

Primjena Weka softvera i konvolucijskih neuronskih mreža u procesu klasifikovanja digitalnih fotografija

Branislava Cvijetić
Agencija za statistiku Bosne i Hercegovine
Sarajevo, Bosna i Hercegovina
branislava.cvijetic@bhas.gov.ba

Zaharije Radivojević
Elektrotehnički fakultet, Univerzitet u Beogradu
Beograd, Republika Srbija
zaki@etf.bg.ac.rs

Sažetak — Algoritmi za prepoznavanje oblika na slikama, kao i ljudskih lica proučavaju se dugo, ali su punu primjenu pronašli tokom posljednjih desetak godina. Postoji veliki broj algoritama mašinskog učenja integrisanih u *open source* softver Weka. U ovom radu korišćena su tri algoritma sa ciljem da se utvrdi koliko se uspješno Weka softver može primjenjivati u procesu klasifikovanja digitalnih fotografija. Takođe je opisano korišćenje konvolucijskih neuronskih mreža u procesu klasifikovanja digitalnih fotografija. Urađena su dva eksperimenta i prikazani su dobijeni rezultati.

Ključne riječi - računarska vizija; mašinsko učenje; Weka; konvolucijska neuronska mreža

I. UVOD

Čulo vida je jedno od pet čula koje posjeduje čovjek. Pomoću njega primjećujemo i razaznajemo intenzitet svjetlosti, boje, oblike i međusobne udaljenosti između objekata. Kako za ljude, tako i za mnoge životinje, vizuelna percepcija je jedno od najvažnijih čula, pošto uglavnom ono što se dešava oko nas mi doživljavamo gledajući. Naravno, mi gledamo očima, ali vidimo zahvaljujući mozgu, koji pri tome ima vrlo važnu ulogu. Svjetlosni zraci prolaze kroz zjenicu oka i stvaraju na mrežnjači lik, odnosno odraz nečeg što gledamo. Ono što mozak „vidi“ razlikuje se od slike stvorene na mrežnjači. Na primjer, ako smo u prirodi, mi gledamo pejzaž oko sebe, drveće, ptice i sl. Centar za vid u mozgu prima svaku sliku koju detektuje oko i pamti je. Zatim sve informacije sažima i daje im značenje, te vidimo sliku u cjelini, a ne dio po dio. Vrlo brzo u mozgu se obavlja prepoznavanje ranije viđenih oblika. Pošto smo, na primjer neku vrstu drveta već ranije vidjeli, dovoljan je samo brz pogled na to drvo i da prepoznamo o kojoj vrsti drveta je riječ. Čovjek nikada aktivno ne razmišlja o ovim koracima obrade vizuelnih informacija. Međutim, ono što se čini lako za ljudski mozak još uvijek predstavlja veliki izazov za računarske sisteme, tj. za računarsku viziju.

Računarska vizija (engl. *computer vision*) je interdisciplinarna nauka, čiji je osnovni zadatak kako da računari postignu visok nivo razumijevanja digitalne slike ili videa. Sa tačke gledišta inženjerstva, ona nastoji da automatizuje zadatke koje vizuelni sistem čovjeka može da uradi. Ovo interdisciplinarno polje simulira i automatizuje elemente sistema ljudskog vida pomoću hardvera (senzora, kamera), softvera (algoritama mašinskog učenja) i setova

podataka koje kreiraju hardver i softver. Računarska vizija je dio oblasti vještačke inteligencije.

Počeci razvoja računarske vizije vežu se za jedan od prvih radova na temu računarske vizije. Zapravo radi se o doktorskoj tezi naučnika Lavrenca Gilman Roberta (Lawrence Gilman Roberts) pod nazivom „Mašinska percepcija trodimenzionalnih čvrstih tijela (engl. *Machine Perception of Three-Dimensional Solids*)“ iz 1963. godine. [1]

1966. godine naučnik i profesor Marvin Minski (Marvin Minsky) angažovao je studente prve godine studija na jednom projektu, a njihov zadatak je bio da povežu kamere na računar i dobiju opis šta mašina vidi. Ovaj naučnik je 1969. godine dobio i Tjuringovu nagradu (engl. *Turing Award*), tj. priznanje u oblasti računarskih nauka za njegovu centralnu ulogu u stvaranju, oblikovanju, promovisanju i unapređenju polja vještačke inteligencije. [2]

1973. godine naučnici Martin A. Fišler (Martin A. Fischler) i Robert A. Elšlager (Robert A. Elshlager) objavili su rad „Predstavljanje i usklađivanje slikovitih struktura (engl. *The Representation and Matching of Pictorial Structures*)“ u kojem je ostvaren napredak u interpretaciji odabrane slike. [3]

Tokom osamdesetih godina prošlog vijeka, naučnici spominju vještačke neuronske mreže. Iako je matematički model već tada bio spreman, računari jednostavno nisu bili dovoljno „jaki“ da bi se neuronske mreže mogle učiti na velikim količinama podataka. Nakon toga nastaje period kada su one „zaboravljene“, a tek oko 2000-te godine neuronske mreže se ponovo susreću u radovima naučnika. U periodu do pojave neuronskih mreža, dosta se radilo na razvoju algoritama mašinskog učenja. Čak su neki od njih integrisani i u softverske alate, kao što je i Weka. Za 2000. godinu vežu se i počeci obrade slika sa video zapisa.

2001. godine, naučnici Paul Viola (Paul Viola) i Majkl Džouns (Michael Jones) u svom radu [4] prikazali su algoritam za brzu detekciju lica koji omogućava da se ljudske figure identifikuju kroz njihove osobine lica. Naučnici Navnit Dalal (Navneet Dalal) i Bil Trigs (Bill Triggs) u svom radu [5] iz 2005. godine, opisuju kako prepoznati pješaka u sigurnosnim sistemima.

Tokom narednih desetak godina veliki broj naučnika, kao i mnoge komercijalne kompanije, kao što su: Gugl (engl. *Google*), Amazon, Fejsbuk (engl. *Facebook*) i dr. primjenjuju

algoritme za prepoznavanje lica zasnovane na neuronskim mrežama. Razvoj ovih algoritama otišao je toliko daleko, da su algoritmi u stanju da prepoznaju i emocije na licima osoba koje se nalaze na fotografijama ili videu. [6]

Automatska dijagnoza u medicini je jedna od glavnih i najčešćih primjena prepoznavanja oblika na digitalnoj fotografiji. Koristi se prvenstveno u radiologiji u kombinaciji sa radiologijom pomoću rendgenskih zraka, magnetne rezonance ili ultrazvučnim tehnikama medicinske slike. Uvođenjem automatske dijagnoze, značajno je smanjen faktor ljudske greške pri dijagnozama, a u isto vrijeme olakšan je rad doktorima. Uglavnom se koristi za otkrivanje tumora na mozgu i plućima, ali i na drugim mjestima. Osim za otkrivanje raznih tumora, automatska dijagnoza se primjenjuje i na druge bolesti, kao što su: bolesti arterija, srčane bolesti (analizom kardiografa) itd. Upotreba Weka softvera u oblasti medicine, zabilježena je u radovima [7], [8], [9] i [10].

Računarska vizija se primjenjuje i kod autonomnih vozila, tj. automobila koji sami upravljaju vožnjom bez ili sa malo ljudske pomoći. Takođe se primjenjuje i u industriji za automatizovanje poslovnih procesa, automatsko prepoznavanje teksta, e-trgovini, sigurnosnim sistemima itd.

Klasifikacija digitalnih fotografija je osnovni dio računarske vizije i susreće se u svim, ranije navedenim, oblastima njene primjene. U ovom radu opisan je postupak klasifikacije digitalnih fotografija korišćenjem dva različita pristupa nad istim setom podataka. Prvi pristup baziran je na Weka softveru. Da bi se izvršilo klasifikovanje fotografija u Weka softveru, primjenom integrisanih algoritama mašinskog učenja, digitalne fotografije je bilo potrebno opisati numeričkim informacijama. Za razliku od pristupa opisanog u [10], u ovom radu priprema fotografija nije rađena izvan Weka softvera. Drugi pristup odnosio se na primjenu konvolucijskih neuronskih mreža (engl. *Convolutional Neural Network – CNN*). Za razliku od pristupa opisanog u [10] i [11], u ovom radu digitalne fotografije nisu konvertovane da budu crno bijele, već je klasifikacija rađena nad digitalnim fotografijama u boji. Ovaj rad predstavlja početak istraživanja sa ciljem da se uporede prednosti i nedostaci dva različita pristupa rješavanju istog problema, tj. klasifikacije digitalnih fotografija.

Ostatak rada organizovan je na sljedeći način. U narednom poglavlju opisana su tri algoritma mašinskog učenja koja su integrisana u Weka softver. Takođe je ukratko opisan rad konvolucijskih neuronskih mreža i dat prikaz korišćenih softverskih biblioteka (engl. *software labrary*) pri implementaciji rješenja klasifikacije digitalnih fotografija primjenom programskog jezika Pajton (engl. *Python*). U trećem poglavlju opisan je set podataka koji je korišćen u eksperimentalnom dijelu. Eksperimentalni dio je opisan u četvrtom poglavlju. Naime, u eksperimentalnom dijelu opisana su dva eksperimenta. Cilj oba eksperimenta bio je da se na osnovu ulaznog seta digitalnih fotografija procijeni da li se na fotografijama nalazi ljudsko lice ili ne. U prvom eksperimentu prikazana je klasifikacija digitalnih fotografija korišćenjem Weka softvera primjenom tri algoritma mašinskog učenja. U drugom eksperimentu prikazana je klasifikacija digitalnih fotografija primjenom jednostavne arhitekture konvolucijske

neuronske mreže realizovane u Pajton programskom jeziku. Na kraju je dat zaključak i lista korišćene literature.

II. SOFTVERSKI ALATI

U ovom dijelu rada biće opisan Weka softver, kao i tri algoritma mašinskog učenja koji su korišćeni prilikom prvog eksperimenta. Takođe će ukratko biti opisana jednostavna arhitektura konvolucijske neuronske mreže, programski jezik Pajton, kao i softverske biblioteke koje su korišćene za implementaciju odabrane arhitekture CNN.

A. Weka

Weka je softver koji sadrži kolekciju algoritama mašinskog učenja, a koji se koriste u zadacima rudarenja podataka (engl. *data mining*). Sadrži alate za pripremu podataka, klasifikaciju, regresiju, grupisanje, vizualizaciju itd. Weka je razvijena na Univerzitetu *Waikato* Novi Zeland, pa otuda i Weka predstavlja skraćenicu od „*Waikato Environment for Knowledge Analysis*“. Weka je *open source* softver koji je pod GNU (*General Public License*), što znači da se može koristiti, proučavati, dijeliti ili mijenjati. Može se pokrenuti na bilo kojoj platformi. Prva verzija softvera bila je urađena u programskom jeziku C, a migrirana je na Javu (engl. *Java*) 1997. godine.

Weka sadrži veliki broj algoritama mašinskog učenja, a u tekstu niže, ukratko će biti opisana tri algoritma koja su korišćena u eksperimentalnom dijelu.

1) J48 klasifikator

Algoritam J48 je zasnovan na algoritmu poznatom kao C4.5. C4.5 je proširena verzija ID3 algoritma autora Rosa Kvinlana (Ross Quinlan) iz 1979. godine. Prva verzija C4.5 pojavila se 1993. godine. Druga verzija je izdata 1996. godine i bila je to verzija C4.8 koja, u to vrijeme, još uvijek nije postala komercijalna. Verzija C5.0 je već komercijalna verzija algoritma. Kada je razvijan Weka softver, preuzeta je posljednja *open source* verzija C4.8, a tim koji je razvijao Weka softver je preveo taj kod u Javu, pošto je Weka pisana u Java programskom jeziku. Kako je korišćen C4.8 algoritam, a J je zbog Jave, u Weka softveru ovaj algoritam ima naziv J48.

Algoritam J48 spada u grupu algoritama koji grade klasifikacioni model od trening podataka koristeći stabla odluke (engl. *decission trees*). Ovaj algoritam zasnovan je na strategiji odozgo prema dolje (engl. *top-down*). On bira koji će atribut prvo podijeliti. Zatim kreira grane za svaku vrijednost datog atributa. Dijeli podatke za treniranje u podskupove. Ponavlja proceduru za svaku granu, birajući atribut u svakoj tački na osnovu broja instanci koje dopiru do te tačke.

Prednosti stabla odluke su: jednostavnost i lakoća razumijevanja, mogućnost rada sa numeričkim i kategorijskim vrijednostima, kao i brzina.

2) Naive Bayes klasifikator

Ovaj algoritam baziran je na teoremi engleskog statističara i filozofa Tomas Bajesa (Thomas Bayes) iz 18 vijeka. Ova teorema opisuje vjerovatnoću događaja na osnovu prethodnog poznavanja uslova koji bi mogli biti povezani sa događajem.

Naive Bayes je zasnovan na dvije pretpostavke: da su atributi podjednako bitni, te da su oni statistički nezavisni, tj. da poznata vrijednost jednog atributa ne govori ništa o

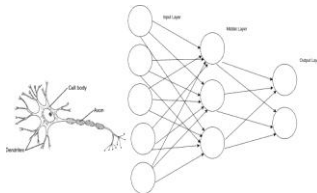
vrijednosti bilo kojeg drugog atributa. Ova pretpostavka o nezavisnosti nikada nije zapravo ispravna, ali metod zasnovan na njoj radi dobro u praksi bez obzira na to što u praksi atributi nisu toliko nezavisni jedni od drugih.

3) MLP

U vještačkoj inteligenciji, neuronske mreže predstavljaju skup modela koji je inspirisan biološkim neuronima. Neuronske mreže su se pokazale kao koristan alat u rješavanju problema klasifikacije kada je teško ili nemoguće pronaći rješenje klasičnim algoritmima koji su zasnovani na eksplicitnim pravilima.

Neuronska mreža sastoji se od jedinica, tj. neurona, raspoređenih u slojeve. Neuroni primaju i prosleđuju signale, tj. pretvaraju ulazni vektor u neki izlaz. Svaka jedinica uzima ulaz, primjenjuje na nju određenu funkciju i zatim prosleđuje izlaz na sledeći sloj. Generalno, neuronske mreže su definisane kao nepovratne (engl. *feed-forward*), a to znači da jedna jedinica šalje svoj izlaz svim jedinicama na sljedećem sloju, ali nema povratne informacije o prethodnom sloju. Učenje neuronskih mreža je zapravo podešavanje težinskih funkcija sve dok se ne dobije zadovoljavajuća aproksimacija između ulaznih i izlaznih veličina.

Na Sl. 1 prikazan je model jednostavne neuronske mreže koja se sastoji od jednog ulaznog sloja (engl. *input layer*), jednog ili više skrivenih slojeva (engl. *hidden layer*), te izlaznog sloja (engl. *output layer*).



Slika 1. Primjer standardne neuronske mreže (slika preuzeta sa [12])

U Weka softveru, kod MLP (engl. *Multilayer Perceptron*), na ulaznom sloju nalazi se po jedan ulaz za svaki atribut, a na izlaznom sloju po jedan izlaz za svaku klasu. Broj skrivenih slojeva se može mijenjati.

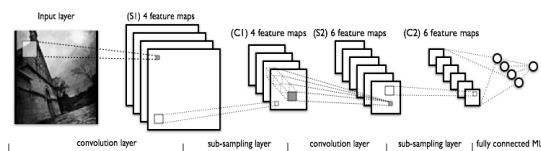
B. Konvolucijske neuronske mreže

1) Arhitektura konvolucijske neuronske mreže

Na Sl. 2 prikazana je opšta arhitektura konvolucijske neuronske mreže, tj. LeNet. LeNet-5 arhitektura prvi put se pojavila 1998. godine. Kreirana je od strane francuskog naučnika Jan Lukana (Yann LeCun), a namjena joj je bila prepoznavanje rukom pisanih slova i brojeva. Ovo je ujedno i prva konvolucijska neuronska mreža. Osim LeNet mreže postoje još i drugi modeli, kao na primjer: AlexNet, VGG16, GoogLeNet, ResNet, SENet itd.

Na Sl. 2 može se vidjeti da ulaz u konvolucijsku mrežu može biti jedna crno-bijela fotografija, ali može biti i fotografija u boji. Zatim slijede naizmjenično konvolucijski slojevi i slojevi sažimanja (engl. *pooling*). Na samom kraju se nalazi nekoliko potpuno povezanih slojeva koji su jednodimenzionalni, uključujući i izlazni sloj. Tipični primjeri konvolucijskih neuronskih mreža imaju oko desetak slojeva

čime jasno opravdavaju svoje mjesto u kategoriji „dubokih“ neuronskih mreža. Konvolucijski slojevi i slojevi sažimanja imaju dvodimenzionalne neurone koji se nazivaju mapama odlika (engl. *feature maps*), koji u svakom sloju postaju sve manjih dimenzija. Zadnji takav sloj je uvijek dimenzija 1x1 (iako je i dalje formalno dvodimenzionalan), te predstavlja vezu na perceptron koji se nalazi u zadnjim slojevima konvolucijske neuronske mreže. Konvolucijski slojevi uvijek imaju vezu jedan-na-više sa prethodnim slojem, te uče težine u jezgrama (engl. *kernels*) sa kojima obavljaju konvoluciju, dok slojevi sažimanja su uvijek u jedan-na-jedan vezi sa prethodnim slojem, te ne uče nikakve vrijednosti, odnosno eventualno neke varijante pamte način propagacije greške.



Slika 2. Primjer LeNet arhitekture konvolucijske neuronske mreže (Slika preuzeta sa [13])

2) Programski alati

Za potrebe eksperimenta kreirane su dvije skripte u programskom jeziku Pajton. Jednostavnost, kao i dostupnost velikog broja softverskih biblioteka i gotovih rješenja, uticao je na odabir Pajton programskog jezika za razvoj skripti koje su korišćene u eksperimentalnom dijelu.

Prva skripta kreira LeNet prototip konvolucijske neuronske mreže, dok druga skripta služi za testiranje dobijenog modela. Obje skripte su razvijene korišćenjem softverske biblioteke *Keras* i frejmvorka (engl. *framework*) *TensorFlow*.

- *Keras* je programska biblioteka otvorenog koda napisana u programskom jeziku Pajton. Namjenjena je za dizajniranje i treniranje dubokih neuronskih mreža. Kao pozadinski kod može koristiti *TensorFlow*, kao i CNTK (*Microsoft Cognitive Toolkit*) ili *Theano*.
- *TensorFlow* je *open source* softverska biblioteka koja je izvorno razvijena od strane istraživača i inženjera iz „*Google Brain*“ tima unutar Guglove organizacije vještačke inteligencije. Dolazi s jakim podrškom za mašinsko učenje i duboko učenje, a fleksibilna numerička računarska jezgra koriste se u mnogim drugim naučnim domenima. Njegova fleksibilna arhitektura omogućava jednostavno korišćenje na različitim platformama i različitim uređajima (serveri, personalni računari, mobilni uređaji i sl.).

III. SET PODATAKA

Za potrebe klasifikacije digitalnih fotografija korišćen je javno dostupan set podataka Caltech-256 [14]. Caltech-256 je kreiran od strane Kalifornijskog Instituta za tehnologiju (California Institute of Technology), a objavljen je 2007. godine. Sadrži ukupno 30.607 digitalnih fotografija kategorisanih u 256 kategorija.

Iz ovog seta podataka korišćena su tri foldera, iz kojih je 435 fotografija klasifikovano kao Face i 191 fotografija klasifikovano kao NoFace. (Tabela I)

TABELA I. PODACI KORIŠĆENI ZA EKSPERIMENT

	Ukupan broj fotografija	Dodijeljena klasa
253.faces-easy-101	435	Face
204.sunflower-101	80	NoFace
024.butterfly	112	NoFace

IV. EKSPERIMENTI

U ovom radu prikazana su dva eksperimenta klasifikacije digitalnih fotografija. Prvi eksperiment odnosi se na klasifikaciju sa Weka softverom, a drugi na klasifikaciju primjenom jednostavne konvolucijske neuronske mreže implementirane primjenom Pajton programskog jezika.

A. Eksperiment primjenom Weka softvera

1) Nadogradnja Weka softvera za potrebe klasifikacije digitalnih fotografija

Za potrebe izvođenja eksperimenta korišćenjem Weka softvera, bilo je potrebno pomoću *Package Manager* instalirati *imageFilters*. Posljednja verzija *imageFilters*, tj. verzija 1.3 izdata je 19.10.2015. godine, a prva verzija ovih filtera u Weku je integrisana 19.12.2014. godine.

Prije uvođenja ovih filtera u softver Weka, digitalne fotografije je prvo bilo potrebno obraditi korišćenjem drugih alata, kao što je npr. *Matlab*, *ImageJ* i sl., da bi se dobile informacije o fotografijama u formatu koji bi se mogao dalje obraditi primjenom algoritama mašinskog učenja sadržanih u Weka softveru. Međutim, nakon uvođenja ovih filtera, u Weka softveru je moguće uraditi pripremu fotografija za dalji postupak klasifikacije.

2) Priprema ulaznih podataka za Weka softver

Priprema ulaznih podataka počinje tako što se prvo kreira jedan folder koji sadrži sve digitalne fotografije iz Tabele I. Na osnovu ovog seta podataka kreira se *Infoteh2019.arff* fajl koji sadrži 627 instanci. ARFF (*Attribute Relation File Format*) je format koji koristi Weka softver.

Da bi se mogli pokrenuti odabrani algoritmi mašinskog učenja, potrebno je da *Infoteh2019.arff* fajl bude smješten u istom folderu gdje se nalaze i fotografije koje se klasifikuju.

Na Sl. 3 prikazan je dio *Infoteh2019.arff* fajla, u kojem su fotografije klasifikovane u dvije kategorije „Face“ i „NoFace“. Ovaj fajl sastoji se od naziva digitalne fotografije i pridružene klase.

```
@relation infoteh2019
@attribute filename string
@attribute class {NoFace, Face}
@data
204_0001.jpg,NoFace
204_0002.jpg,NoFace
```

Slika 3. Prikaz dijela *Infoteh2019.arff* fajla

Drugi korak u pripremi podataka za klasifikovanje, jeste primjena instaliranih filtera nad *Infoteh2019.arff* fajlom, zatim uklanjanje atributa „filename“ i potom snimanje izmjena. U Tabeli II prikazana je pojedinačna primjena filtera nad *Infoteh2019.arff* fajlom. Iz ove tabele može se vidjeti koliko, svaki od filtera, dodaje atributa za svaku pojedinačnu instancu.

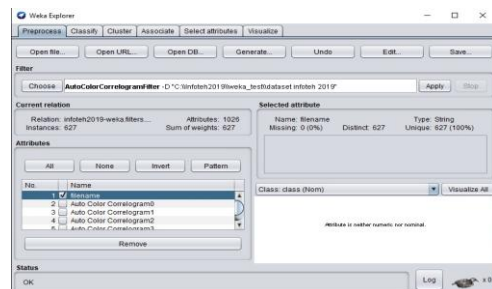
Za potrebe eksperimenta odabran je *AutoColorCorrelogramFilter*, pošto sa najviše informacija opisuje digitalnu fotografiju u boji. Naime, ovaj filter svaku

instancu opisuje sa 1024 karakteristike, a svaka vrijednost atributa opisuje stepen povezanosti instance sa određenom bojom. Korelogram u boji kodira prostornu korelaciju boja sadržanih na digitalnoj fotografiji i predstavlja efektivnu karakteristiku koja je robusna na promjenu pozicije gledanja i zumiranja kamere.

TABELA II. PRIKAZ DOSTUPNIH FILTERA ZA FOTOGRAFIJE U WEKA SOFTVERU I BROJ ATRIBUTA SA KOJIM SVAKI OD NJIH OPISUJE INSTANCU

Naziv filtera	Broj atributa sa filename
<i>AutoColorCorrelogramFilter</i>	1026
<i>BinaryPatternsPyramidFilter</i>	758
<i>ColorLayoutFilter</i>	35
<i>EdgeHistogramFilter</i>	82
<i>FCTHFilter</i>	194
<i>FuzzyOpponentHistogramFilter</i>	578
<i>GaborFilter</i>	62
<i>JpegCoefficientFilter</i>	194
<i>PHOGFilter</i>	632
<i>SimpleColorHistogramFilter</i>	66

Na Sl. 4 prikazan je postupak učitavanja ulaznog fajla *Infoteh2019.arff* i primjena *AutoColorCorrelogramFilter*. Nakon primjene ovog filtera dobijen je novi fajl *Infoteh2019_F1.arff* koji je dalje korišćen u procesu klasifikacije.

Slika 4. Učitavanje *Infoteh2019.arff* fajla u Explorer i primjena filtera

3) Klasifikacija

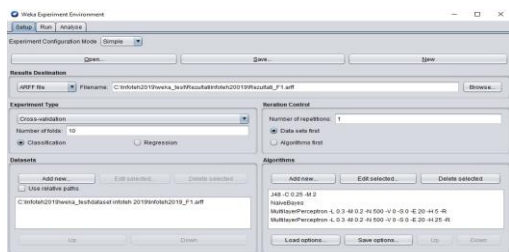
Za potrebe analize podataka korišćena je Weka aplikacija *Experimenter*. Ova aplikacija na jednostavniji način omogućava poređenje rezultata, nego kada se pojedinačno pokreću algoritmi kroz Weka *Explorer* aplikaciju.

U ovom eksperimentu postoji samo jedan set podataka, koji će se koristiti za treniranje i za testiranje modela. Naime, kod svakog algoritma korišćiće se samo opcija desetorostrukog unakrsnog testiranja (engl. *10-fold cross-validation*) Unakrsna validacija radi slično kao razdvajanje fajla na procenete. Ovdje se radi o razdvajanju fajla na određeni broj dijelova (engl. *folds*). U Weka softveru podrazumijevana vrijednost za broj dijelova je 10. Ovo znači da se set podataka dijeli na 10 dijelova. Devet dijelova koristi se za učenje, a jedan dio se ostavi za testiranje. Svaki put uzima se drugi, različit dio za testiranje. Na ovaj način dobija se 10 rezultata procjene, a zatim se računa prosječna tačnost. U stratifikovanoj unakrsnoj validaciji, tj. kada se vrši inicijalna podjela, osigurava se da svaka faza sadrži otprilike tačan dio vrijednosti klase. Nakon što je izvršena desetorostruka unakrsna provjera i izračuna se rezultat evaluacije, Weka pokreće konačni algoritam učenja, tj.

11 put, nad cijelim skupu podataka kako bi se dobio finalni model. [15]

Za potrebe eksperimenta, broj skrivenih slojeva za *Multilayer Perceptron*, postavljen je na 5 (dalje u tekstu MLP-5) i na 25 (dalje u tekstu MLP-25).

Nakon što je kreiran *Weka Experiment Configuration* fajl, pod nazivom *Infoteh2019_F1.exp*, importovan je *Infoteh2019_F1.arff* fajl, tj. fajl nad kojim se radi klasifikacija. Ostavljena je podrazumijevana vrijednost za testiranje, tj. desetorostruko unakrsno testiranje. Zatim su, redom dodavani algoritmi: *J48*, *Naive Bayes*, *MLP-5* i *MLP-25*. (Sl. 5)



Slika 5. Weka *Experimenter* aplikacija

Neki od parametara dobijenih postupkom klasifikovanja digitalnih fotografija, sa tri različita algoritma mašinskog učenja nakon primjene *AutoColorCorrelogramFilter* filtera, prikazani su u Tabeli III.

TABELA III. REZULTATI EKSPERIMENTA SA WEKA SOFTVEROM

	J48	Naive Bayes	MLP-5	MLP-25
<i>Percent_correct</i>	91.87	88.05	88.04	96.50
<i>Percent_incorrect</i>	8.13	11.95	11.96	3.50
<i>Kappa_statistic</i>	0.80	0.73	0.63	0.92
<i>IR_precision</i>	0.90	0.78	0.98	0.96
<i>IR_recall</i>	0.84	0.87	0.63	0.93
<i>F_measure</i>	0.86	0.82	0.93	0.94
<i>Elapsed_Time_training</i>	0.15	0.04	20.82	96.13
<i>Elapsed_Time_testing</i>	0.00	0.02	0.00	0.01

Iz Tabele III može se vidjeti da najveći procenat korektno klasifikovanih instanci ima algoritam MLP-25 (96.50), a zatim J48 (91.87). Algoritam MLP-25 ima najveću vrijednost *Kappa* statistike 0.92, a zatim J48 0.80. Kako algoritam MLP-25 ima veću vrijednost za *Kappa* statistiku, to znači da postoji bolja usklađenost između atributa i vrijednosti instanci. Najbolju *Precision* dao je algoritam MLP-5. Sa druge strane, najveću vrijednost za *Recall* dali su algoritmi *Naive Bayes* i MLP-25, tj. oni imaju najbolji odnos ispravnih klasifikacija i predviđenih klasifikacija. Kod algoritma MLP-25 postoji najveća sličnost između vrijednosti *Precision* i *Recall*, pa se može reći da taj algoritam po tačnosti najbolje funkcioniše. Iako se radi o malom setu podataka, postoji značajna razlika u vremenu potrebnom za trening. Algoritam MLP-25 je najviše vremenski zahtjevan. Ukoliko bi zanemarili vrijeme potrebno za kreiranje modela, onda bi mogli reći da je najbolje rezultate dao algoritam MLP-25.

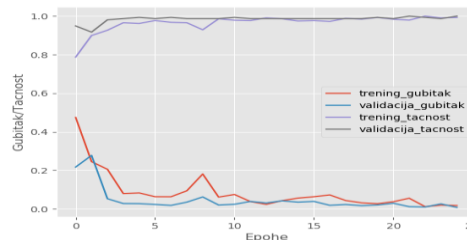
B. Eksperiment primjenom konvolucijskih neuronskih mreža

1) Kreiranje modela konvolucijske neuronske mreže

Pokretanje skripte za treniranje mreže obavljeno je u Anakonda Promptu. Skripta se poziva u prethodno

pripremljenom i aktivnom infoteh okruženju. Zadatak ove skripte je da kreira *Face_NoFace.model*, tj. model koji će biti u mogućnosti da na osnovu nove fotografije, tj. one koja nije korišćena u procesu kreiranja modela, prepozna da li se na fotografiji nalazi ljudsko lice ili ne.

Na Sl. 6 prikazani su rezultati prilikom procesa kreiranja modela konvolucijske neuronske mreže.



Slika 6. Rezultat skripte *train_network.py* prilikom kreiranja modela mreže

U skripti *train_network.py* postavljeno je, da se obučavanje mreže radi nad 75% podataka, a trening nad ostalih 25% podataka. Broj epoha je postavljen na 25, a zatim na 11. Naime, broj epoha označava koliko puta cijeli set podataka prođe kroz mrežu. Na Sl. 6 može se vidjeti da je u jednom trenutku (oko 11 epoha) došlo do pojave da su trening gubici pali ispod validacijskih gubitaka i ta pojava se naziva *overfitting*.

2) Rezultati CNN i analiza rezultata

Za potrebe testiranja kreiranog „obučenog“ modela neuronske mreže, kreirana je skripta pod nazivom *test_network.py*. Test je urađen nad dvanaest digitalnih fotografija koje nisu korišćene prilikom obučavanja, tj. kreiranja modela konvolucijske neuronske mreže. Dobijeni rezultati prikazani su kao matrica zabune (engl. *confusion matrix*) (Tabela IV).

TABELA IV. MATRICA ZABUNE

		predviđena klasa 25 epoha		predviđena klasa 11 epoha	
		Face	NoFace	Face	NoFace
stvarna klasa	Face	TP=3	FN=0	TP=5	FN=0
	NoFace	FP=3	TN=6	FP=1	TN=6

Kada su poznata četiri parametra (TP, FN, TN, FP) iz Tabele IV, moguće je izračunati osnovne parametre validacije (prikazane u Tabeli V), po sljedećim formulama:

- Sveukupna stopa uspjeha je broj ispravnih klasifikacija podijeljen sa ukupnim brojem klasifikacija.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (1)$$

- Stopa greške u procentima je broj neispravnih klasifikacija podijeljen sa ukupnim brojem klasifikacija.

$$Error\ rate = \frac{FP+FN}{TP+TN+FP+FN} \times 100 \quad (2)$$

- Precision* predstavlja odnos ispravnih klasifikacija i ukupnih pozitivno predviđenih klasifikacija.

$$Precision = \frac{TP}{TP+FP} \quad (3)$$

- *Recall* je odnos ispravno predviđenih pozitivnih klasifikacija na sva zapažanja u stvarnoj klasi.

$$Recall = \frac{TP}{TP+FN} \quad (4)$$

- *Fmeasure* je mjera koja kombinuje *Precision* i *Recall*.

$$FMeasure = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

TABELA V. REZULTATI EKSPERIMENTA SA CNN

	11 epoha	25 epoha
<i>Accuracy</i>	91.67	75
<i>Error rate</i>	8.33	25
<i>Precision</i>	0.83	0.5
<i>Recall</i>	1	1
<i>FMeasure</i>	0.91	0.67

Iz Tabele V može se vidjeti da su rezultati testiranja za gotovo sve parametre validacije, bolji u slučaju kreiranog modela sa 11 epoha u odnosu na model sa 25 epoha. Naime, posle 11 epohe došlo je do pojave *overfitting* kreiranog modela, a to je uticalo na lošije rezultate u slučaju primjene modela sa 25 epoha.

ZAKLJUČAK

U radu je dat prikaz dva različita pristupa klasifikaciji digitalnih fotografija. Naime, na osnovu 627 ručno klasifikovanih digitalnih fotografija, primijenjeno je nadgledano mašinsko učenje sa Weka softverom. Takođe je klasifikovanje podataka urađeno i korišćenjem jednostavne arhitekture konvolucijske neuronske mreže, tj. LeNet arhitekture, implementirane u Pajton programskom jeziku.

Klasifikacija digitalnih fotografija primjenom Weka softvera zahtijevala je prethodnu obradu digitalnih fotografija, tj. pripremu podataka u formatu koji se može klasifikovati sa algoritmima u Weka softveru. U cilju realizovanja ovog zadatka primijenjen je *AutoColorCorrelogramFilter*. Navedeni su i drugi Weka filteri za digitalne fotografije, te je prikazan broj atributa koje, svaki od filtera, dodaje svakoj instanci. Analizom rezultata utvrđeno je da najbolje rezultate daje algoritam MLP sa 25 skrivenih slojeva. Takođe se utvrdilo da ovaj algoritam zahtjeva i najviše vremena za izvršavanje obučavanja. S druge strane, kreiranje modela konvolucijske neuronske mreže zahtjeva poznavanje programiranja, kao i način funkcionisanja arhitekture mreže. Rezultati testiranja modela pokazali su bolje rezultate u slučaju korišćenja 11 epoha zbog pojave problema *overfitting* modela konvolucijske neuronske mreže. Takođe, rezultati testiranja u slučaju malog seta podataka, pokazali su da je moguće i to sa približnom preciznošću, koristiti dva različita pristupa, tj. Weka softver i konvolucijske neuronske mreže za klasifikovanje digitalnih fotografija.

U budućem radu potrebno je istražiti uticaj primjene drugih filtera, sadržanih u Weka softveru, na rezultate testiranja nad odabranim setom podataka. S druge strane, u slučaju konvolucijskih neuronskih mreža, potrebno je poboljšati postojeći model, da bi se izbjegla pojava problema *overfitting*.

ZAHVALNICA

Rad na ovom projektu je delimično bio finansiran od strane Ministarstva prosvete i nauke Republike Srbije (III44009). Autori zahvaljuju na finansijskoj podršci.

LITERATURA

- [1] L. Roberts, „Machine Perception of Three-Dimensional Solids,“ Massachusetts Institute of Technology, 1963.
- [2] Association for Computing Machinery, „Chronological Listing of A.M. Turing Award Winners,“ <https://amturing.acm.org/byyear.cfm> [pristupljeno 22.01.2019.]
- [3] M. Fischler and R.Elshlager, „The Representation and Matching of Pictorial Structures,“ Transactions on Computers, IEEE, 1973.
- [4] P. Viola and M. Jones, „Rapid Object Detection using a Boosted Cascade of Simple Features,“ Conference on Computer Vision and Pattern Recognition, IEEE, 2001.
- [5] N. Dalal and B. Triggs „Histograms of Oriented Gradients (HOG),“ Conference on Computer Vision and Pattern Recognition, IEEE, 2005.
- [6] A. Fathallah, L. Abdi and A. Douik „Facial Expression Recognition via Deep Learning,“ International Conference on Computer Systems and Applications, IEEE, 2017.
- [7] P. Barnaghi, V. Sahzabi, and A. Bakar, „A Comparative Study for Various Methods of Classification,“ International Proceedings of Computer Science and Information Technology, vol. 27, 2012.
- [8] M. Othman and T. Yau, „Comparison of different classification techniques using WEKA for breast cancer,“ 3rd Kuala Lumpur International Conference on Biomedical Engineering, 2006, pp. 520-523.
- [9] W. Ismail, R. Hassan, A. Payne, and S. Swift, „The detection and classification of blast cell in Leukaemia Acute Promyelocytic Leukaemia (AML M3) blood using simulated annealing and neural networks,“ Conference AIME Artificial Intelligence in Medicine, 2011.
- [10] J. Alkrimi, H. Jalab, L. George, A. Ahmad, A. Suliman, and K. Al-Jashamy, „Comparative Study Using Weka for Red Blood Cells Classification,“ World Academy of Science, Engineering and Technology International Journal of Medical, Health, Pharmaceutical and Biomedical Engineering, vol. 9, 2015.
- [11] D. Jaswal and S. Soman, „Image Classification Using Convolutional Neural Networks,“ International Journal of Scientific and Engineering Research, vol. 5, 2014.
- [12] <https://pranjut.files.wordpress.com/2017/06/ann-vs-bnn.jpg?w=640> [pristupljeno 22.01.2019.]
- [13] <http://deeplearning.net/tutorial/lenet.html> [pristupljeno 22.01.2019.]
- [14] http://www.vision.caltech.edu/Image_Datasets/Caltech256/, [pristupljeno 22.01.2019.]
- [15] I. Witten and E. Frank, „Data Mining Practical Machine Learning Tools and Techniques,“ Elsevier, 2005, p. 150.

ABSTRACT

Algorithms for identifying shapes in images, as well as human faces are studied for a long time, but have been fully applied over the past ten years. There are a number of machine learning algorithms integrated in the open source software Weka. In this paper, three algorithms were used in order to determine how successful Weka software can be applied in the process of classifying digital photographs. The use of convolutional neural networks in the process of classifying digital photographs is also described. Two experiments were performed and the obtained results were presented.

Application of Weak Software and Convolutional Neural Networks in the Process of Digital Photography Classification

Branislava Cvijetic and Zaharije Radivojevic