcije su: Identitet (ADALINE-neuron), funkcija skoka (TLU-perceptron), sigmoidna funkcija, tangens hiperbolni, zglobnica (engl. Rectified Linear Unit, ReLU), propusna zglobnica (engl. Leaky Rectified Linear Unit, LReLU).[3]



Slika 3.1: Umjetni neuron

3.1.3. Neuronske mreže

Za svaki ulaz, množi se ulazna vrijednost x_i sa svojom težinom w_i i sumira se njihov umnožak. Ova težina govori koliko je jaka veza između dva neurona koja ih povezuje.[4] Ako je neka veza jača od druge, odnosno ako više utječe na izlaz neurona, bit će reprezentirana većom težinom.

$$\sum = (x_1 * w_1) + (x_2 * w_2) + \dots + (x_n * w_n) \quad (3.1)$$

Ulaz je zapravo vektor x , a težine vektor w pa je njihov umnožak zapravo skalarni produkt.

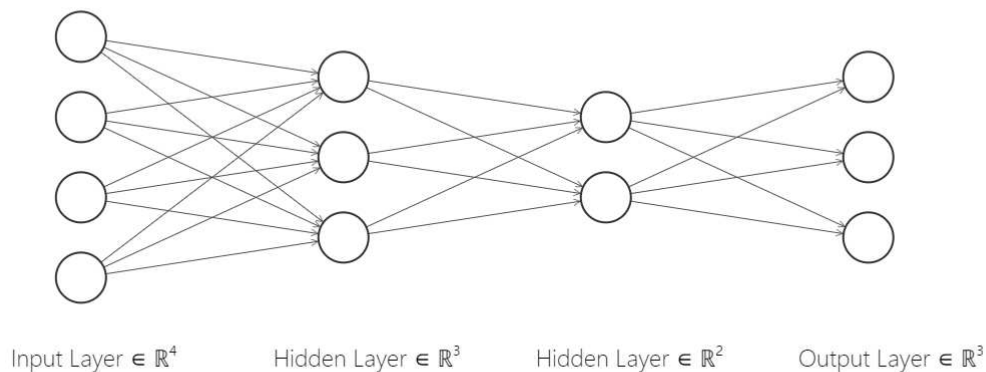
$$\sum = x * w \quad (3.2)$$

Ukupnoj sumi dodaje se i pomak (engl. *bias*) w_0 . Ta se vrijednost propušta kroz prijenosnu funkciju.

3.2. Arhitektura neuronske mreže

Umjetne neuronske mreže sastoje se od ulaznog sloja, jednog ili više skrivenih slojeva i izlaznog sloja (Slika 3.2). Arhitektura mreže govori kako su neuroni međusobno povezani i koliko ih ima.

Protok informacija odvija se na dva načina: mreže prema naprijed (engl. *feed-forward*), gdje signali putuju samo u jednom smjeru, prema izlazu i mreže s povratnom informacijom (engl. *recurrent*), gdje signali putuju u oba smjera kroz petlje u mreži.



Slika 3.2: Neuronska mreža, izrađeno pomoću alata [5]

3.2.1. Ulazni sloj

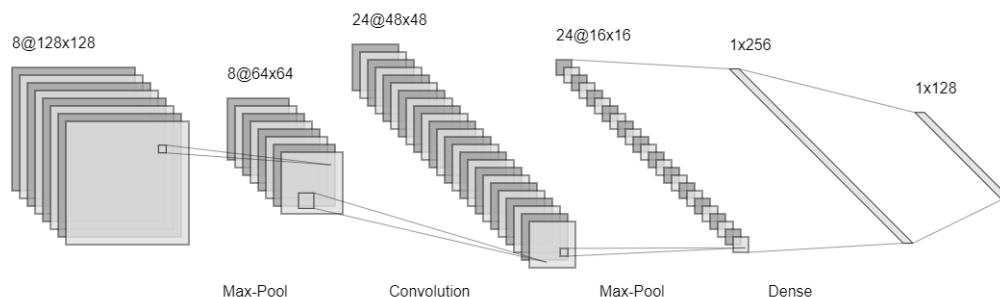
U našem primjeru ulazni sloj sadrži vrijednosti piksela slike. Svaki neuron prima ulaz od vanjskog izvora, povezan je s čvorovima sljedećeg sloja, a svaka veza ima svoju težinu. Kada se sve vrijednosti čvorova iz ulaznog sloja pomnože i zbroje, generira se vrijednost za prvi skriveni čvor. Slojevi imaju unaprijed definiranu prijenosnu funkciju, kojom se neuron aktivira ili ne na temelju ove vrijednosti.

3.2.2. Skriveni slojevi

Neuronska mreža može imati jedan ili više skrivenih slojeva, a tu se odvija glavno računanje. Često se ovi slojevi smatraju i opisuju kao crna kutija. Skriveni sloj uzima sve izlaze iz prethodnog sloja, radi potreban izračun te se on šalje sljedećem sloju.

3.3. Konvolucijske neuronske mreže

Konvolucijska neuronska mreža vrsta je umjetne neuronske mreže koja koristi matematičku operaciju *konvolucija* umjesto općeg množenja matrice u barem jednom od svojih slojeva. Najčešće se koriste za analizu vizualnih slika. Imaju tri glavne vrste slojeva: konvolucijski sloj, sloj sažimanja i potpuno povezani sloj (Slika 3.3). Slojevi blizu ulaza uče značajke niske razine (rubovi, linije), dok oni dublji slojevi uče apstraktnije značajke, poput oblika.



Slika 3.3: Konvolucijska neuronska mreža, izrađeno pomoću alata [5]

3.3.1. Konvolucijski sloj

U mreži može biti jedan ili više konvolucijskih slojeva. U konvolucijskom sloju izvodi se ranije spomenuta matematička operacija konvolucije - matematička kombinacija dviju funkcija, rezultat je dvodimenzionalni niz izlaznih vrijednosti - *mapa značajki*. Neuroni u istom sloju dijele težinu što povećava učinkovitost učenja smanjujući broj slobodnih parametara. Izlaz nam daje informacije o značajki kao što su kutovi i rubovi.[6]

3.3.2. Sloj sažimanja

Sloj sažimanja (engl. pooling layer) u većini slučajeva slijedi nakon konvolucijskog sloja. Ovaj sloj sažima mapu značajki tako da smanji dimenzije mape značajki, čime smanjuje količinu parametara potrebnih za učenje neuronske mreže.[7] Za to koristi dvodimenzionalni filter koji prelazi preko mape značajki s određenim pomakom. *Max - pooling* vrsta je sažimanja koja odabire najveći element koji trenutno vidi filter na mapi značajki. Još jedna vrsta je prosječni *pooling*.

3.3.3. Potpuno povezani sloj

Potpuno povezani sloj je tipična neuronska mreža s prijenosom unaprijed. Izlazi iz konvolucijskog sloja i sloja sažimanja predstavljaju značajke visoke razine. Pomoću tih značajki u potpuno povezanom sloju klasificiramo sliku.

4. Model

4.1. TensorFlow

TensorFlow je biblioteka otvorenog koda za strojno učenje. On olakšava proces učenja i izgradnje modela. Najveća prednost je apstrakcija, kod izgradnje modela najveći fokus je na logiku, dok TensorFlow u pozadini brine o detaljima.

Tensor u kontekstu TensorFlowa je vektor ili višedimenzionalna matrica koja predstavlja sve vrste podataka. Može nastati iz ulaznih podataka ili rezultata izračuna.[8]

4.2. Izgradnja modela od početka

Model je izgrađen kao konvolucijska neuronska mreža. Građen je pomoću Keras Sequential API koji omogućava umetanje sloja po sloj. Model se sastoji od slojeva konvolucije, slojeva sažimanja, potpuno povezanih slojeva i dva sloja koja nasumično postavljaju ulazne vrijednosti na nula kako bi se spriječilo prekomjerno prilagođivanje (engl. dropout).

2D konvolucijski sloj stvara jezgru koja se konvoluiru s ulaznim slojem kako bi se stvorio izlazni tenzor. Kao parametar mu se predaje broj izlaznih filtara, dimenzije prozora koji provodi konvoluciju, funkcija aktivacije i nadopuna koju smo ovdje postavili na *valid*.

Sloj maksimalnog sažimanja definiran je s *pool size* koji opisuje veličinu prozora i sa *strides* koji govori koliki je pomak tog prozora.

Sloj *Dropout* nasumično odabire neurone koji neće sudjelovati u učenju neuronske mreže, a predani argument je vjerojatnost odabira neurona.[10]

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 218, 222, 7)	448
conv2d_1 (Conv2D)	(None, 212, 221, 7)	693
max_pooling2d (MaxPooling2D)	(None, 106, 110, 7)	0
conv2d_2 (Conv2D)	(None, 102, 110, 5)	180
conv2d_3 (Conv2D)	(None, 98, 110, 5)	130
max_pooling2d_1 (MaxPooling2D)	(None, 49, 55, 5)	0
dropout (Dropout)	(None, 49, 55, 5)	0
conv2d_4 (Conv2D)	(None, 47, 55, 3)	48
conv2d_5 (Conv2D)	(None, 45, 55, 3)	30
max_pooling2d_2 (MaxPooling2D)	(None, 22, 27, 3)	0
dropout_1 (Dropout)	(None, 22, 27, 3)	0
flatten (Flatten)	(None, 1782)	0
dense (Dense)	(None, 256)	456448
dense_1 (Dense)	(None, 2048)	526336
dense_2 (Dense)	(None, 1024)	2098176
dense_3 (Dense)	(None, 9)	9225

=====
Total params: 3,091,714
Trainable params: 3,091,714
Non-trainable params: 0
=====

Slika 4.1: Slojevi modela

Potpuno povezani sloj implementira funkciju aktivacije, a drugi parametar su dimenzije izlaznog prostora.

Optimizer je algoritam koji mijenja značajke neuronske mreže kako bi smanjila gubitke. Optimizer *Adam* (engl. Adaptive Moment Estimation) temelji se na adaptivnoj procjeni momenta prvog i drugog reda s ciljem održavanja konstantne stope učenja.[9] *Stopa učenja* je parametar koji kontrolira koliko se model mijenja s obzirom na gradijent gubitka.

Kako bi poboljšali učenje mreže uveli smo objekt *ReduceLROnPlateau* koji smanjuje stopu učenja ako uoči da se promatrani parametar ne mijenja neko određeno vrijeme

4.2.1. Evaluacija modela

Funkcijom *model.evaluate* izračunat ćemo koliko dobro model radi na podacima za učenje, a koliko dobro na podacima za testiranje.

```
score = model.evaluate(train_set, verbose=1)
print("Prediction accuracy on train data =", score[1])

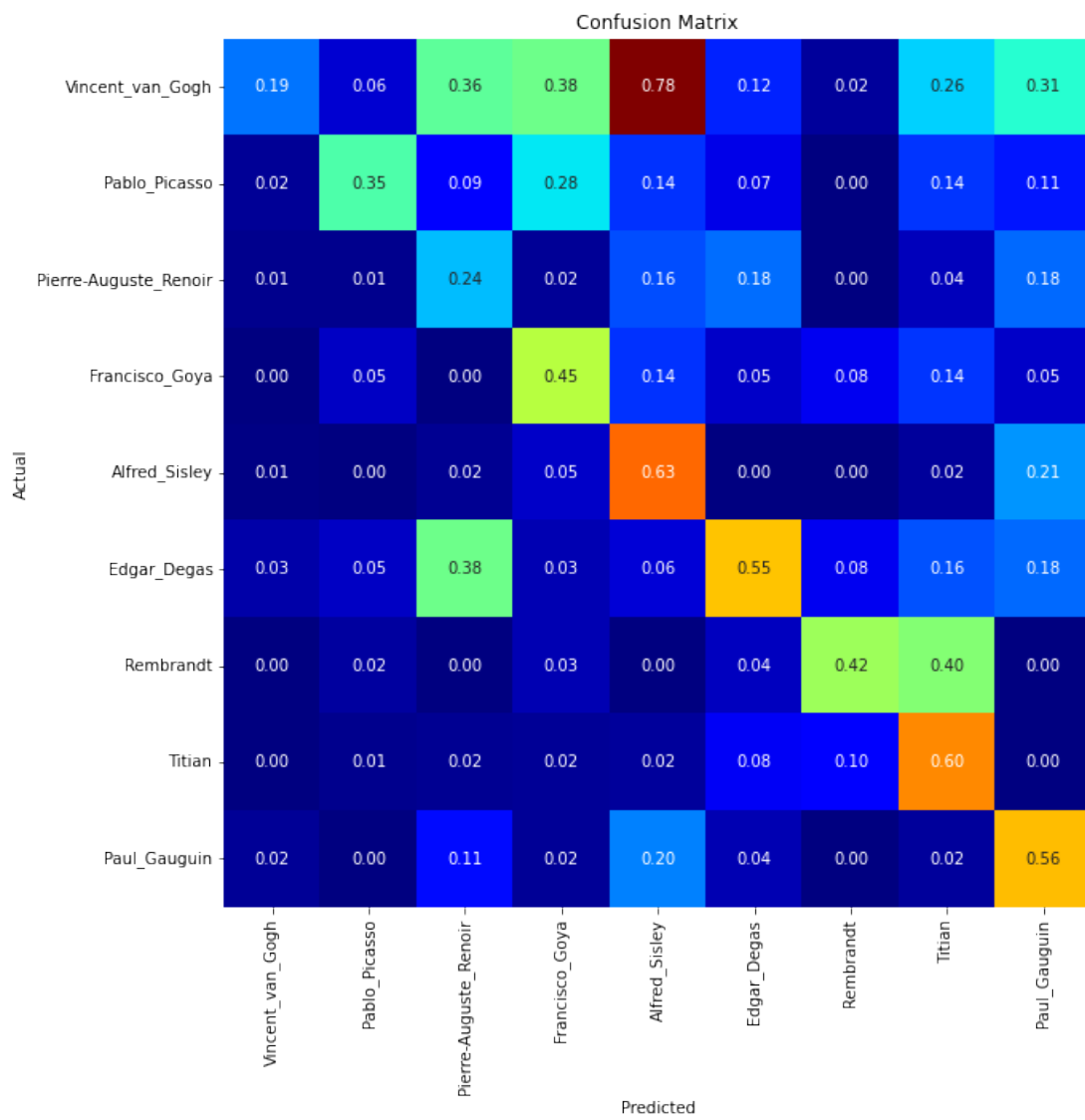
187/187 [=====] - 86s 462ms/step - loss: 1.5795 - accuracy: 0.4052
Prediction accuracy on train data = 0.40515223145484924

score = model.evaluate(test_set, verbose=1)
print("Prediction accuracy on test data =", score[1])

47/47 [=====] - 21s 444ms/step - loss: 1.5938 - accuracy: 0.3742
Prediction accuracy on test data = 0.3741588294506073
```

Slika 4.2: Evaluacija neuronske mreže

Matricom konfuzije vizualiziramo točno za svaku klasu kako ju naš model klasificira. Na osi x nalazi se stvarna klasa slike, a na osi y predviđena klasa. Što su veće vrijednosti po dijagonali, to je model bolji. Iz matrice možemo uočiti da model loše radi predikciju za slikara Vincenta van Gogha. Razlog tomu može biti problem kod implementacije težine klase koje smo u 2.2. *Priprema podataka* izračunali.



Slika 4.3: Matrica konfuzije

5. Izgradnja modela na temelju prethodno naučene mreže

5.1. ResNet50

Residual neural network (ResNet50) konvolucijska je neuronska mreža s 50 slojeva. Ovo je prethodno naučena neuronska mreža koja može klasificirati slike u 1000 kategorija objekata. Kao rezultat toga mreža je naučila bogate prikaze značajki za širok raspon slika.[11] Ulazna veličina slike je (224, 224). ResNet može preskakivati slojeve. To mu pomaže ublažiti problem nestajanja gradijenta, pojednostavljuje mrežu i tako ubrzava proces učenja.

5.2. Prijenosno učenje

Prijenosno učenje odnosi se na proces kada se model izgrađen za jedan problem koristi na nekom novom problemu. Ovakav pristup smanjuje vrijeme potrebno učenju neuronskih mreža za kompleksne probleme.

5.3. Izgradnja modela

Prethodno naučenoj mreži dodali smo par slojeva kako bi ju bolje pripremili za problem klasifikacije slikara. Za isti skup podataka kao i u prošlom poglavlju, sada je točnost nešto bolja. (Slika 5.1)

```

score = model.evaluate(train_set, verbose=1)
print("Prediction accuracy on train data =", score[1])

187/187 [=====] - 59s 318ms/step - loss: 0.0504 - accuracy: 0.9940
Prediction accuracy on train data = 0.9939779043197632

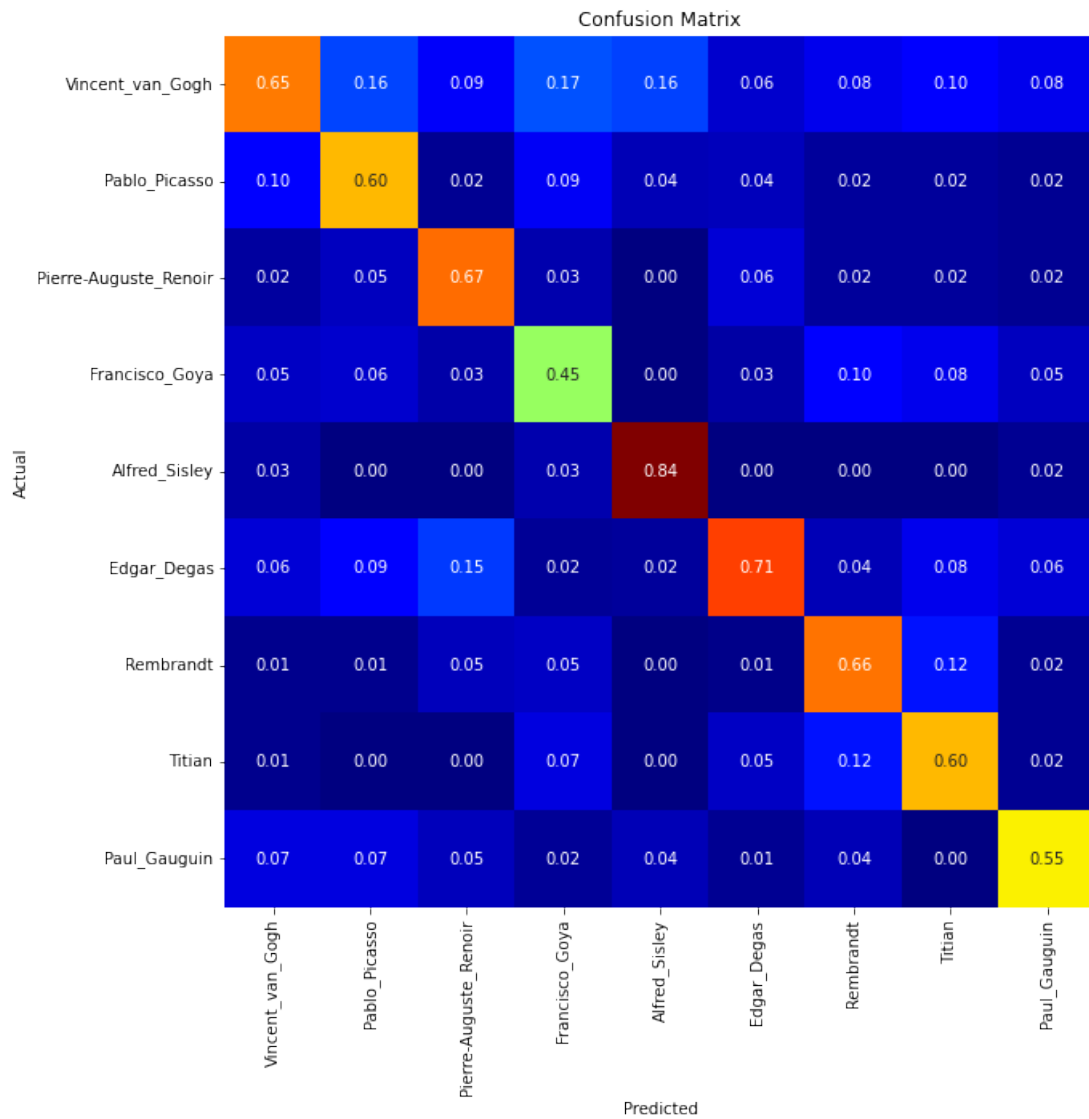
score = model.evaluate(test_set, verbose=1)
print("Prediction accuracy on test data =", score[1])

47/47 [=====] - 16s 344ms/step - loss: 1.3158 - accuracy: 0.6420
Prediction accuracy on test data = 0.641991913318634

```

Slika 5.1: Evaluacija neuronske mreže

Iz matrice konfuzije vidimo da je stvarno ovaj model bolji. Najveće se vrijednosti nalaze na dijagonali. (Slika 5.2)



Slika 5.2: Matrica konfuzije

6. Izrada web aplikacije

6.1. Flask

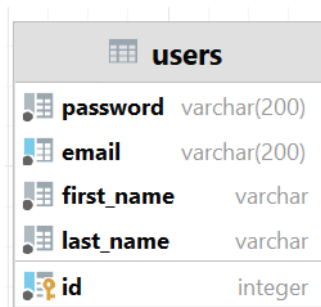
Flask je web okvir za Python. Osmišljen je tako da dizajn aplikacije bude što jednostavniji. Flask ne sadrži sloj apstrakcije baze podataka ili provjeru valjanosti obrazaca, ali to se može implementirati s ekstenzijama. Ekstenzije korištene u ovom radu su: flask login, werkzeug.security.

6.2. Google Colab

Za učenje i razvoj modela korištena je online platforma Google Colab. Ona nam omogućava izvođenje koda u Pythonu putem web preglednika. Posebno je pogodna za potrebe strojnog učenja jer ima pristup dodatnoj sklopovskoj potpori. Kod se izvršava na virtualnom računalu vezanom uz Google račun.

6.3. Baza podataka

Baza podataka sastoji se samo od jednog entiteta *users* (Slika 6.1) u kojem je pohranjeno ime i prezime korisnika, email i lozinka zaštićena *hash* funkcijom.

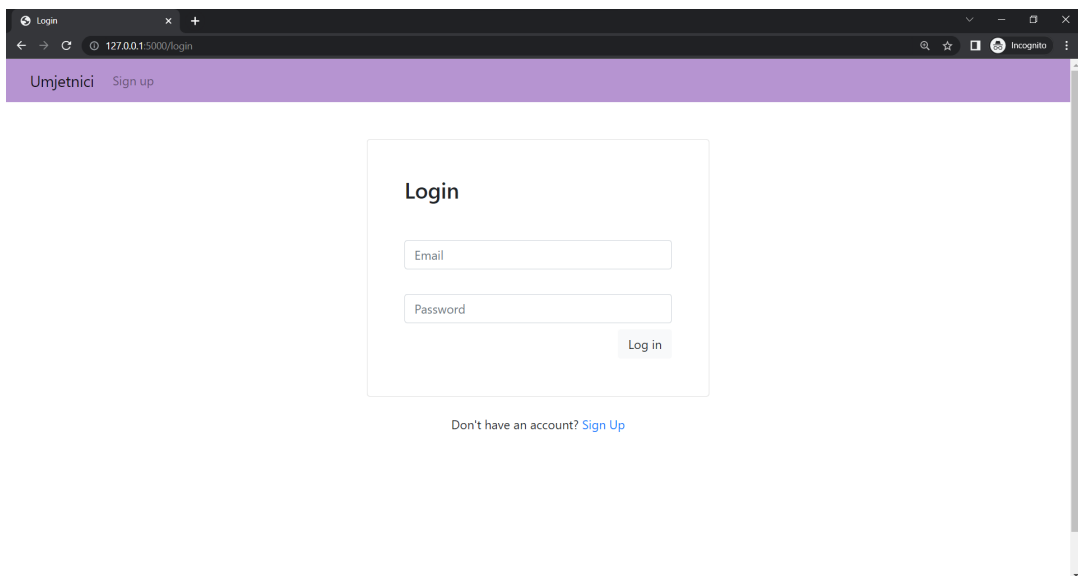


users	
password	varchar(200)
email	varchar(200)
first_name	varchar
last_name	varchar
id	integer

Slika 6.1: Model baze podataka

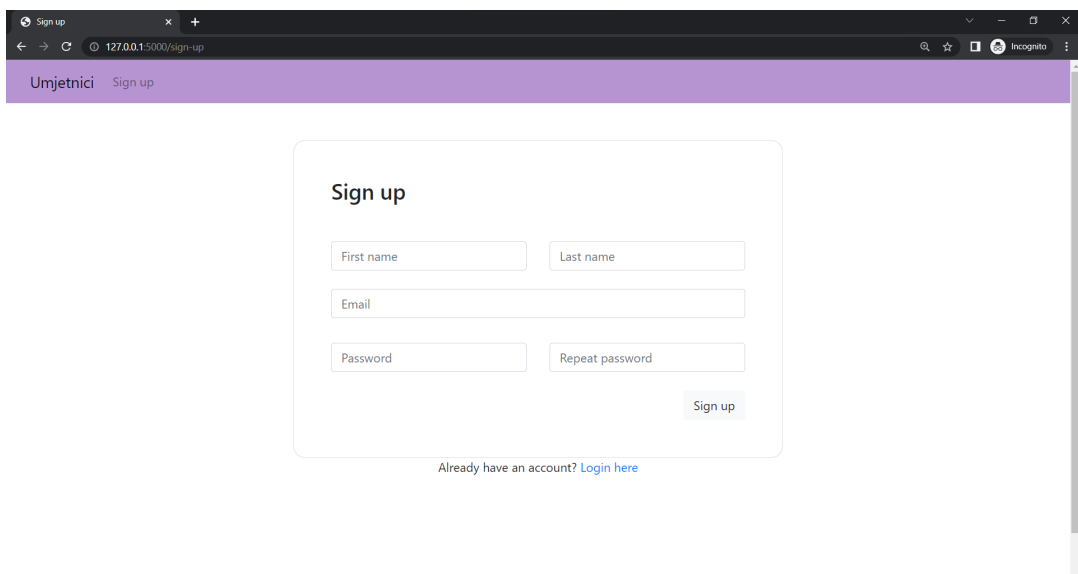
6.4. Opis rada aplikacije

Web aplikacija omogućava korisniku prijavu ako već ima profil (Slika 6.2) ili registraciju putem emaila (Slika 6.3). Nakon prijave, korisnika se preusmjeri na stranicu gdje može predati fotografiju za koju želi saznati slikara (Slika 6.4). Aplikacija može prepoznati umjetnine sljedećih slikara: Vincent van Gogh, Pablo Picasso, Pierre-Auguste Renoir, Francisco Goya, Alfred Sisley, Edgar Degas, Rembrandt, Titian, Paul Gauguin.



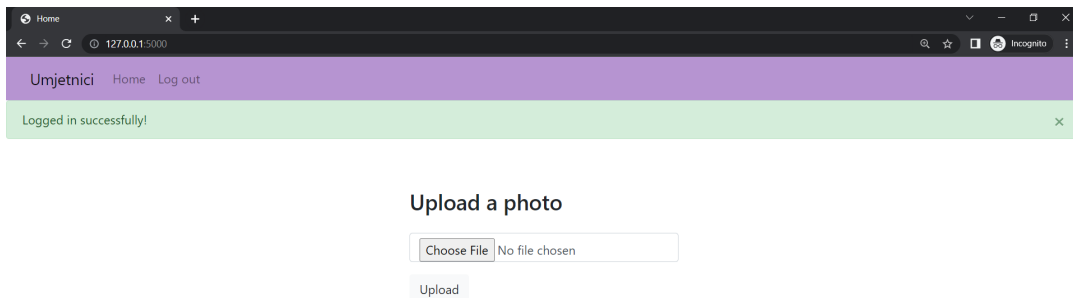
The screenshot shows a web browser window with the URL `127.0.0.1:5000/login`. The page has a purple header with the text "Umjetnici" and "Sign up". The main content is a white box titled "Login" containing two input fields: "Email" and "Password". A "Log in" button is positioned to the right of the password field. Below the box, there is a link: "Don't have an account? [Sign Up](#)".

Slika 6.2: Forma za prijavu u sustav



The screenshot shows a web browser window with the URL `127.0.0.1:5000/sign-up`. The page has a purple header with the text "Umjetnici" and "Sign up". The main content is a white box titled "Sign up" containing four input fields: "First name", "Last name", "Email", and "Password". A "Repeat password" field is also present. A "Sign up" button is positioned to the right of the password fields. Below the box, there is a link: "Already have an account? [Login here](#)".

Slika 6.3: Forma za registraciju korisnika

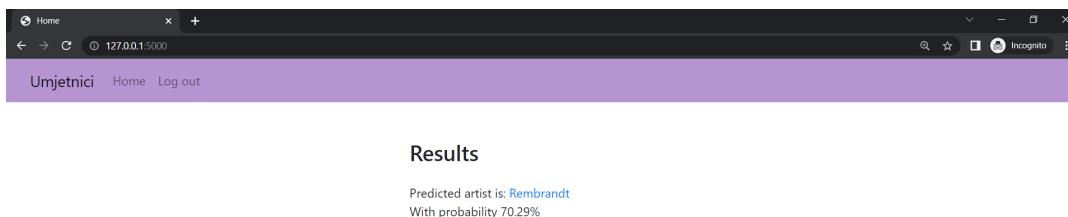


Slika 6.4: Primjer stranice za prijenos fotografije

Prilikom registracije provjerava se postoji li već korisnik s tom email adresom, ako postoji prikazuje mu se ta informacija. U slučaju da ne postoji, provjerava se zadovoljava li unesena lozinka sigurnosne zahtjeve i jesu li unesene lozinke jednake. Lozinka mora imati barem sedam znakova i jedno veliko slovo.

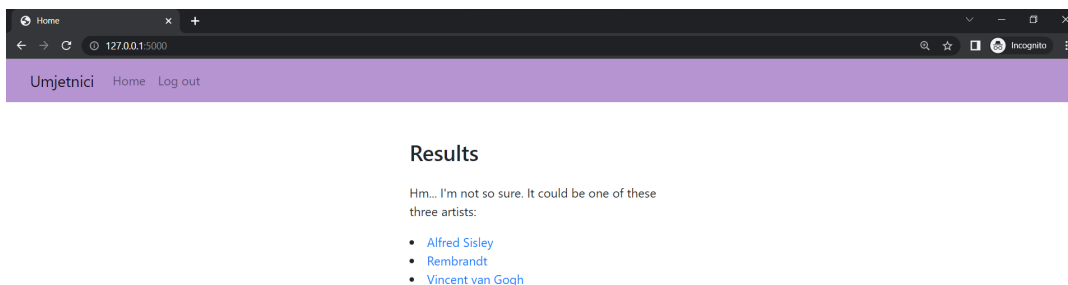
Prilikom prijave korisnika provjerava se postoji li korisnik s tom email adresom i podudara li se unesena lozinka s onom spremljenom.

Prijavljeni korisnik ima mogućnost učitati sliku u aplikaciju. Pritiskom na gumb *Upload*, korisnika se odvede na stranicu gdje vidi o kojem se slikaru radi i s kojom sigurnošću. (Slika 6.5)



Slika 6.5: Primjer stranice kada je velika sigurnost

Dodatna funkcionalnost je da ako je vjerojatnost manja od 50%, korisnika se na to dodatno upozori i sugerira mu se o kojim još umjetnicima bi moglo biti riječ (Slika 6.6). Na stranici s rezultatima, također je i poveznica koja vodi na stanicu Wikipedije dotičnog umjetnika.



Slika 6.6: Primjer stranice kada je mala sigurnost

7. Zaključak

U ovom radu objašnjeni su ključni pojmovi i metode vezane uz rješavanje problema klasifikacije slika. Razvijena su dva modela konvolucijske neuronske mreže, jedan *"from scratch"* s točnošću 0.37415 i drugi kao dodatno učenje već prethodno naučene mreže s točnošću 0.64199. Točnost druge mreže očekivano je bolja jer učenje ne počinje od nule. Dok smo prvu mrežu učili od početka, drugu smo trebali doučiti dodavanjem nekoliko slojeva.

Kako bi ova aplikacija bila korisna korisnicima, točnosti bi trebale biti veće. To bi mogli postići s većim i boljim skupom podataka, te većim resursima za učenje neuronske mreže.

Aplikacija je osmišljena kako bi ljudima zainteresiranim za umjetnost koji nisu obučeni u ovom području olakšala prepoznavanje slikara.

Izradom ovog rada stečena su nova znanja o neuronskim mrežama i njihove izgradnje pomoću Kerasa i TensorFlowa.

LITERATURA

[1] Jason Browlee, *How to Configure Image Data Augmentation in Keras*, 12.04.2019., <https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/>

[2] *An Easy Guide to Neuron Anatomy with Diagrams*, <https://www.healthline.com/health/neurons>

[3] Bojana Dalbelo Bašić, Marko Čupić, Jan Šnajder, Uvod u umjetnu inteligenciju, materijali s predavanja, *11. Umjetne neuronske mreže*, ak. god 2019./2020., Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva

[4] Desaradh S K, *A Gentle Introduction To Math Behind Neural Networks*, 7.10.2020., <https://towardsdatascience.com/introduction-to-math-behind-neural-networks-e8b60dbbdeba>

[4] Jason Browlee, *Transfer Learning in Keras with Computer Vision Models*, 15.05.2019., <https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/>

[5] Alat za vizualizaciju neuronske mreže, <https://alexlenail.me/NN-SVG/index.html>

[6] Sumit Saha, *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way*, 15.12.2018., <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

[7] *CNN | Introduction to Pooling Layer*, 19.07.2021., <https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>

[8] Daniel Johnson, *TensorFlow Basics: Tensor, Shape, Type, Sessions & Operators*, 14.05.2022., <https://www.guru99.com/tensor-tensorflow.html>

[9] *Dropout layer*, https://keras.io/api/layers/regularization_layers/dropout/

[10] Jason Brownlee, *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning*, 3.07.2021., <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>

[11] *ResNet-50 convolutional neural network*, <https://www.mathworks.com/help/deeplearning/ref/resnet50.html>

Web aplikacija za automatsko raspoznavanje slikara prema umjetničkom djelu

Sažetak

U ovom radu analiziran je skup podataka nad kojim će se provoditi učenje neuronske mreže i objašnjeni su ključni pojmovi vezani uz neuronske mreže (neuron, arhitektura neuronske mreže i njeni slojevi). Napravljena su dva modela, jedan "from scratch" i jedan na temelju prethodno naučene mreže, od kojih je bolji implementiran u web aplikaciju.

Ključne riječi: Biološki neuron, Umjetni neuron, Neuronske mreže, Konvolucijske neuronske mreže, Prijenosno učenje, Flask, TensorFlow

Web application for automatic painter recognition from the work of art

Abstract

This paper analyzes the set of data on which neural network learning will be conducted and explains the key concepts related to neural networks (neuron, neural network architecture and its layers). Two models were created, one "from scratch" and one based on previously trained model, the better of which was implemented in a web application.

Keywords: Biological neuron, Artificial neuron, Neural networks, Convolutional neural networks, Transfer learning, Flask, TensorFlow