Seminarski rad predmeta istraživanje podataka B

Klasifikacija rukom pisanih karaktera na skupu podataka DoubledMNIST

Student: Milan Čugurović

Profesor: prof. dr. Nenad Mitić

Asistent: Mirjana Maljković

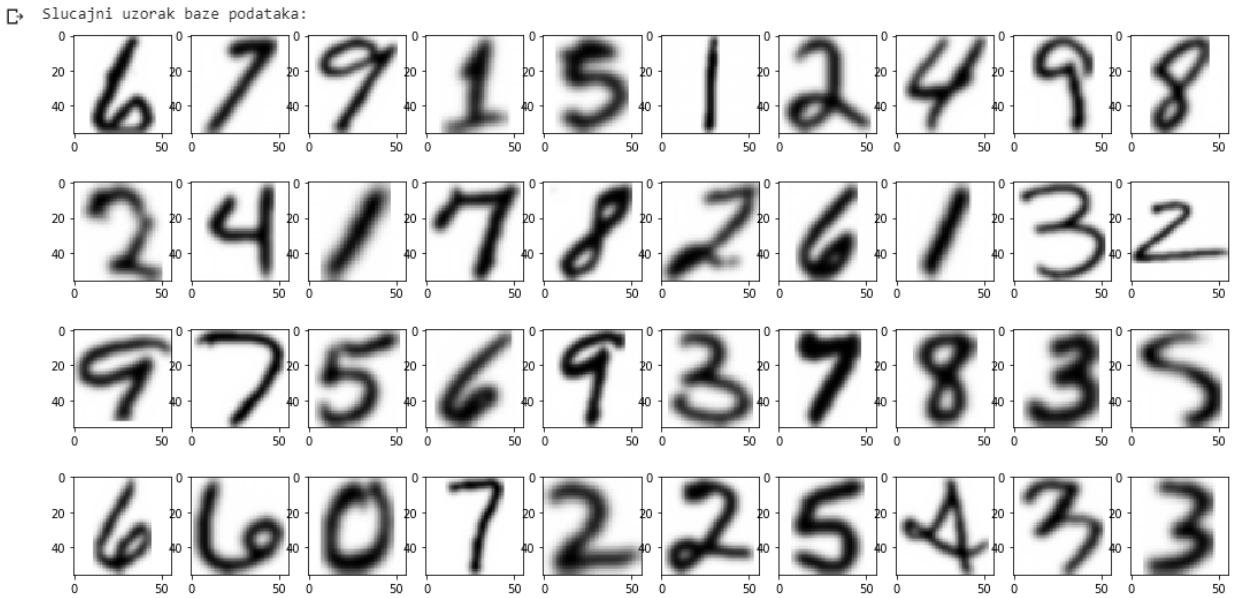
Uvod

Klasifikacija rukom pisanih karaktera predstavlja jedan nadasve praktičan zadatak. Ovaj problem je stoga razmatran kako u akademskoj, tako i u industrijskoj zajednici. Poslednja isti komercijalizuje uključujući klasifikatore rukom pisanog teksta u softver modernih uređaja poput tableta i pametnih telefona. Time svakodnevni život prosečnog korisnika postaje jednostavniji.

Najpoznatiji skup rukom pisanih karaktera svakako jeste čuveni MNIST dataset[1]. Objavljen davne 1997. godine isti je postao kultni skup podataka ne samo konkretne oblasti, već nauke o podacima uopšte. Ovaj skup podataka sastoji se od $70\,000$ slika rezolucije 28×28. Trening skup sadrži $60\,000$ a test $10\,000$ slika. Mnogi javno objavljeni modeli dostižu skoro stoprocentnu preciznost na istom.

Ideja ovog rada jeste evaluacija raznih klasifikacionih modela na skupu podataka DoubledMNIST[2] koji treba da bude naslednik pomenutog skupa podataka. DoubledMNIST se sastoji od $140\,000$ slika rezolucije 56×56. Otuda dolazi i sam njegov naziv. Originalni MNIST dupliran je kako u smislu broja konkretnih instanci, tako i u smislu rezolucije pojedinačnih instanci. Broj atributa konkretne instance jednak je broju piksela iste, odnosno jednak je 3136.

Konkretan uzorak elemenata (slika) baze podataka DoubledMNIST dat je na slici ispod.

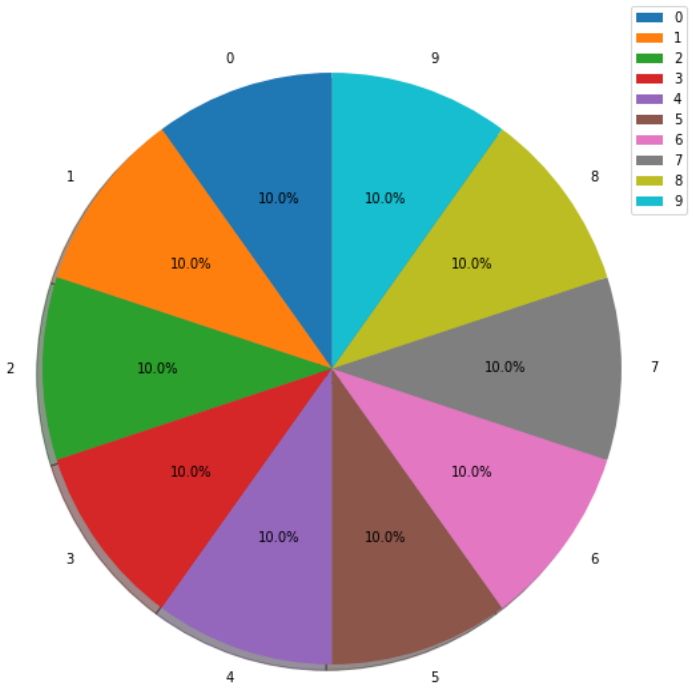


S obzirom na pomenuti veliki broj atributa konkretnih instanci, kao i na veliki broj samih instanci, iz skupa podataka izdvojen je jedan manji podskup, reda veličine 1000 instanci (slučajan uzorak) koji služi tome da se na njemu izaberu parametri konkretnih modela (modela najbližih suseda, metoda potpornih

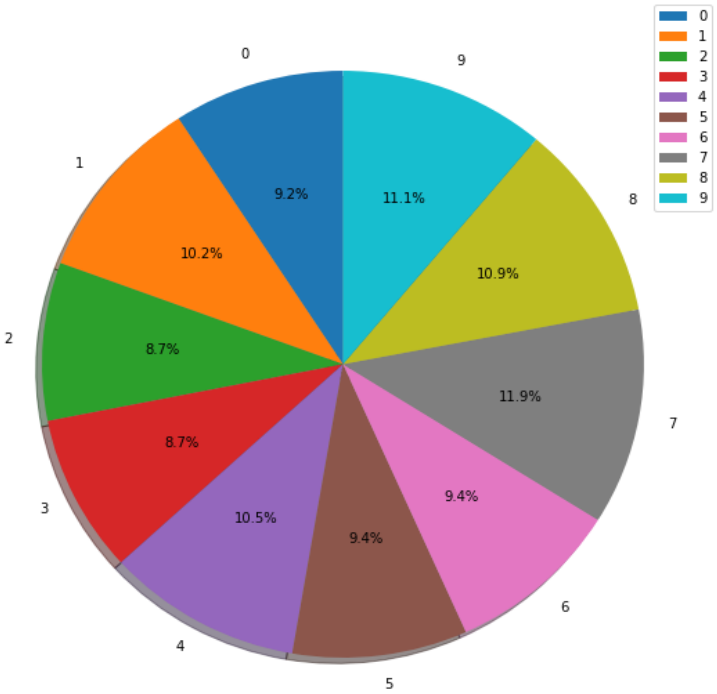
[1] http://yann.lecun.com/exdb/mnist/
[2] Autor Milan Čugurović, 2019.

vektora, kao i odgovarajući parametri stabla odlučivanja). Konkretna raspodela labela polaznog skupa podataka kao i slučajnog uzorka, data je na slici ispod.

Raspodela labela na celom skupu podataka



Raspodela labela na skupu za izbor parametara



Klasifikacioni modeli koji će biti razmatrani u okviru ovog rada jesu metoda k najbližih suseda, metod potpornih vektora za klasifikaciju, odgovarajuća stabla odlučivanja (slučajne šume) kao i duboka neuronska mreža. Za implementaciju će biti korišćen programski jezik Python, dok će za vizuelizaciju biti korišćen kako Python tako i IBM SPSS Modeler.

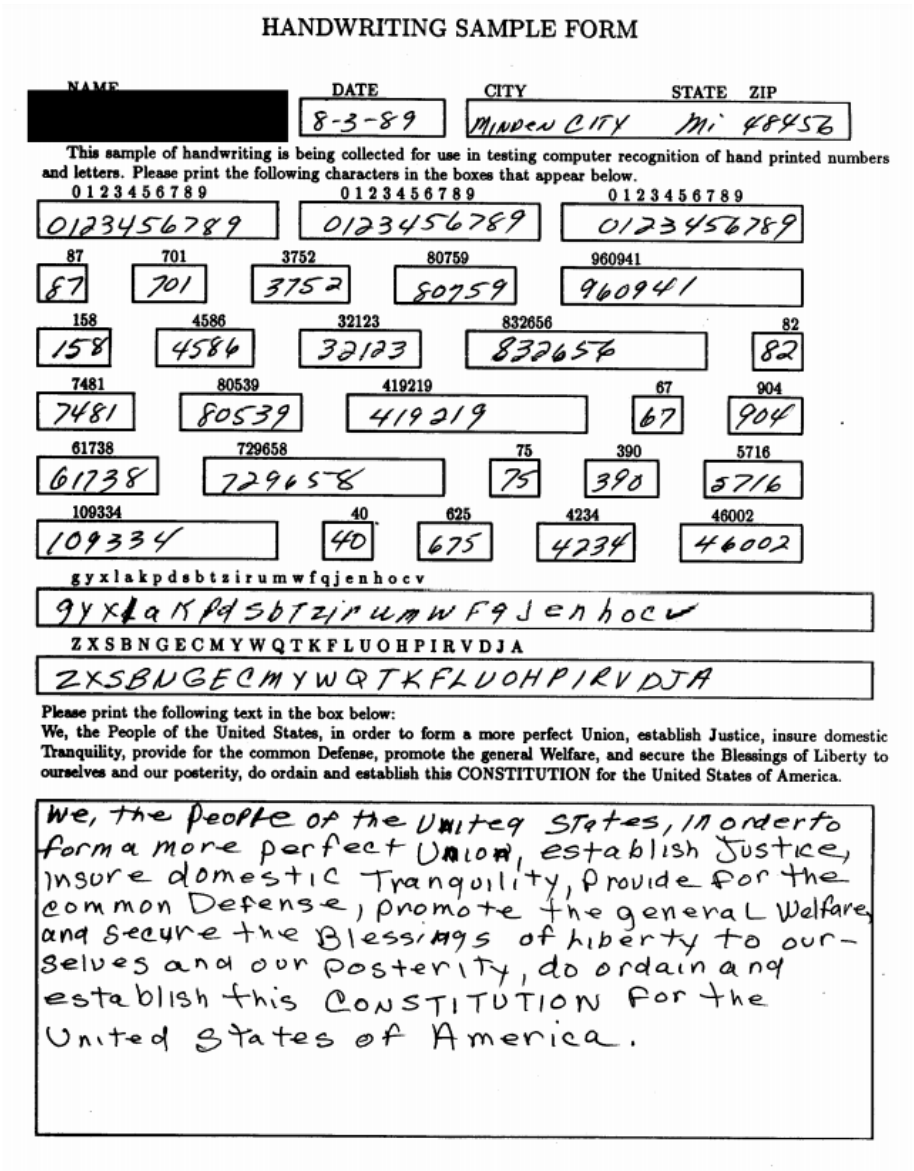
Preprocesiranje podataka

Skup podataka DoubledMNIST nastao je na osnovu skupa podataka NIST Special Database 19, objavljenog od strane Američkog nacionalnog instituta za standarde i tehnologiju. Pomenuta baza sadrži više od $800\,000$ slika rukom pisanih karaktera, od kojih je za bazu DoubledMNIST odabrano i na odgovarajući način transformisano njih $140\,000$.

Preprocesiranje podataka odrađeno je na osnovu časova predavanja profesora Nenada Mitića kao i po preporukapa odgovarajuće prateće literature. Isti proces uključuje sledeće stavke:

- Izdvajanje karakteristika
- Prenosivost tipova podataka
- Čišćenje podataka
- Izbor i transformacija podataka
- Redukcija podataka

Izdvajanje karakteristika odrađeno je na osnovu odgovarajućih formi, koje su popunjavali konkretni autori u procesu prikupljanja podataka. Isti zadatak odrađen je prilikom kreiranja skupa podataka NIST Special Database 19, i to od strane pomenutog Instituta. Primer forme dat je na narednoj slici.



HANDWRITING SAMPLE FORM

NAME [redacted] DATE 8-3-89 CITY MINDEN CITY STATE MI ZIP 48456

This sample of handwriting is being collected for use in testing computer recognition of hand printed numbers and letters. Please print the following characters in the boxes that appear below.

0 1 2 3 4 5 6 7 8 9

0123456789 0123456789 0123456789

87 701 3752 80759 960941

158 4586 32123 832656 82

7481 80539 419219 67 904

61738 729658 75 390 5716

109334 40 625 4234 46002

g y x l a k p d s b t z i r u m w f q j e n h o c v

ZXSBNGECMYWQTKFLUOHPIRVDJA

Please print the following text in the box below:
We, the People of the United States, in order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common Defense, promote the general Welfare, and secure the Blessings of Liberty to ourselves and our posterity, do ordain and establish this CONSTITUTION for the United States of America.

We, the People of the United States, in order to form a more perfect Union, establish Justice, insure domestic Tranquility, provide for the common Defense, promote the general Welfare, and secure the Blessings of liberty to ourselves and our posterity, do ordain and establish this CONSTITUTION for the United States of America.

Prenosivost tipova podataka vidi se kroz odgovarajuću binarizaciju odgovarajućih kategoričkih atributa koji predstavljaju odgovarajuće cifre. Deset mogućih cifara '0', '1', …, '9' enkodiraju se pomoću niza binarnih promenljivih dužine 10, pri čemu i-ta cifra ima jedinicu na odgovarajućoj i-toj poziciji u vektoru.

Čišćenje podataka uključuje rad sa nedostajućim vrednostima, skaliranje i normalizaciju. Rad sa nedostajućim vrednostima odnosi se na polja koja nedostaju u okviru rukom pisanih formi pojedinačnih korisnika. Izabrano je najjednostavnije moguće rešenje, ista se prosto ignorišu. Ovo za posledicu ima činjenicu da različiti autori imaju različit skup ekstrahovanih karaktera. Skaliranje i normalizacija odnose se na fazu treninga konkretnih modela, kada se pikseli pomenute slike predstavljaju kao vrednosti intervala $[0, 1]$. Ovo je korišćeno u svih pet modela klasifikacije.

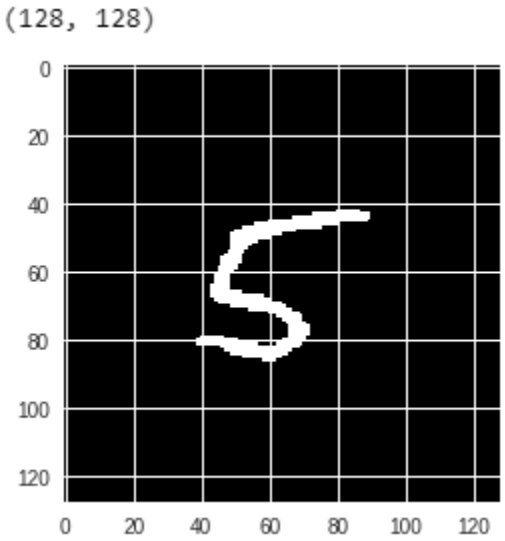
Redukcija i transformacija podataka odnosi se na proces kreiranja slika skupa DoubledMNIST. Originalne slike skupa podataka NIST predstavljene su u rezoluciji 128×128 piksela. Na iste se

primenjuje dodavanje Gausovog zamućivanja sa parametrom $\sigma = 1$, isecanje konkretnog karaktera, centriranje u odgovarajući kvadratni frejm kao i bikbnubna interpolacija u željenu rezoluciju 56×56. Pomenute transformacije nalaze se u okviru Jupyther sveske DoubledMNIST.ipynb. Funkcija *read_gray* implementira čitanje originalnih slika NIST baze sa diska. Funkcija *crop_image* implementira isecanje odgovarajućeg karaktera sa marginom debljine dva piksela iz originalnih slika. Funkcija *add_noise* implementira dodavanje konkretnog Gausovog zamućenja. Funkcija *square_image* implementira transformaciju isečenog karaktera u sliku kvadratnih dimenzija, zadržavanjem originalnih proporcija karaktera. Funkcija *to8bit* transformiše binarnu sliku u osmobitnu monohromatsku reprezentaciju, pogodnu za čuvanje na disku.

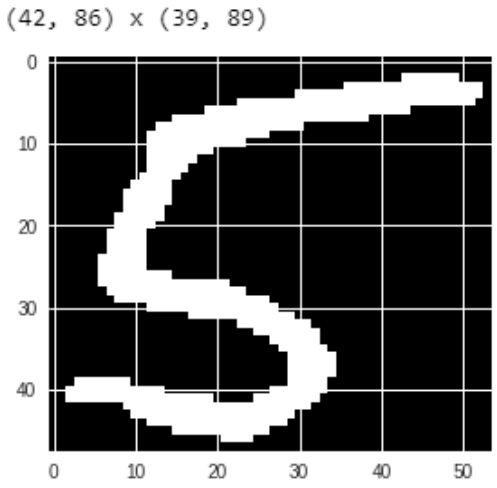
Preprocesiranje slika ukljucuje dodatno i njihovo deljenje sa 255.0 koje svaki piksel iz vrednosti $[0, 255]$ slika u segment $[0, 1]$.

Konkretan primer preprocesiranja dat je u nastavku.

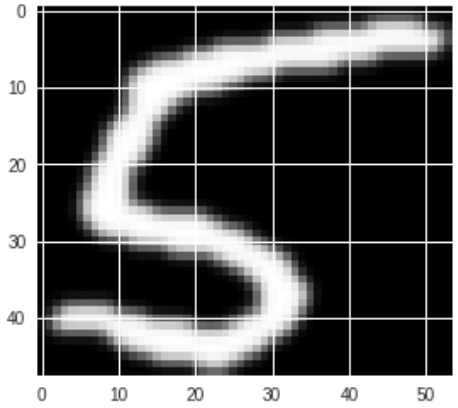
Originalna slika NIST dataseta jeste monohromatska slika u rezoluciji 128×128 piksela. Ista je prikazana na narednoj slici.



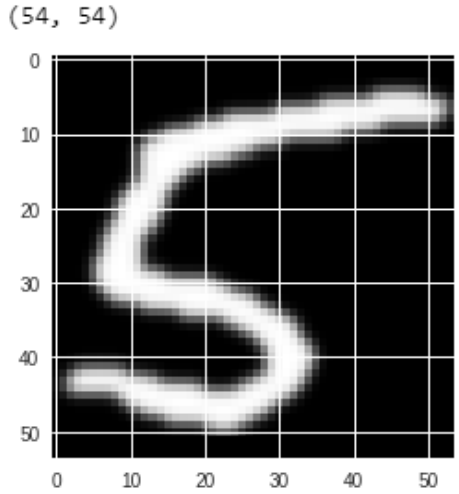
Sledeći korak podrazumeva odgovarajuće isecanje konkretnog karaktera iz cele slike. Dobija se slika kao na ilustraciji.



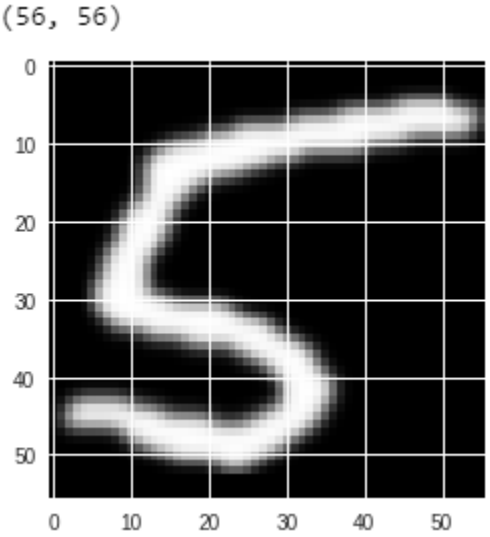
Nakon isecanja na konkretniu sliku dodaje se Gausovo zamućenje sa parametrom sigma jednako 1. Ilustracija je data na narednoj slici.



Nakon dodavanja šuma vrši se centriranje slike u kvadratne dimenzije.

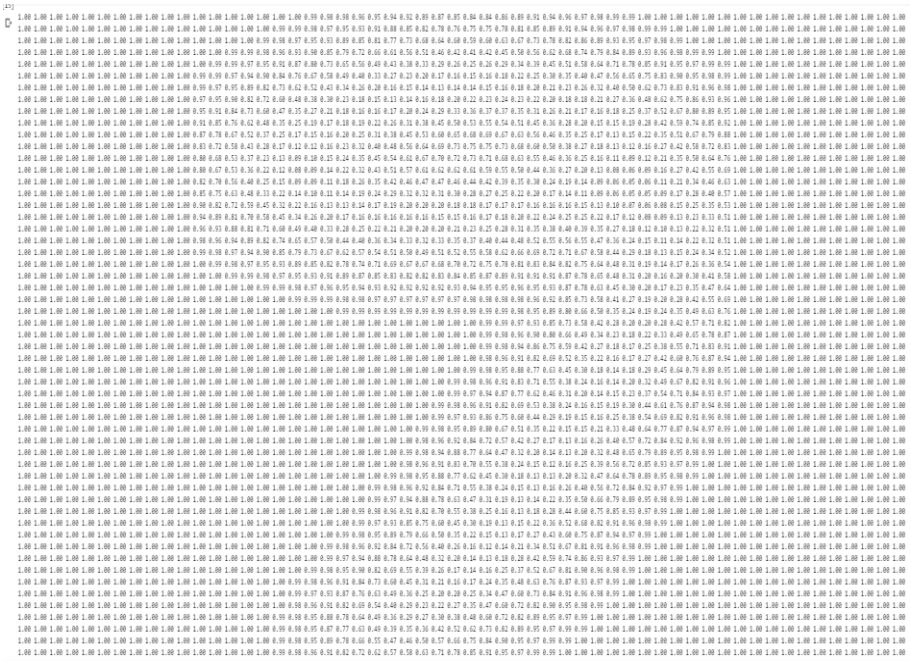


Finalno bikubnom interpolacijom slika se transformiše u rezoluciju 56×56 piksela.



Konkretan kod kojim se vrše odgovarajuće transformacije može se naći u pratećoj svesci.

Ilustracije radi, u nastavku je data jedna konkretna slika pomenutog dataseta predstavljena datim numeričkim vrednostima[3]:

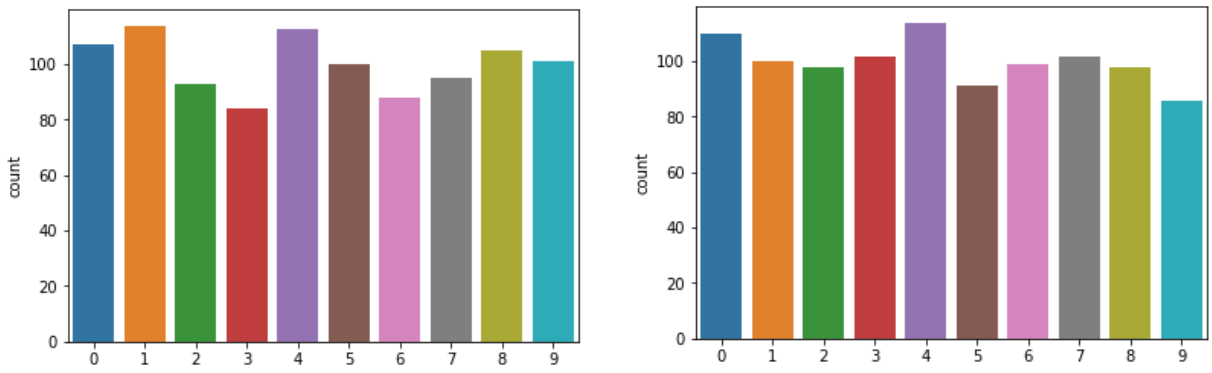


[3] Da se stekne osećaj o "težini" problema odnosno konkretnom broju atributa pojedinačne instance

Metod k najbližih suseda

Za implementaciju metoda k najbližih suseda koristi se klasa $KNeighborsClassifier$[4] paketa $sklearn$.

S obzirom na gabarite samog skupa podataka, izbor odgovarajućih parametara metoda k najbližih suseda vršen je na skupu $x\_train\_choose/y\_train\_choose$ odnosno $x\_test\_choose/y\_test\_choose$ koji sadrže po hiljadu instanci osnovnog skupa podataka. Konkretna raspodela instanci u pomenutim skupovima tim redom, data je na sledećoj slici:



Konkretne statistike za svaki od 3136 pojedinačnih piksela konkretne slike, računate na ovom skupu podataka (koje aproksimiraju osobine celog skupa podataka) date su[5] sa:

	0	1	2	3	4	5	6	7	8	9	10
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	0.999376	0.999169	0.998580	0.998075	0.997259	0.996467	0.995267	0.994145	0.991922	0.990333	0.987639
std	0.002654	0.004151	0.006436	0.009676	0.012753	0.015853	0.019160	0.022327	0.026049	0.030777	0.036937
min	0.933333	0.894118	0.835294	0.768627	0.717647	0.698039	0.709804	0.749020	0.760784	0.733333	0.713725
25%	1.000000	1.000000	1.000000	0.999020	0.996078	0.996078	0.996078	0.996078	0.996078	0.996078	0.996078
50%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.996078	1.000000	0.996078
75%	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Računate su očekivanje i standardna devijacija, minimum odnosno maksimum kao i odgovarajući percentili. Na osnovu konkretnih vrednosti za prvi piksel (odgovara gornjem levom uglu konkretnih slika) jasno se vidi da je on u većini slika iste, crne boje. Isto je i očekivano, prosto konkretan karakter centriran je na odgovarajućoj slici.

Kao odgovarajuća metrika u ovom radu koristi se Minkovski rastojanje sa odgovarajućom vrednosti parametra p: jedan (Menhetn rastojanje) i dva (Euklidsko rastojanje). Pored parametra p, ispituju se još broj suseda, koji se bira iz skupa $[1,2,3,4,5,6,7,8,9,10,30]$, kao i činjenica da li se pri glasanju metodom

[4] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
[5] Za prvih deset piksela; informacije o statistikama za sve piksele mogu se naći u pratećoj svesci

najbližih suseda uzima u obzir konkretno rastojanje tih *k* instanci ili ne. Odgovarajući proces izbora, daje rezultate prikazane na sledećoj slici:

```
knn(k=1, p=1, weights=uniform) : 0.871
knn(k=1, p=1, weights=distance) : 0.871
knn(k=1, p=2, weights=uniform) : 0.883
knn(k=1, p=2, weights=distance) : 0.883
knn(k=2, p=1, weights=uniform) : 0.845
knn(k=2, p=1, weights=distance) : 0.871
knn(k=2, p=2, weights=uniform) : 0.868
knn(k=2, p=2, weights=distance) : 0.883
knn(k=3, p=1, weights=uniform) : 0.864
knn(k=3, p=1, weights=distance) : 0.874
knn(k=3, p=2, weights=uniform) : 0.888
knn(k=3, p=2, weights=distance) : 0.894
knn(k=4, p=1, weights=uniform) : 0.86
knn(k=4, p=1, weights=distance) : 0.875
knn(k=4, p=2, weights=uniform) : 0.882
knn(k=4, p=2, weights=distance) : 0.892
knn(k=5, p=1, weights=uniform) : 0.852
knn(k=5, p=1, weights=distance) : 0.862
knn(k=5, p=2, weights=uniform) : 0.88
knn(k=5, p=2, weights=distance) : 0.888
knn(k=6, p=1, weights=uniform) : 0.844
knn(k=6, p=1, weights=distance) : 0.864
knn(k=6, p=2, weights=uniform) : 0.871
knn(k=6, p=2, weights=distance) : 0.882
knn(k=7, p=1, weights=uniform) : 0.842
knn(k=7, p=1, weights=distance) : 0.847
knn(k=7, p=2, weights=uniform) : 0.864
knn(k=7, p=2, weights=distance) : 0.873
knn(k=8, p=1, weights=uniform) : 0.846
knn(k=8, p=1, weights=distance) : 0.856
knn(k=8, p=2, weights=uniform) : 0.869
knn(k=8, p=2, weights=distance) : 0.876
knn(k=9, p=1, weights=uniform) : 0.846
knn(k=9, p=1, weights=distance) : 0.857
knn(k=9, p=2, weights=uniform) : 0.87
knn(k=9, p=2, weights=distance) : 0.875
knn(k=10, p=1, weights=uniform) : 0.85
knn(k=10, p=1, weights=distance) : 0.858
knn(k=10, p=2, weights=uniform) : 0.873
knn(k=10, p=2, weights=distance) : 0.873
knn(k=30, p=1, weights=uniform) : 0.808
knn(k=30, p=1, weights=distance) : 0.818
knn(k=30, p=2, weights=uniform) : 0.836
knn(k=30, p=2, weights=distance) : 0.841
```

Optimalni parametri određeni prethodnom evaluacijom su: k=3, p=2, rastojanje=*'distance'*. Prethodno je i očekivano. Naime, s obzirom na prirodu samih podataka jasno je da je Euklidska metrika bolji izbor nego što je Menhetn. Množenje recipročnim rastojanjem prilikom glasanja metodom najbližih suseda se takođe u većini primena ponaša kao optimalno, pogotovo u slučaju prostora male gustine kao što je prostor slika.

Evaluacija na celokupnom skupu podataka prilično je vremenski zahtevna, aproksimativno oko tri sata i trideset minuta[6]. Preciznost koja se postiže ovim metodom jeste impresivnih (s obzirom na kompleksnost samog modela) 98.195%. Ovako velika preciznost postignuta je zahvaljujući pažljivom odabiru parametara metoda.

Odgovarajući izveštaj klasifikacije prikazan je na sledećoj slici:

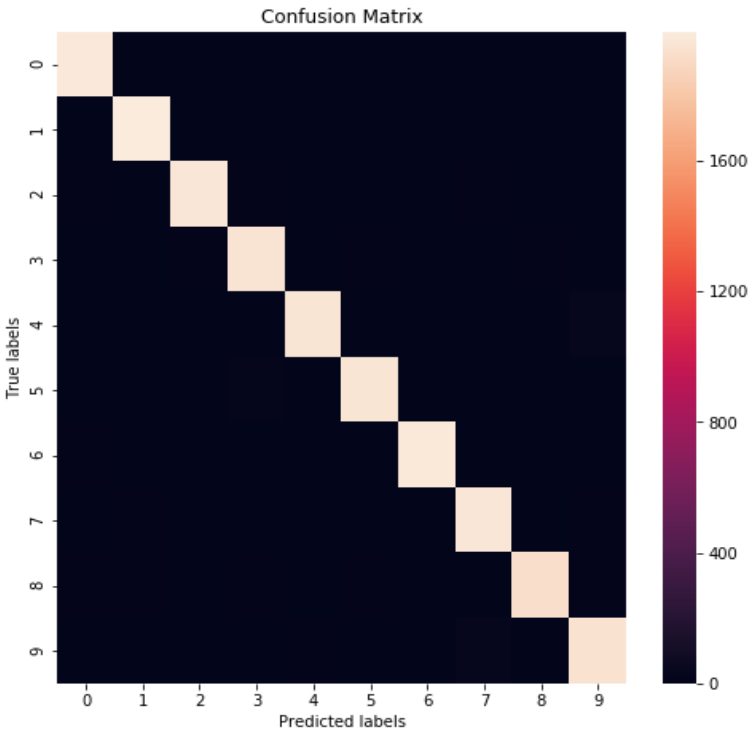
	precision	recall	f1-score	support
0	0.98	0.99	0.99	2000
1	0.98	1.00	0.99	2000
2	0.99	0.98	0.98	2000
3	0.98	0.98	0.98	2000
4	0.99	0.98	0.98	2000
5	0.99	0.98	0.98	2000
6	0.99	0.99	0.99	2000
7	0.98	0.98	0.98	2000
8	0.99	0.96	0.98	2000
9	0.97	0.97	0.97	2000
accuracy			0.98	20000
macro avg	0.98	0.98	0.98	20000
weighted avg	0.98	0.98	0.98	20000

Konkretna matrica konfuzije data je kao:

	0	1	2	3	4	5	6	7	8	9
0	1987	3	1	1	1	1	3	1	0	2
1	0	1995	3	0	1	0	0	1	0	0
2	8	2	1964	10	0	2	1	9	3	1
3	3	4	13	1955	0	9	1	6	8	1
4	1	6	0	0	1958	1	5	2	0	27
5	3	3	1	19	1	1960	6	0	5	2
6	9	5	1	0	1	5	1977	0	2	0
7	1	10	1	0	6	0	0	1969	0	13
8	9	12	4	13	5	10	6	2	1926	13
9	5	3	0	3	12	1	0	24	4	1948

Na osnovu matrice konfuzije zaključujemo da je najčešci pogrešno klasifikovan par upravo par 4 i 9. Ovo je očekivano ponašanje. Naime, iste cifre su I vizuelno slične, stoga su njihove normalizovane reprezentacije dosta bliske, odakle se dešavaju grečke u klasifikaciji. Recimo, par 4 i 2 nikada nije pogrešno klasifikovan.

[6] Dobro je poznato da klasifikacija metodom najbližih suseda predstavlja lenjog klasifikatora, stoga imamo vremenski zahtevan proces predviđanja vrednosti na konkretnim instancama



Napomena: u sedamnaestoj liniji ćelije kojom se biraju pomenuti parametri metoda najbližih suseda prilikom provere "$score > max\_score$" koristi se princip Okamovog žileta. Naime ukoliko dva skupa parametara ostvaruju istu preciznost na skupu za izbor parametara, bira se onaj, uslovno rečeno jednostavniji skup (manje suseda, jednostavnija metrika i sl.).

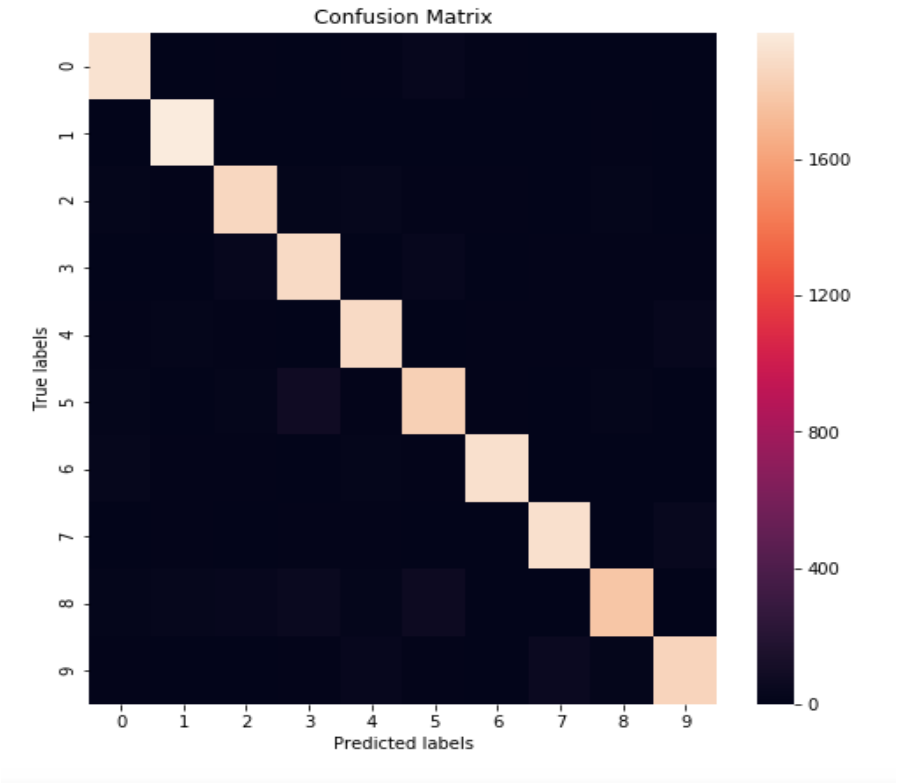
Metod potpornih vektora (SVC)

Linearni SVC

Za rešavanje konkretnog problema klasifikacije primenjena su dve varijacije metoda potpornih vektora: metod potpornih vektora sa linearnim i sa nelinearnim kernelom. S obzirom na konkretnu vremensku složenost algoritma potpornih vektora[7] kao trening skup konkretnog algoritma korišćeno je samo dvadeset odnosno pedeset procenata skupa za trening u cilju izbora potpornih vektora. Isti procenat određen je eksperimentalno, na osnovu gabaritnih ograničenja konkretnog skupa podataka. Konkretan rezultat na kompletnom skupu podataka koji ostvaruje ovakav linearni kernel (treniran na osnovu dvadeset procenata skupa za treniranje) jeste 94.02%. Matrica konfuzije kao i klasifikacioni izveštaj dati su na narednim slikama.

	precision	recall	f1-score	support
0	0.94	0.96	0.95	2000
1	0.95	0.99	0.97	2000
2	0.94	0.93	0.94	2000
3	0.91	0.94	0.93	2000
4	0.93	0.94	0.94	2000
5	0.91	0.91	0.91	2000
6	0.98	0.96	0.97	2000
7	0.95	0.96	0.95	2000
8	0.94	0.89	0.91	2000
9	0.95	0.93	0.94	2000
accuracy			0.94	20000
macro avg	0.94	0.94	0.94	20000
weighted avg	0.94	0.94	0.94	20000

[7] $O(n^3)$, gornja granica za veliko C



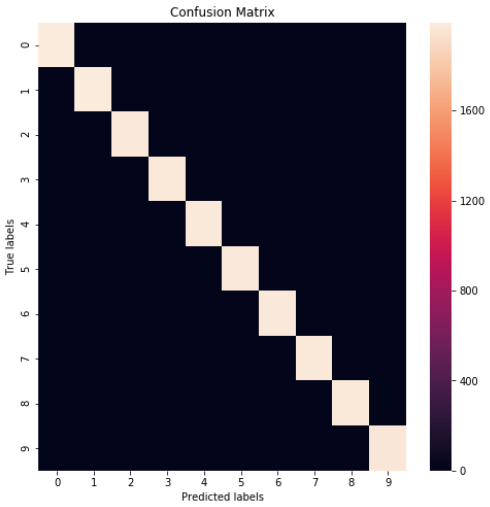
Vidimo daje najveća preciznost ostvarena upravi za klasu 6. Prethodno je očekivano, s obzirom na manji broj instanci u skupu za trening (samo njih dvadeset procenata) cifra 6 jeste najspecifičnija cifra, koja nema slične karaktere među ostalim ciframa (kao što je recimo petica koja se lako meša kako sa osmicom, tako i sa trojkom odnosno devetkom, otuda najmanja preciznost na njoj).

Nelinearni SVC

Konkretni parametri nelinearnog metoda potpornih vektora izabrani su koristeći pomenuti redukovani skup instanci. Na ovom mestu testiraju se polinomijalni kernel, rbf kernel kao i sigmoidni kernel. Vrednost parametra C izabrana je iz skupa *[0.05, 0.1, 1, 5, 10, 100],* dok je parametar gama biran iz skupa *[0.01, 0.1, 0.25, 0.5, 1.0].* Kao i u slučaju izbora parametara metoda k najbližih suseda korišćen je princip Okamovog Žileta. Najbolji parametri dati su sa: $C = 5, gamma = 0.01, kernel = rbf$. Konkretna preciznost ostvarena na ovom mestu jeste 98.815%.

Trening skup se sastoji od pedeset procenata originalnih slika. Konkretna vizuelizacija ostvarenih rezultata data je na narednim slikama.

```
[[1985    0    2    1    3    2    3    1    3    0]
 [   0 1990    4    0    1    2    0    1    2    0]
 [   3    1 1979    4    3    0    1    1    7    1]
 [   0    1   15 1970    0    6    1    2    5    0]
 [   0    0    2    0 1971    2    6    3    1   15]
 [   1    1    1   11    3 1975    1    0    6    1]
 [   4    2    2    0    4    5 1981    0    2    0]
 [   0    9    2    1    3    0    0 1974    3    8]
 [   4    3    4    4    2    6    1    1 1972    3]
 [   4    1    0    3    9    0    0   14    3 1966]]
```



Izveštaji po klasama dati su sa:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	2000
1	0.99	0.99	0.99	2000
2	0.98	0.99	0.99	2000
3	0.99	0.98	0.99	2000
4	0.99	0.99	0.99	2000
5	0.99	0.99	0.99	2000
6	0.99	0.99	0.99	2000
7	0.99	0.99	0.99	2000
8	0.98	0.99	0.99	2000
9	0.99	0.98	0.98	2000
accuracy			0.99	20000
macro avg	0.99	0.99	0.99	20000
weighted avg	0.99	0.99	0.99	20000

Na osnovu matrice konfuzije zaključujemo da je par 4 i 9 I na ovom mestu par sa najviše pogrešnih klasifikacija.

Stabla odlučivanja

Stablo odlučivanja

Stabla odlučivanja pokazuju sve svoje prednosti na konkretnom skupu podataka. To su pre svega njihova kako efikasnost, tako i interpretabilnost. U ovom radu korišćene su dve varijante klasifikacije stablima odlučivanja: pojedinačno stablo odlučivanja i slučajna šuma.

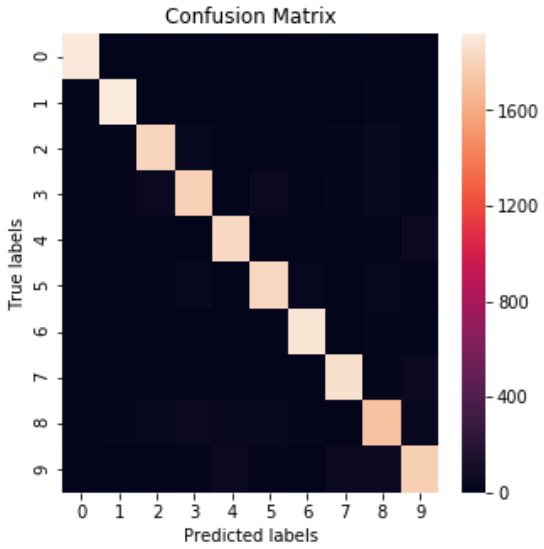
Stablo odlučivanja kreirano je na svim dostupnim podacima. Proces kreiranja stabla traje nešto manje od petnaest minuta. Ostvarena je preciznost *91.515%.* Stablo je kreirano sa podrazumevanim parametrima. Izveštaj klasifikacije dat je na narednoj slici:

91.515

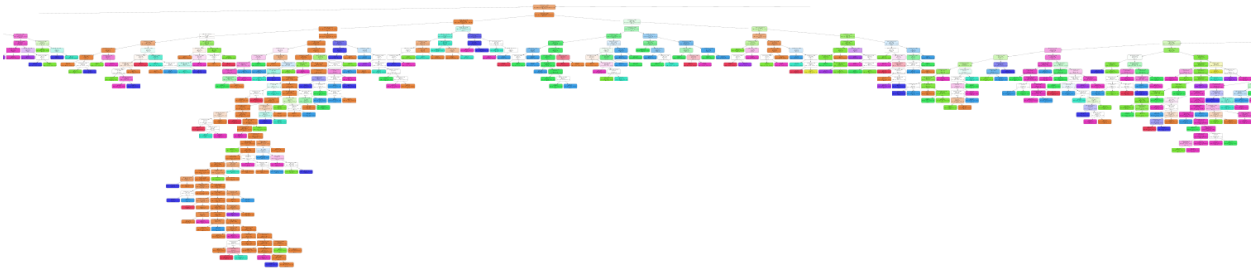
	precision	recall	f1-score	support
0	0.95	0.95	0.95	2000
1	0.95	0.96	0.95	2000
2	0.92	0.91	0.91	2000
3	0.90	0.89	0.90	2000
4	0.91	0.92	0.91	2000
5	0.91	0.91	0.91	2000
6	0.93	0.94	0.94	2000
7	0.92	0.93	0.93	2000
8	0.86	0.86	0.86	2000
9	0.89	0.89	0.89	2000
accuracy			0.92	20000
macro avg	0.92	0.92	0.92	20000
weighted avg	0.92	0.92	0.92	20000

Matrica konfuzije data je sa:

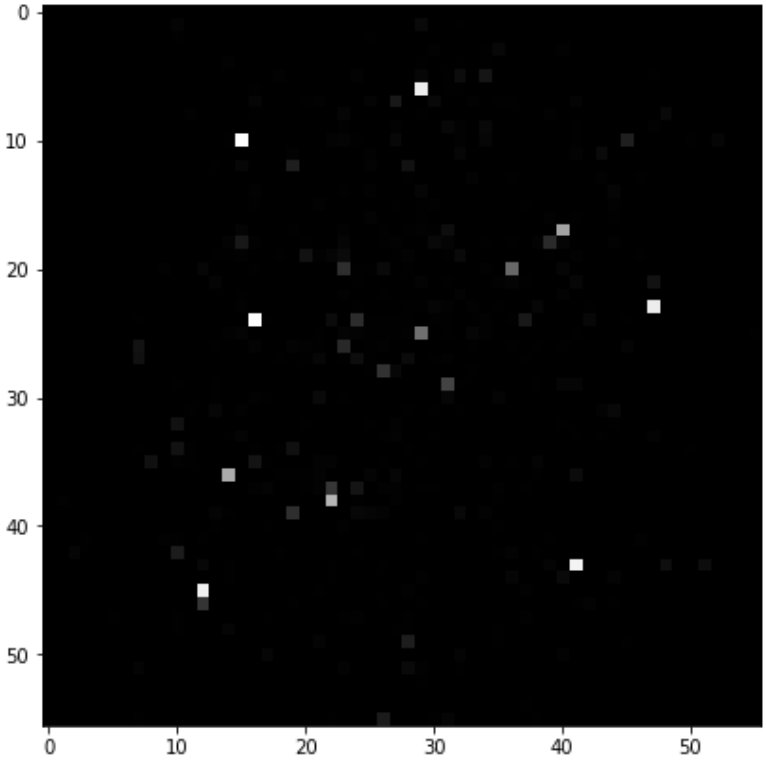
	0	1	2	3	4	5	6	7	8	9
0	1987	3	1	1	1	1	3	1	0	2
1	0	1995	3	0	1	0	0	1	0	0
2	8	2	1964	10	0	2	1	9	3	1
3	3	4	13	1955	0	9	1	6	8	1
4	1	6	0	0	1958	1	5	2	0	27
5	3	3	1	19	1	1960	6	0	5	2
6	9	5	1	0	1	5	1977	0	2	0
7	1	10	1	0	6	0	0	1969	0	13
8	9	12	4	13	5	10	6	2	1926	13
9	5	3	0	3	12	1	0	24	4	1948



Na ovom mestu data je zanimljiva vizuelizacija stabla odlučivanja. Naime, u odgovarajućim propratnim materijalima ovog dokumenta moguće je naći kompletan grafik koji odgovara na ovaj način kreiranom stablu. Isti prikazuje degenerisano veliki rast stabla u širinu, koji odgovara konkretnom skupu podataka. Isečak tog grafikona (apstrahujući detalje, prikazujemo samo strukturu stabla) dat je na sledećoj slici.



Na slici vidimo veliku širinu stabla i spuštanje u dubinu samo u retkim slučajevima, koji ispostavlja se, odgovaraju “bitnijim” pikselima. Bitnost piksela odnosno njihov uticaj na predikciju modela vizuelizovan je na narednoj slici. Ujedno dati su i najznačajniji pikseli.




```
feature (pixel) importances:

pixel (24, 16)    0.055072
pixel (10, 15)    0.054781
pixel (43, 41)    0.052373
pixel (45, 12)    0.051723
pixel (6, 29)     0.051247
pixel (23, 47)    0.051071
pixel (38, 22)    0.039216
pixel (36, 14)    0.037051
pixel (17, 40)    0.035436
pixel (25, 29)    0.023956
pixel (20, 36)    0.022219
pixel (29, 31)    0.014434
pixel (37, 22)    0.011676
pixel (46, 12)    0.011298
pixel (28, 26)    0.010911
pixel (20, 23)    0.009975
```

Prethodna vizuelizacija ima dosta smisla. Naime, ključni pikseli (označeni belom bojom) nalaze se upravo na očekivanim mestima. Ivice slike i bočni delovi ostaju totalno crni, s obzirom da se sami karakteri obično nalaze u centru slike.

Slučajne šume

U radu su implementirane slučajne šume koje koriste kao meru nečistoće bilo ginijev indeks.[8] Kreirane su dve šume od po *100* stabala koje kao meru ne čistoće koriste jedna entropiju druga ginijev indeks. Oba ansambla postigli su istu preciznost *97.475%.* Korišćenje ansambla modela dovodi do drastičnog povećanja preciznosti u odnosu na onu koju je dalo jedno stablo odlučivanja. Prethodno ponašanje je očekivano. Naime, korišćenje većeg broja modela koji grade ansambl teži smanjenju greške klasifikacije s obzirom da se obučava veći broj nezavisnih modela čije su greške nezavisne. Ideja koja se krije iza svega toga jeste da se prilikom agregacije greške koje prave pojedinačna stabla međusobno poništavaju.

[8] Šuma od sto stabala se isto ponaša kada kao meru nečistoće koristi bili ginijev indeks bilo entropiju. Stoga su gušće šume kao meru nečistoće koristile ginijev indeks; Šume sa većim brojem stabala ne povećavaju preciznost u odnosu na šumu od 100 stabala, samo su vremenski nešto zahtevniha. Stoga ovakve šume odnosno modeli nisu dalje razmatrani u konkretnom radu.

Ovaj metod odlučivanja implementiran je uz pomoć biblioteke *RandomForestsClassifier* paketa sklearn.ensamble programskog jezika *Python*.[9]

Konkretni rezultati evaluacije ova dva ansambla dati su u nastavku.

Broj stabala: 100, Mera nečistoće: Ginijev indeks:

accuracy: 97.475

	precision	recall	f1-score	support
0	0.98	0.98	0.98	2000
1	0.98	0.99	0.99	2000
2	0.97	0.98	0.98	2000
3	0.97	0.97	0.97	2000
4	0.96	0.98	0.97	2000
5	0.98	0.97	0.98	2000
6	0.99	0.98	0.98	2000
7	0.98	0.97	0.98	2000
8	0.96	0.96	0.96	2000
9	0.96	0.97	0.96	2000
accuracy			0.97	20000
macro avg	0.97	0.97	0.97	20000
weighted avg	0.97	0.97	0.97	20000

	0	1	2	3	4	5	6	7	8	9
0	1987	3	1	1	1	1	3	1	0	2
1	0	1995	3	0	1	0	0	1	0	0
2	8	2	1964	10	0	2	1	9	3	1
3	3	4	13	1955	0	9	1	6	8	1
4	1	6	0	0	1958	1	5	2	0	27
5	3	3	1	19	1	1960	6	0	5	2
6	9	5	1	0	1	5	1977	0	2	0
7	1	10	1	0	6	0	0	1969	0	13
8	9	12	4	13	5	10	6	2	1926	13
9	5	3	0	3	12	1	0	24	4	1948

[9] Detaljnije informacije mogu se naći na adresi: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Broj stabala: 100, Mera nečistoće: Entropija:

accuracy: 97.475

	precision	recall	f1-score	support
0	0.98	0.98	0.98	2000
1	0.98	0.99	0.99	2000
2	0.97	0.98	0.98	2000
3	0.97	0.97	0.97	2000
4	0.96	0.98	0.97	2000
5	0.98	0.97	0.98	2000
6	0.99	0.98	0.98	2000
7	0.98	0.97	0.98	2000
8	0.96	0.96	0.96	2000
9	0.96	0.97	0.96	2000
accuracy			0.97	20000
macro avg	0.97	0.97	0.97	20000
weighted avg	0.97	0.97	0.97	20000

	0	1	2	3	4	5	6	7	8	9
0	1987	3	1	1	1	1	3	1	0	2
1	0	1995	3	0	1	0	0	1	0	0
2	8	2	1964	10	0	2	1	9	3	1
3	3	4	13	1955	0	9	1	6	8	1
4	1	6	0	0	1958	1	5	2	0	27
5	3	3	1	19	1	1960	6	0	5	2
6	9	5	1	0	1	5	1977	0	2	0
7	1	10	1	0	6	0	0	1969	0	13
8	9	12	4	13	5	10	6	2	1926	13
9	5	3	0	3	12	1	0	24	4	1948

Duboka neuronska mreža

Konkretan model kreiran je u okviru biblioteke Keras, programskog jezika Python.[10]

Konvolutivne duboke neuronske mreže predstavljaju klasifikatore koji su se izuzetno dobro pokazali u obradi sirovih podataka, kakve su slike. Oni u obzir uzimaju i spacijalnu informaciju o međusobnom susedstvu i lokalnim odnosima konkretnih piksela, koja se pak većim delom zanemaruje ako se slika dimenzija 56×56 posmatra kao vektor dimenzija 3136. Ispostavlja se da prethodni odnosi nose u sebi veliku količinu informacija, koja pak dovodi do toga da je ova vrsta klasifikatora ostvaruje najveću preciznost na konkretnom zadatku klasifikacije. Dodatno, gabaritna ograničenja konkretnog skupa podataka koja su otežavala primenu prethodnih metoda (zbog čega je recimo za trening skup metoda potpornih vektora korišćeno svega dvadeset procenata originalnog trening skupa) na ovom mestu se prevazilaze korišćenjem grafičkih kartica. Naime, tehnikom korišćenja GPU resursa u kombinaciji sa bibliotekom Keras koja je pak implementirana na Tensorflow softveru omogućava direktnu paralelizaciju izračunavanja gradijenata koji pak predstavljaju suštinu obučavanja neuronske mreže, odnosno algoritma propagacije unazad. Time se prevazilaze ograničenja prostora i vremena koja nameće konkretan skup podataka. Umesto control-flow paradigme, na ovom mestu koristi se data-flow paradigma koja je u novije vreme žargonski rečeno direktno odgovorna za velike uspehe računarstva.

Arhitektura konkretne neuronske mreže rađena je po uzoru na takozvani CaffeNet, arhitekturu objavljenu 2014. godine. Mreža se sastoji iz dva dela: konvolutivne i potpuno povezane neuronske mreže koja je na nju nadovezana. Konvolutivna mreža sastoji se od tri bloka, od kojih svaki čine dva sloja dvodimenzione konvolucije, za kojim sledi sloj unutrašnje normalizacije. Nakon unutrašnje normalizacije sledi sloj agregacije koji ulaz agregira kernelima širine 2×2 odnosno 3×3. Korišćeni su filteri dimenzija 3×3, kao i ispravljena linearna jedinica kao aktivaciona funkcija. Potpuno povezana neuronska mreža sastoji se od tri sloja, dimenzija 768, 256 i 10 neurona redom. Pri tome poslednji sloj predstavlja sloj koji daje raspodelu verovatnoća za svaku od deset klasa, a samim tim i konkretno predviđanje za dati ulaz. U okviru potpuno povezane neuronske mreže korišćena je regularizacija izbacivanjem (eng. Dropout) kao tehnika smanjenja preprilagođavanja klasifikatora trening instancama. Neuronski klasifikator kao ulaze prima slike dimenzija 56×56 transformisane u odgovarajući oblik (eng. Channel last) a kao izlaz daje predikciju jedne od deset mogućih cifara. Mreža kao takva ima nešto više od pola miliona parametara. Detaljniji opis arhitekture dat je na narednoj slici.

[10] https://keras.io/

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 56, 56, 1)	0
conv2d_1 (Conv2D)	(None, 54, 54, 32)	320
conv2d_2 (Conv2D)	(None, 52, 52, 32)	9248
batch_normalization_1 (Batch	(None, 52, 52, 32)	128
max_pooling2d_1 (MaxPooling2	(None, 26, 26, 32)	0
conv2d_3 (Conv2D)	(None, 24, 24, 48)	13872
conv2d_4 (Conv2D)	(None, 22, 22, 48)	20784
batch_normalization_2 (Batch	(None, 22, 22, 48)	192
max_pooling2d_2 (MaxPooling2	(None, 11, 11, 48)	0
conv2d_5 (Conv2D)	(None, 9, 9, 64)	27712
conv2d_6 (Conv2D)	(None, 7, 7, 64)	36928
batch_normalization_3 (Batch	(None, 7, 7, 64)	256
max_pooling2d_3 (MaxPooling2	(None, 2, 2, 64)	0
flatten_1 (Flatten)	(None, 256)	0
dense_1 (Dense)	(None, 768)	197376
dropout_1 (Dropout)	(None, 768)	0
next_to_last (Dense)	(None, 256)	196864
dropout_2 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 10)	2570

Total params: 506,250
Trainable params: 505,962
Non-trainable params: 288

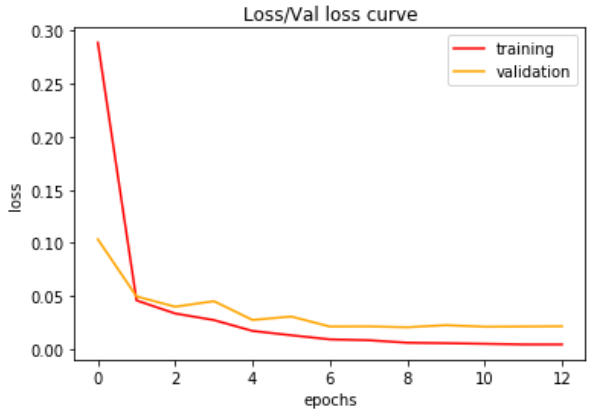
Za obučavanje neuronske mreže korišćen je optimizacioni pristup. Konkretan optimizator koji je korišćen jeste široko popularni Adam, sa parametrima: korak učenja je 10^{-3} a odgovarajući momenti $\beta_1 = 0.9, \beta_2 = 0.999$.

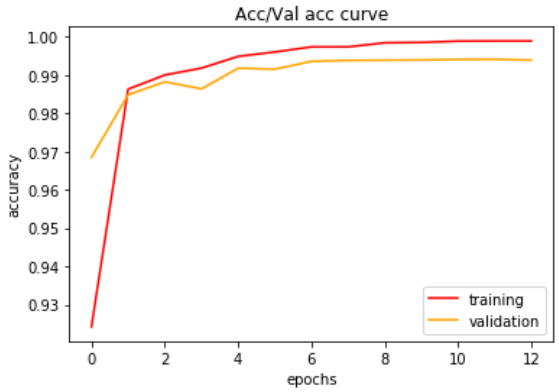
Od pomoćnih tehnika, tokom procesa učenja korišćeno je rano zaustavljanje, kao i smanjenje koraka učenja prilikom nailaska na ravne delove prostora greske. S obzirom na jako veliku količinu podataka za trening, prilikom treninga neuronske mreže korišćen je objekat generator koji je u svakom trenutku iz

RAM memorije modelu dostavljao određenu količinu podataka (eng. Batch) za računanje gradijenata odnosno računanje parametara. Dvadeset procenata slika korišćeno je u svrhe validacije.

Sam model je treniran petnaest epoha, pri čemu se sam zaustavio nakon trinaeste, uz pomoć tehnike ranijeg zaustavljanja. Model je ostvario preciznost*: 99.89%/99.38%/99.415%* na trening, validacionom odnosno test skupu redom. Time je, na konkretnom zadatku klasifikacije nadmašio prethodno prezentovane tehnike.

Grafik promene greške (eng. loss) odnosno preciznosti (eng. accuracy) tokom procesa trening dati su na sledećim slikama:



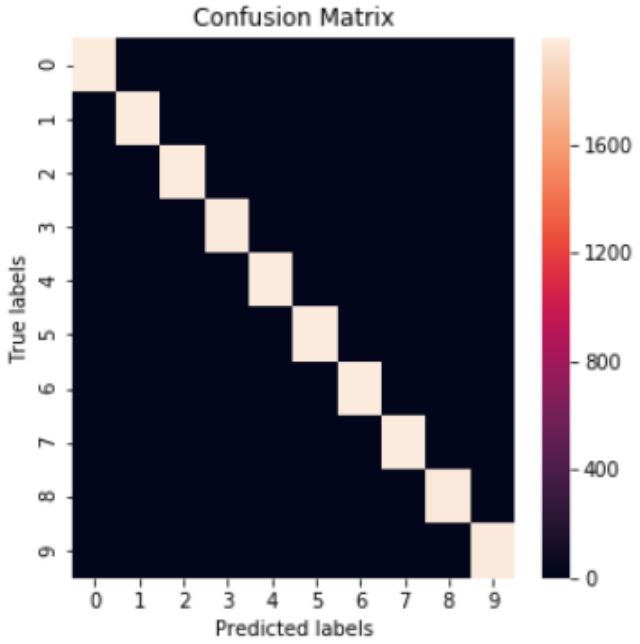


Odgovarajuće informacije o kvalitetu klasifikacije po klasama (preciznost, odziv, f1 mera) dati su na sledećoj slici:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2000
1	0.99	1.00	1.00	2000
2	0.99	0.99	0.99	2000
3	1.00	0.99	0.99	2000
4	0.99	0.99	0.99	2000
5	0.99	0.99	0.99	2000
6	1.00	1.00	1.00	2000
7	0.99	0.99	0.99	2000
8	0.99	0.99	0.99	2000
9	0.99	0.99	0.99	2000
accuracy			0.99	20000
macro avg	0.99	0.99	0.99	20000
weighted avg	0.99	0.99	0.99	20000

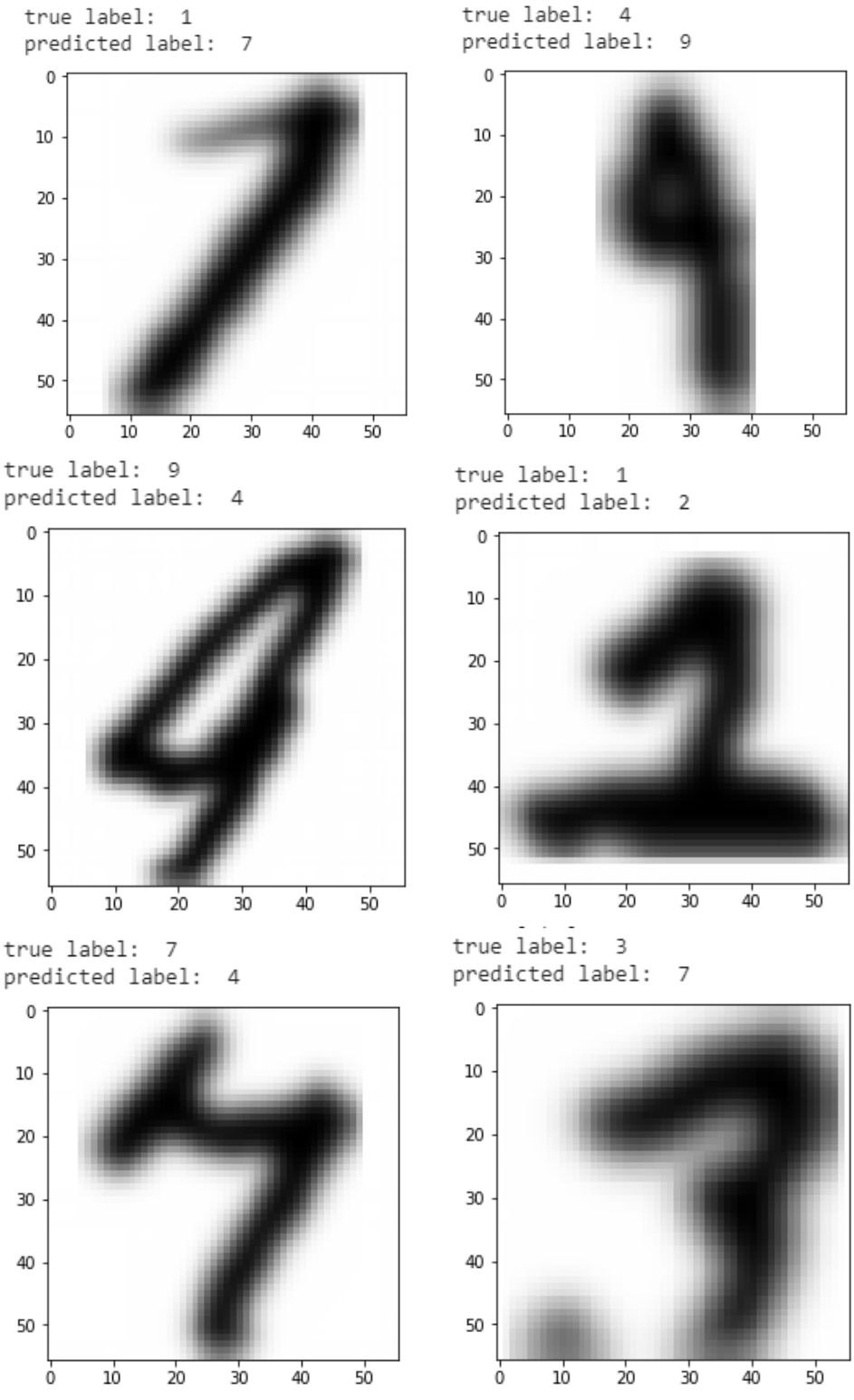
Na osnovu prethodnog vidimo da se model najbolje ponaša na instancama klasa tri, šest i jedan. Ovo je očekivano, s obzirom na rezultate orethodnih modela koji su upravo na ovim instancama davali najveću preciznost.

Konkretnu matricu konfuzije navodimo u nastavku, kako u numeričkom, tako i u grafičkom obliku.



	0	1	2	3	4	5	6	7	8	9
0	1991	1	1	0	0	1	3	0	2	1
1	0	1994	4	0	0	0	0	2	0	0
2	2	1	1987	1	1	0	0	6	2	0
3	0	0	7	1984	0	4	0	3	2	0
4	0	1	0	0	1988	0	2	2	1	6
5	0	0	1	3	2	1990	1	0	3	0
6	2	2	1	0	0	1	1992	0	2	0
7	1	6	2	0	2	1	0	1987	0	1
8	2	0	4	0	0	4	0	0	1985	5
9	3	0	0	0	7	0	0	4	1	1985

Neki primeri pogrešno klasifikovanih instanci dati su u nastavku. Više ovakvih primera moguće je videti u konkretnoj Jupyther svesci u koja prati ovaj dokument.



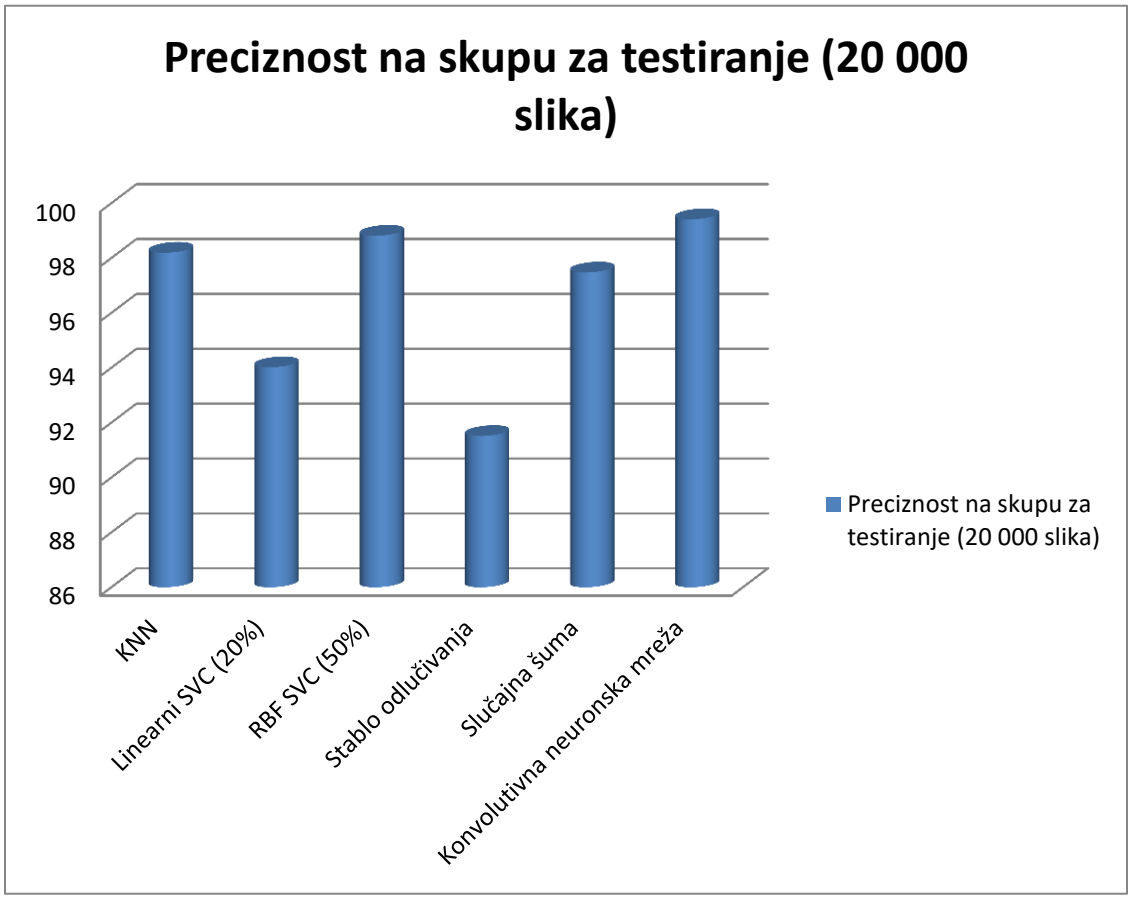
Na osnovu prethodnih primera vidimo da model greši u onim situacijama u kojima čak ni ljudsko oko sa sigurnošću ne može trvrditi konkretnu labelu. Ovo nam predstavlja potvrdu da je model postigao jako veliku preciznost. Subjektivno govoreći, ako bi model imao preciznost jedan to ne bi bilo realno

ponašanje već neka vrsta preprilagođavanja podacima, s obzirom da je rukopis jako netipična stvar, i uvek postoji netipičnih predstavnika.

Zaključak

Konkretni finalni rezultati na skupu za evaluaciju dati su sa:

Metod klasifikacije	Preciznost
KNN	98.195%
Linearni SVC (korišćeno 20% trening skupa)	94.020%
RBF SVC (korišćeno 50% trening skupa)	98.815%
Stablo odlučivanja	91.515%
Šuma stabala odlučivanja	97.475%
Konvolutivna neuronska mreža	99.415%



Odgovarajuće vreme potrebno za trening odnosno test dato je sa:

Model	Trening (s)	Evaluacija (s)
KNN	57.9	12780.3
Linearni SVC (20%)	342	382
RBF SVC (50%)	3604.3	1538
Stablo odlučivanja	780.73	1
Slučajna šuma	600	10
Konvolutivna neuronska mreža	299	2